Technická univerzita v Košiciach

Fakulta elektrotechniky a informatiky

Návrh metódy výberu príznakov na báze k–NN algoritmu

Diplomová práca

2018 **Peter Bugata**

Technická univerzita v Košiciach

Fakulta elektrotechniky a informatiky

Návrh metódy výberu príznakov na báze k–NN algoritmu

Diplomová práca

Študijný program:	Informatika
Študijný odbor:	9.2.1 Informatika
Školiace pracovisko:	Katedra počítačov a informatiky (KPI)
Školiteľ:	doc. Ing. Peter Drotár, PhD.

Košice 2018 **Peter Bugata**

Abstrakt

V súčasnosti sme zaplavení veľkým množstvom údajov, pričom často ide o dáta veľkých rozmerov. Vysoká dimenzionalita dát nepriaznivo vplýva na výsledky algoritmov strojového učenia a ich spracovanie vyžaduje veľa času a pamäťového priestoru. Často používanou technikou predspracovania dát, ktorá slúži na redukciu dimenzionality, je výber príznakov, resp. premenných. Diplomová práca predstavuje novú metódu výberu príznakov založenú na algoritme k–NN (k najbližších susedov). Metóda využíva atribútovo vážené k–NN a k–NN vážené podľa vzdialenosti, pričom vychádza z predpokladu, že cieľovú premennú najlepšie určujú najviac relevantné premenné najbližších susedov. Na nájdenie optimálnych váh premenných používa metódu najväčšieho spádu. Prezentovaná metóda je parametrizovaná z hľadiska definície vzdialenosti, hodnotiacej funkcie vzdialenosti a funkcie chyby. V práci boli na hodnotenie navrhovanej metódy použité niektoré synteticky generované súbory dát a tiež vysokodimenzionálne reálne dáta. Experimentálne výsledky ukazujú, že metóda je plne porovnateľná s populárnymi metódami výberu príznakov z hľadiska schopnosti identifikovať relevantné premenné a z hľadiska presnosti predikcie pri využití vybraných algoritmov strojového učenia.

Kľúčové slová

Výber príznakov, redukcia dimenzionality, k–NN, metóda najväčšieho spádu, strojové učenie

Abstract

We are surrounded by huge amounts of data that often have high dimensionality. High–dimensional data adversely affect results of machine learning algorithms and their processing requires a lot of computational time and storage space. Feature selection is frequently used preprocessing technique for dimensionality reduction. In the diploma thesis, new feature selection method based on k–nearest neighbours (k–NN) algorithm is presented. The method uses distance and attribute–weighted k–NN and assumes that target variable is the most accurately determined by the most relevant features of the nearest neighbours. New method uses gradient descent algorithm to find optimal weights of features. The method is parametrized to use various metrics, kernels and loss functions. In the thesis, newly proposed method has been evaluated on some artificial and high–dimensional real datasets. Experimental results show that it is fully comparable with the state of the art feature selection algorithms in terms of the ability to select relevant features and prediction accuracy using selected machine learning algorithms.

Keywords

Feature selection, dimensionality reduction, k–NN, gradient descent, machine learning

TECHNICKÁ UNIVERZITA V KOŠICIACH

FAKULTA ELEKTROTECHNIKY A INFORMATIKY

Katedra počítačov a informatiky

ZADANIE DIPLOMOVEJ PRÁCE

Študijný odbor: **Informatika**
Študijný program: **Informatika**

Názov práce:

Návrh metódy výberu príznakov na báze k-NN algoritmu

Feature selection based on K nearest neighbours algorithm

Študent: **Bc. Peter Bugata**
Školiteľ: **doc. Ing. Peter Drotár, PhD.**
Školiace pracovisko: **Katedra počítačov a informatiky**
Konzultant práce:
Pracovisko konzultanta:

Pokyny na vypracovanie diplomovej práce:

1. V rámci riešenia diplomovej práce navrhnite metódu výberu príznakov vhodnú pre aplikáciu na dátach veľkých rozmerov.
2. Naštudujte problematiku výberu príznakov a existujúce metódy výberu príznakov.
3. Naštudujte princíp etódy k-najbližších susedov (k-NN).
4. Navrhnite metódu výberu príznakov založenú na k-NN.
5. Porovnajte navrhnutú metódu s existujúcimi riešeniami.

Jazyk, v ktorom sa práca vypracuje: slovenský
Termín pre odovzdanie práce: 27.04.2018
Dátum zadania diplomovej práce: 31.10.2017

.....
doc. Ing. Jaroslav Porubän, PhD.
vedúci garantujúceho pracoviska

TECHNICKÁ UNIVERZITA V KOŠICIACH FAKULTA ELEKTROTECHNIKY A INFORMATIKY -12-

.....
prof. Ing. Liberios Vokorokos, PhD.
dekan fakulty

Čestné vyhlásenie

Vyhlasujem, že som diplomovú prácu vypracoval samostatne s použitím uvedenej odbornej literatúry.

Košice 27. 4. 2018 .....

Vlastnoručný podpis

Poďakovanie

Touto cestou by som chcel vyjadriť poďakovanie doc. Ing. Petrovi Drotárovi, PhD. za jeho odborné vedenie, podporu a cenné rady, ktoré mi poskytol pri spracovaní diplomovej práce.

Táto práca bola realizovaná s podporou Agentúry pre podporu výskumu a vývoja v rámci projektu APVV–16–0211.

Obsah

Zoznam obrázkov

Zoznam tabuliek

Zoznam symbolov a skratiek

AE	Absolute Error
ANOVA	Analysis of Variance
CE	Cross–Entropy
CW	Weighted Consistency Index
dCor	Distance Correlation
GD	Gradient Descent
HL	Huber Loss function
k–NN	k–Nearest Neighbours
LOO	Leave–One–Out
LR	Linear Regression
MAE	Mean Absolute Error
MHL	Mean Huber Loss
MI	Mutual Information
MID	Mutual Information Difference Criterion
MIQ	Mutual Information Quotient Criterion
mRMR	minimal Redundancy Maximal Relevance
MSE	Mean Squared Error
NB	Gaussian Naive Bayes
RBF	Radial Basis Function

RF	Random Forrest
RMSE	Root Mean Squared Error
SE	Squared Error
SGD	Stochastic Gradient Descent
Suc.	Index of Success
SVM	Support Vector Machine
TF	True Features
WkNN–FS	Weighted Nearest Neighbours Feature Selection

Úvod

Vývoj v oblasti získavania digitálnych údajov a v oblasti technológií ich spracovania a ukladania spôsobil, že sprievodným javom takmer každej ľudskej činnosti sa stalo generovanie a zbieranie obrovského množstva dát [1]. Okrem svojho prvotného použitia, prevažne v transakčných databázových systémoch, poskytujú zozbierané dáta možnosti pre ďalšiu analýzu a výskum. Disciplína, ktorá sa zaoberá extrakciou užitočných informácií z dát, je známa pod názvom *data mining* alebo *knowledge discovery from data*, teda objavovanie znalostí v dátach [2].

Data mining môže byť definovaný ako proces analýzy údajov za účelom nájdenia nových, neočakávaných vzťahov a zaujímavých vzorov užitočných pre ich vlastníkov. V tomto procese sú využívané databázové technológie, štatistické nástroje, ako aj algoritmy strojového učenia – *machine learning* [1]. Cieľom strojového učenia, ako súčasti umelej inteligencie, je vývoj počítačových systémov schopných zlepšovať sa s pribúdajúcimi skúsenosťami [3]. *Data mining* je oblasťou, kde sa algoritmy strojového učenia bežne využívajú na získavanie znalostí z dát. Pritom učenie znamená, že systém získava skúsenosti analyzovaním známej množiny dát, na základe nich robí rozhodnutia a s každou novou množinou dát je schopný sa zlepšovať.

Jeden z typov strojového učenia je známy ako učenie s učiteľom – *supervised learning*. Jeho hlavným cieľom je na základe známych údajov vybudovať model, ktorý môže byť použitý na predpovedanie, teda predikciu vlastností neznámych alebo nových údajov [4]. Používané súbory dát sú obvykle reprezentované maticou, v ktorej riadkoch sú uložené záznamy, čiže pozorovania. Stĺpce matice zodpovedajú ich jednotlivým vlastnostiam – atribútom alebo premenným. Osobitné postavenie má tzv. cieľová premenná, ktorej hodnoty sú predikované v závislosti od ostatných – vysvetľujúcich premenných alebo príznakov. Algoritmus vytvorí model na základe tréningovej množiny dát, v ktorej sú hodnoty cieľovej premennej pre všetky pozorovania známe. Vybudovaný model potom možno použiť na predikciu hodnôt cieľovej premennej pre nové pozorovania.

V prediktívnom modelovaní existujú dva typy úloh, a to klasifikácia a regresia. V klasifikačných úlohách je cieľom predikovať pre nové pozorovania príslušnosť k určitej skupine – triede. Ak sú v úlohe uvažované dve triedy, ide o binárnu klasifikačnú úlohu. Príkladom môže byť úloha, v ktorej na základe symptómov a výsledkov rôznych vyšetrení je potrebné predikovať, či pacient má určitú diagnózu alebo nie. V regresných úlohách má cieľová premenná spojitý charakter, t. j. predikované hodnoty sú reálne čísla. Ak v predchádzajúcom príklade bude potrebné predikovať výšku nákladov na diagnostiku a liečbu pacienta v ďalšom období, pôjde o regresnú úlohu.

Ako bolo spomenuté vyššie, údaje používané pri objavovaní nových znalostí boli väčšinou pôvodne zbierané na iný účel. Z hľadiska cieľov predikcie môžu byť preto niektoré atribúty málo užitočné, avšak dopredu nie je známe, ktoré z nich by bolo možné z analýzy vylúčiť. Pritom súbory dát často obsahujú veľmi veľký počet atribútov, resp. vysvetľujúcich premenných, a vytvárajú tak vysokorozmerný priestor. Vysoká dimenzionalita kladie zvýšené požiadavky na výpočtové zdroje a predlžuje čas výpočtu. U mnohých algoritmov strojového učenia sa navyše prejavuje fenomén známy ako prekliatie dimenzionality (*curse of dimensionality*) [5], ktorý spôsobuje výrazné zníženie presnosti predikcie vo vysokorozmerných priestoroch.

S problémom veľkého počtu atribútov sa možno stretnúť v mnohých doménových oblastiach, napríklad v oblasti spracovania textov internetových dokumentov, v oblasti biomedicíny, chémie a pod. Preto je veľká pozornosť venovaná vývoju metód výberu premenných – *feature selection*, ktoré z množiny všetkých vysvetľujúcich premenných vyberajú menšie podmnožiny relevantných premenných dôležitých pre presnosť predikcie [6], [7]. Veľký význam má aj ďalšia analýza vyvinutých metód s cieľom určiť, pre aké typy súborov dát sú vhodné a aké je optimálne nastavenie ich parametrov.

1 Formulácia úlohy a ciele práce

Prvým cieľom predkladanej práce je štúdium problematiky výberu premenných (*feature selection*) spojené s analýzou a implementáciou niektorých existujúcich metód. Podrobnejšie skúmanou metódou bude metóda mRMR [8], chápaná ako flexibilný rámec pre výber premenných. Algoritmus mRMR bude preformulovaný na ekvivalentnú optimalizačnú úlohu a bude analyzovaný jeho vzťah k algoritmu na hľadanie podmnožiny vysvetľujúcich premenných s maximálnou relevanciou voči cieľovej premennej. Metóda mRMR bude implementovaná v jazyku Python v zovšeobecnenom tvare tak, aby bolo možné využívať rôzne miery korelácie.

Ďalším cieľom práce je naštudovať metódu k–NN (k najbližších susedov) [4] a navrhnúť jej špeciálnu úpravu, na základe ktorej bude spracovaný podrobný návrh novej metódy na výber premenných. V rámci návrhu novej metódy budú popísané jej teoretické východiská a matematický základ. Následne bude definovaný presný algoritmus pre realizáciu výberu premenných a vytvorená jeho implementácia v jazyku Python. Pre lepšie vysvetlenie metódy bude uvedený jednoduchý príklad jej použitia a spôsobu aplikovania výsledku. V novej metóde budú podporené široké možnosti parametrizácie s použitím rôznych metrík, funkcií pre výpočet chyby predikcie a podobne. Navrhnutá metóda bude pri vhodnej parametrizácii použiteľná na výber premenných pri riešení regresných úloh, ako aj v prípade binárnej klasifikácie. Pritom je potrebné realizovať jej overenie a vyhľadanie optimálnych parametrov pre rôzne syntetické súbory dát aj voľne dostupné vysokodimenzionálne reálne dáta, napríklad z biomedicínskej oblasti.

Posledným cieľom práce je porovnanie výsledkov výberu premenných pomocou novej metódy pri rôznych nastaveniach parametrov s výsledkami dosiahnutými rôznymi variantmi metódy mRMR a ďalšími populárnymi metódami pre výber premenných. Vybrané metódy budú porovnané z hľadiska úspešnosti výberu relevantných premenných, vplyvu na presnosť predikcie pri použití známych metód strojového učenia a z hľadiska stability.

2 Analýza problému

Pri riešení predikčných úloh sa často vyskytuje problém vysokej dimenzionality, keď súbor dát (*dataset*) obsahuje veľmi veľký počet atribútov, resp. vysvetľujúcich premenných. Redukcia dimenzionality môže byť dosiahnutá dvomi spôsobmi, a to výberom premenných (*variable selection*) alebo konštrukciou odvodených príznakov (*feature extraction*, *feature construction*) [7]. Kým prvý spôsob odstraňuje z pôvodných premenných tie, ktoré prinášajú malú alebo žiadnu pridanú informáciu o cieľovej premennej, druhý spôsob znižuje dimenzionalitu vytvorením novej množiny príznakov odvodených z pôvodných premenných. Nová množina býva obvykle kompaktnejšia a efektívnejšia z hľadiska predikcie [6].

Niekedy sa v rovnakom význame ako výber premenných používa aj pojem výber príznakov (*feature selection*) pre prípad, že výber sa realizuje nielen zo základných (*raw*) premenných, ale aj z odvodených príznakov, ktoré boli získané z pôvodných premenných napríklad agregáciou alebo zložitejším predspracovaním [6].

2.1 Metódy pre výber premenných

Výber premenných možno charakterizovať ako techniku strojového učenia, ktorá detekuje relevantné premenné a odstraňuje tie, ktoré sú irelevantné alebo redundantné [7]. Zjednodušene možno povedať, že vysvetľujúca premenná je relevantná, ak obsahuje určité množstvo informácie o cieľovej premennej. Keď možno niektorú vysvetľujúcu premennú odvodiť z iných vysvetľujúcich premenných, tak je redundantná. Nemusí byť však jednoduché určiť redundanciu, ak nejde o závislosť od jednej premennej, ale od celej množiny iných premenných [7].

Cieľom výberu premenných je získať relatívne malú podmnožinu premenných, ktorá zjednoduší definíciu problému. Selekcia premenných umožňuje zvyšovať výkonnosť algoritmov strojového učenia, zlepšiť pochopenie dát a uľahčiť ich vizualizáciu. Výberom premenných dochádza k redukcii dát, čo pomáha šetriť výpočtové zdroje a zjednodušovať modely, a teda napokon aj zrýchliť spracovanie.

Metódy pre výber premenných možno na základe spôsobu ich spolupráce s metódami strojového učenia rozdeliť podľa [6] na tri základné typy, a to filter, obalenie (*wrapper*) a vloženú (*embedded*) metódu.

- Metódy typu filter vyberajú premenné ako samostatný krok predspracovania dát [6]. Výber premenných nezávisí od algoritmu strojového učenia, ktorý je spúšťaný až po výbere premenných. Tieto metódy často odhadujú tzv. skóre relevancie premenných voči cieľovej premennej a potom vyberajú premenné s najlepšími hodnotami skóre [9]. Výhodou je ich rýchlosť a nižšie výpočtové nároky ako pri metódach typu obalenie, a preto sú vhodné na použitie pri veľkých súboroch dát.

- Metódy typu obalenie využívajú na výber premenných nejaký algoritmus strojového učenia, ktorý používajú ako čiernu skrinku [6]. V jednotlivých krokoch postupne generujú podmnožiny premenných, aplikujú na nich metódu strojového učenia a dosiahnutú presnosť predikcie použijú na ohodnotenie vybranej podmnožiny premenných. Kvôli interakcii s vybraným predikčným modelom obyčajne dávajú lepšie výsledky ako filtre, ale sú výpočtovo náročnejšie.

- Vložené metódy vykonávajú výber premenných v procese trénovania, sú priamo zabudované do konštrukcie algoritmu strojového učenia. Sú oveľa efektívnejšie ako metódy typu obalenie, pretože nie je potrebné v každom kroku budovať model nanovo. Navyše sú tieto metódy v porovnaní s metódami typu obalenie odolnejšie voči pretrénovaniu [10].

V posledných rokoch sa objavujú nové prístupy (podľa [10]), ktoré kombinujú existujúce metódy alebo ďalšie techniky strojového učenia, aby boli schopné plniť požiadavky na spracovanie stále objemnejších a zložitejších dát. Reprezentantmi týchto prístupov sú napríklad hybridné a skupinové (*ensemble*) metódy:

- Hybridné metódy [10] sú kombináciou dvoch alebo viacerých metód často rôznych typov, ako napríklad filter pre rýchly výber z veľkého množstva premen-

ných a následne obalenie pre ďalšie upresnenie výberu. Môžu využiť výhody jednotlivých metód kombinovaním ich komplementárnych vlastností. Používajú rôzne vyhodnocovacie kritériá v rôznych fázach výberu premenných, čím zvyšujú efektívnosť a výkon predikcie.

- Skupinové metódy konštruujú niekoľko podmnožín premenných a tie potom agregujú do výslednej podmnožiny [9]. Jednotlivé podmnožiny premenných získavajú tak, že na náhodne vybranej vzorke pozorovaní zo súboru dát spúšťajú základnú metódu výberu premenných. Agregáciou získaná výsledná množina vybraných premenných je potom stabilnejšia ako v prípade použitia základnej metódy na celom súbore dát.

Pri výbere premenných sa často používa tzv. pažravý algoritmus (*greedy search*). Takýto algoritmus v každom kroku vyberá lokálne optimálne riešenie a „dúfa", že je to aj globálne optimum. Pritom nikdy neopravuje svoje predchádzjúce rozhodnutia na základe nových skutočností. Podľa [6] je pažravý algoritmus vo väčšine prípadov výpočtovo výhodný a odolný voči pretrénovaniu.

Používajú sa dva základné spôsoby vyhľadávania, a to dopredná selekcia (*forward selection*) a spätná eliminácia (*backward elimination*). Pri doprednej selekcii je na začiatku množina vybraných premenných prázdna a algoritmus do nej v každom kroku pridá jednu – z jeho pohľadu najlepšiu premennú. Pri spätnej eliminácii sa začína množinou všetkých premenných a algoritmus z nej v každom kroku odoberie z jeho pohľadu najmenej užitočnú premennú.

Metóda spätnej eliminácie je vhodná len pre dáta s nižšou dimenziou, lebo pre vysokorozmerné dáta je výpočtovo náročná. Metóda doprednej selekcie je použiteľná aj pre vysokorozmerné dáta, avšak v niektorých prípadoch jej použitie nie je vhodné. Príkladom môže byť súbor dát XOR popísaný v [6] s dvomi premennými, ktoré majú spolu veľmi dobrý predikčný výkon, ale jednotlivo sa každá z nich javí ako neužitočná. Keďže metóda doprednej selekcie pridáva v každom kroku len jednu premennú, môže sa stať, že nevyberie žiadnu z relevantných premenných.

2.2 Metóda mRMR

Metóda mRMR (*minimal Redundancy Maximal Relevance*) je filter s doprednou selekciou vysvetľujúcich premenných. V každom kroku vyberá ďalšiu vysvetľujúcu premennú tak, aby vzniknutá množina premenných mala čo najväčšiu relevanciu voči cieľovej premennej a zároveň čo najmenšiu vzájomnú redundanciu. Táto metóda na výber premenných bola predstavená a rozpracovaná v článkoch [8] a [11].

2.2.1 Základná metóda mRMR

Súbor dát možno reprezentovať maticou, v ktorej riadkoch sú uložené pozorovania a stĺpce zodpovedajú vysvetľujúcim premenným. V ďalšom bude používané označenie $\{x_1, x_2, ..., x_n\}$ pre množinu n pozorovaní a $\{X_1, X_2, ..., X_m\}$ pre množinu m vysvetľujúcich premenných. Hodnoty cieľovej premennej Y budú označené ako $y_1, y_2, ..., y_n$. Ak hodnoty vysvetľujúcich premenných sú reálne čísla, tak premenné vytvárajú m–rozmerný priestor R^m. Pozorovania sú bodmi v tomto priestore.

Cieľom výberu premenných je nájsť podmnožinu vysvetľujúcich premenných S, ktorá „optimálne" charakterizuje cieľovú premennú Y. Optimálnosť výberu obvykle znamená najväčšiu štatistickú závislosť cieľovej premennej Y od vybranej podmnožiny. Toto kritérium sa nazýva maximálna závislosť (označené *Max–Dependency*).

V pôvodnej metóde mRMR [8] bola ako miera závislosti náhodných premenných použitá veličina vzájomná informácia (*mutual information*), ktorá je pre množinu spojitých vysvetľujúcich premenných S a cieľovú premennú Y definovaná takto:

$$I(S;Y) = \iint p(S,Y) \log \frac{p(S,Y)}{p(S)p(Y)}\, dS\, dY, \tag{2.1}$$

kde $p(Y)$ je hustota pravdepodobnosti náhodnej premennej Y a $p(S)$, resp. $p(S,Y)$ sú združené hustoty pravdepodobnosti množín náhodných premenných S, resp. $S \cup \{Y\}$ (podľa [7]). V prípade diskrétnych premenných Z a Y možno vzájomnú informáciu vypočítať podľa vzťahu uvedeného v [6] nasledovne:

$$I(Z;Y) = \sum_{z_i} \sum_{y_j} P(Z = z_i, Y = y_j) \log \frac{P(Z = z_i, Y = y_j)}{P(Z = z_i)P(Y = y_j)}, \tag{2.2}$$

pričom pravdepodobnosti P sú potom vypočítané pomocou kontingenčných tabuliek s početnosťou výskytov. V prípade spojitých premenných je výpočet vzájomnej informácie príliš zložitý, a preto sa obvykle zjednodušuje pomocou zdiskrétnenia hodnôt premenných alebo aproximáciou ich hustôt [6].

Ak mierou závislosti je vzájomná informácia, tak účelom výberu premenných je nájsť takú podmnožinu premenných S, od ktorej má premenná Y najväčšiu závislosť, vyjadrenú pomocou vzájomnej informácie podľa [8] takto:

$$\max D(S,Y), \quad D = I(S;Y). \tag{2.3}$$

Keďže kritérium *Max–Dependency* je ťažko implementovateľné [8], alternatíva je vybrať premenné na základe kritéria maximálnej relevancie (*Max–Relevance*). Toto kritérium požaduje, aby bola najväčšia závislosť cieľovej premennej od jednotlivých vybraných premenných. Ide teda o vyhľadávanie premenných vyhovujúcich podmienke:

$$\max D(S,Y), \quad D = \frac{1}{|S|} \sum_{X_i \in S} I(X_i, Y). \tag{2.4}$$

Rovnosť 2.4 aproximuje *Max–Dependency* priemerom hodnôt miery závislosti I cieľovej premennej Y od jednotlivých premenných z množiny S. Jednotlivé vysvetľujúce premenné vybrané týmto spôsobom však môžu byť vysoko redundantné. Ak sú dve premenné navzájom vysoko závislé, predikčná sila množiny vybraných premenných by sa nemala veľmi zmeniť, ak bude jedna z nich z množiny vynechaná [8]. Preto môže byť kvôli výberu navzájom nezávislých premenných pridaná podmienka pre minimálnu redundanciu (*Min–Redundancy*):

$$\min R(S), \quad R = \frac{1}{|S|^2} \sum_{X_i, X_j \in S} I(X_i, X_j). \tag{2.5}$$

Kritérium, ktoré kombinuje obe podmienky, sa označuje ako mRMR (*minimal Redundancy Maximal Relevance*). Nasledovná funkcia uvažuje najjednoduchší spôsob súčasnej optimalizácie relevancie D a redundancie R:

$$\max \Phi(D,R), \quad \Phi = D - R, \quad resp. \quad \Phi = \frac{D}{R} \tag{2.6}$$

Na nájdenie premenných blízkych optimálnym premenným, ktoré definuje funkcia Φ (2.6), môže byť použitá inkrementálna metóda s doprednou selekciou (podľa [11]). V prvom kroku sa vyberie premenná s maximálnou relevanciou voči cieľovej premennej. Ak sú vybrané premenné z množiny S, tak ďalšia premenná je vybraná zo zostávajúcich premenných tak, že sa v nej nadobúda maximum:

- pre rozdiel (MID – *mutual information difference criterion*):

$$\max_{X_i \notin S} \left[I(X_i, Y) - \frac{1}{|S|} \sum_{X_j \in S} I(X_i, X_j) \right] \tag{2.7}$$

- pre podiel (MIQ – *mutual information quotient criterion*):

$$\max_{X_i \notin S} \left[I(X_i, Y) / \frac{1}{|S|} \sum_{X_j \in S} I(X_i, X_j) \right] \tag{2.8}$$

Tento krok sa opakuje dovtedy, kým nie je vybraný požadovaný počet premenných. Poradie výberu premenných zodpovedá ich zoradeniu podľa dôležitosti (*ranking*).

2.2.2 Použitie mRMR s rôznymi mierami závislosti

Metóda mRMR môže byť chápaná ako všeobecný rámec na efektívny výber premenných, ktorý dáva možnosti pre ďalšie sofistikovanejšie a výkonnejšie implementačné schémy. Jednou z možností vylepšovania je výber vhodnej miery závislosti $I(X, Y)$, ktorá je kritickým aspektom v mRMR metodológii [12]. Namiesto vzájomnej informácie je možné použiť dobre známe korelačné koeficienty, napríklad Pearsonov alebo Spearmanov korelačný koeficient [4]. V prípade diskrétnych premenných možno využiť napríklad chí–kvadrát test (*chi–squared test statistic* – χ^2).

Ďalšia, v poslednom čase populárna miera korelácie, ktorá môže poskytnúť zlepšenie metódy mRMR, je korelácia vzdialeností (*distance correlation*) [13]. Pri iných korelačných koeficientoch (napr. Pearsonov alebo Spearmanov) platí, že ak sú premenné navzájom nezávislé, ich korelačný koeficient sa rovná 0. V prípade korelácie vzdialeností platí toto tvrdenie aj opačným smerom [13].

Pre definíciu korelácie vzdialeností je potrebné popísať najskôr kovarianciu vzdialeností. Nech (z_i, y_i) pre $i = 1, 2, \ldots, n$ sú štatistické vzorky dvojice náhodných premenných (Z, Y). Z a Y môžu byť aj viacrozmerné a nemusia mať rovnaký rozmer. Nech $(a_{i,j})$ a $(b_{i,j})$ sú matice vzdialeností $n \times n$ s prvkami:

$$\begin{aligned} a_{i,j} &= L2(z_i, z_j), \quad i, j = 1, 2, \ldots, n, \\ b_{i,j} &= L2(y_i, y_j), \quad i, j = 1, 2, \ldots, n, \end{aligned} \tag{2.9}$$

kde L2 znamená euklidovskú vzdialenosť. Nech $A_{i,j}$ a $B_{i,j}$ pre $i, j = 1, 2, \ldots, n$ sú prvky centrovaných matíc vzdialeností definované takto:

$$\begin{aligned} A_{i,j} &= a_{i,j} - \bar{a}_{i.} - \bar{a}_{.j} + \bar{a}_{..}, \qquad \text{pre } i, j = 1, 2, \ldots, n, \\ B_{i,j} &= b_{i,j} - \bar{b}_{i.} - \bar{b}_{.j} + \bar{b}_{..}, \qquad \text{pre } i, j = 1, 2, \ldots, n, \end{aligned} \tag{2.10}$$

pričom $\bar{a}_{i.}$ je priemerná hodnota i–teho riadku, $\bar{a}_{.j}$ je priemerná hodnota j–teho stĺpca a $\bar{a}_{..}$ je priemerná hodnota zo všetkých prvkov matice.

Potom štvorcová kovariancia vzdialeností je definovaná vzťahom:

$$dCov_n^2(Z, Y) = \frac{1}{n^2} \sum_{i=1}^{n} \sum_{j=1}^{n} A_{i,j} B_{i,j} \tag{2.11}$$

Koreláciu vzdialeností dvoch náhodných premenných možno získať ako podiel:

$$dCor(Z, Y) = \frac{dCov(Z, Y)}{\sqrt{dCov(Z, Z) dCov(Y, Y)}} \tag{2.12}$$

Existuje veľa ďalších definícií pre výpočet korelácie náhodných premenných, v tejto kapitole boli spomenuté len miery použité v práci.

2.3 Ďalšie známe filtre na výber premenných

Metódy typu filter sú založené len na vnútorných charakteristikách dát, ako napríklad vzdialenosť, závislosť a podobne [14]. Niektoré filtre sú univariančné (*univariate*), teda každú vysvetľujúcu premennú hodnotia nezávisle od ostatných. Túto nevýhodu odstraňujú multivariančné (*multivariate*) filtre, ktoré zohľadňujú vzájomné závislosti medzi vysvetľujúcimi premennými. Tie obyčajne vyžadujú viac výpočtových zdrojov [7]. Metóda mRMR je príkladom multivariančného filtra.

2.3.1 Metóda f–score

Metóda f–score [15] je filter určený pre klasifikačné úlohy. Jeho základom je analýza rozptylu vysvetľujúcich premenných (ANOVA). Pre každú vysvetľujúcu premennú X pomocou F–testu testuje hypotézu, že priemer jej hodnôt je pre všetky triedy rovnaký. Pri vykonávaní testu sa vypočíta ANOVA F–test štatistika ako podiel:

$$F = \frac{\textit{odchýlka medzi triedami}}{\textit{odchýlka v rámci tried}}, \tag{2.13}$$

pričom „odchýlka medzi triedami" je:

$$\frac{1}{k-1} . \sum_{i=1}^{k} n_i (\overline{X}_{i.} - \overline{X})^2 \tag{2.14}$$

a „odchýlka v rámci tried" je:

$$\frac{1}{n-k} . \sum_{i=1}^{k} \sum_{j=1}^{n_i} (X_{ij} - \overline{X}_{i.})^2, \tag{2.15}$$

kde X_{ij} je hodnota premennej X pre j–te pozorovanie v i–tej z k tried, $\overline{X}_{i.}$ je priemer hodnôt premennej X pre pozorovania i–tej triedy, $\overline{X}$ priemer všetkých hodnôt premennej X, n_i je počet pozorovaní v i–tej triede a n je celkový počet pozorovaní.

Hodnoty vypočítané pre jednotlivé premenné sa zoradia nerastúco. Čím je hodnota ANOVA F–test štatistiky vyššia, tým je menej pravdepodobná v prípade, ak priemery v jednotlivých triedach sú rovnaké. Výber premenných teda vychádza z predpokladu, že ak je rozdiel medzi priemermi hodnôt premennej v jednotlivých triedach štatisticky významný, potom hodnota vysvetľujúcej premennej X ovplyvňuje hodnotu cieľovej premennej Y.

Keďže vzťah každej vysvetľujúcej premennej voči cieľovej premennej sa analyzuje nezávisle od ostatných, predstavuje táto metóda príklad univariančného filtra.

2.3.2 Metóda ReliefF

ReliefF [16] je metóda selekcie premenných typu filter pre klasifikačné úlohy. Pre každú vysvetľujúcu premennú X vypočíta skóre $W[X]$, ktoré určuje mieru dôležitosti premennej. Na začiatku je skóre každej premennej nulové.

V každej iterácii metóda náhodne vyberie zo súboru dát pozorovanie x_i a k nemu hľadá k najbližších susedov z tej istej triedy (*nearest hits*, h_j, $j = 1, \ldots, k$) a k najbližších susedov z každej inej triedy C (*nearest misses*, $m_j(C)$, $j = 1, \ldots, k$). Pre každú vysvetľujúcu premennú X sa aktualizuje jej skóre v závislosti od jej hodnôt pre pozorovania x_i, h_j a $m_j(C)$. Ak pozorovania x_i a h_j majú rôzne hodnoty premennej X, potom táto premenná oddeľuje pozorovania tej istej triedy, čo je nežiadúce, a preto by malo byť jej skóre znížené. Naproti tomu, ak pozorovania x_i a $m_j(C)$ majú rôzne hodnoty premennej X, tak táto premenná oddeľuje pozorovania rôznych tried, čo je žiadúce, a tak sa skóre premennej X zvýši. Príspevky všetkých h_j a príspevky všetkých $m_j(C)$ pre $j = 1, \ldots, k$ sú spriemerované a vydelené počtom iterácií. Okrem toho, príspevky $m_j(C)$ sú pre každú triedu vážené pravdepodobnosťou triedy zistenou podľa početnosti výskytov v súbore dát.

Skóre každej premennej X sa teda aktualizuje nasledovným spôsobom:

$$W[X] = W[X] - \frac{\sum_{j=1}^{k} \mathit{diff}(X, x_i, h_j)}{t.k} + \\ + \frac{\sum_{C \neq class(x_i)} \left[\frac{P(C)}{1 - P(class(x_i))} \cdot \sum_{j=1}^{k} \mathit{diff}(X, x_i, m_j(C)) \right]}{t.k}, \tag{2.16}$$

kde funkcia $\mathit{diff}(X, x_1, x_2)$ vyjadruje rozdielnosť hodnôt premennej X pre pozorovania x_1 a x_2:

- pre diskrétne premenné:
  $\mathit{diff}(X, x_1, x_2) = 0$, ak $value(X, x_1) = value(X, x_2)$, inak $\mathit{diff}(X, x_1, x_2) = 1$,
- pre spojité premenné:
  $\mathit{diff}(X, x_1, x_2) = \frac{|value(X, x_1) - value(X, x_2)|}{max(X) - min(X)}$.

Funkcia *diff* je použitá aj na výpočet vzdialenosti pre určenie najbližších susedov. Pritom celková vzdialenosť je súčtom za všetky premenné, čo v prípade numerických premenných zodpovedá $\langle 0, 1 \rangle$–normalizácii a manhattanskej metrike.

V metóde ReliefF je uvažovaných t iterácií (podľa vzťahu 2.16), pričom pri každej sa náhodne vyberie jedno pozorovanie. Pre súbory dát s malým počtom pozorovaní je vhodné namiesto náhodného výberu prejsť postupne všetky pozorovania.

Táto metóda predstavuje príklad multivariančného filtra, pretože pri určovaní najbližších susedov používa hodnoty všetkých vysvetľujúcich premenných.

2.4 Metodika porovnávania metód na výber premenných

Metódy na výber premenných možno hodnotiť z viacerých hľadísk, a to z hľadiska úspešnosti výberu relevantných premenných, z hľadiska vplyvu na presnosť predikcie a tiež z hľadiska stability.

2.4.1 Úspešnosť výberu relevantných premenných

Na meranie úspešnosti výberu známych relevantných premenných (*True Features*–TF) slúži podľa [7] index úspešnosti (*index of success*), ktorý odmeňuje výber relevantných premenných a penalizuje zahrnutie irelevantných premenných nasledovne:

$$Suc. = \frac{R_s}{R_t} - \alpha \frac{I_s}{I_t}, \tag{2.17}$$

kde R_s je počet známych relevantných premenných vybraných danou metódou, R_t celkový počet známych relevantných premenných, I_s počet vybraných irelevantných premenných a I_t celkový počet irelevantných premenných v súbore dát. Parameter $\alpha = \min\left\{\frac{1}{2}, \frac{R_t}{I_t}\right\}$ sa používa na zabezpečenie lepšieho hodnotenia v prípade výberu irelevantnej premennej ako v prípade chýbajúcej relevantnej. Ak výber premenných možno interpretovať ako ich zoradenie podľa dôležitosti (*ranking*) a všetky známe relevantné premenné sú vybrané na prvých miestach, bude metóda hodnotená maximálnym indexom $Suc. = 1.0$, teda 100 %.

Počet premenných vo výbere závisí od celkového počtu premenných m v súbore dát a riadi sa nasledovnými pravidlami [7]:

- ak $m < 10$, vyberie sa 75 % premenných,
- ak $10 < m < 75$, vyberie sa 40 % premenných,
- ak $75 < m < 100$, vyberie sa 10 % premenných,
- ak $m > 100$, vyberú sa 3 % premenných.

2.4.2 Vplyv metód výberu premenných na presnosť predikcie

Pri hodnotení vplyvu výberu premenných na presnosť predikcie sa pre získanie objektívnejšieho hodnotenia obvykle používa viacero metód strojového učenia. V práci sú využívané nižšie uvedené metódy popísané napr. v [4].

Bayesovský klasifikátor je jednoduchý pravdepodobnostný model strojového učenia pre riešenie klasifikačných úloh, ktorý je založený na aplikácii Bayesovej vety o podmienenej pravdepodobnosti. Vo verzii *Naive Bayes* sa pridáva „naivný" predpoklad, že vysvetľujúce premenné sú navzájom nezávislé. *Gaussian Naive Bayes* navyše predpokladá, že jednotlivé vysvetľujúce premenné majú Gaussovo normálne rozdelenie. Napriek týmto zjednodušujúcim predpokladom sú pomocou neho dosahované dobré výsledky v mnohých zložitých reálnych situáciách [17].

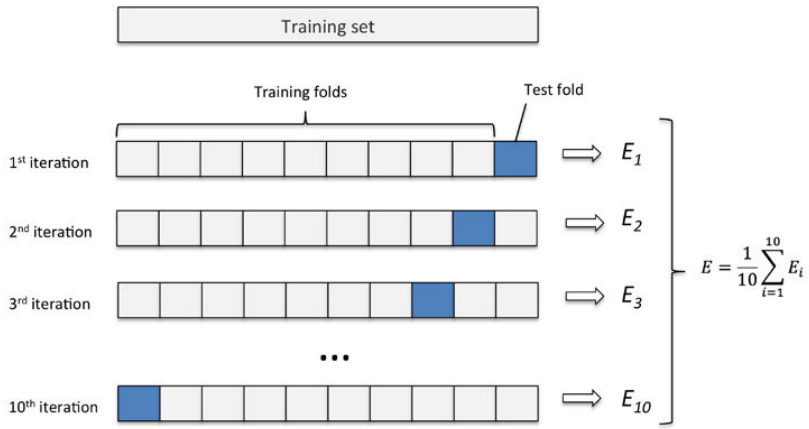
Lineárna regresia (*Linear Regression*) predstavuje základnú metódu pre riešenie regresných úloh. Nad tréningovou množinou vytvorí lineárny model použitím metódy najmenších štvorcov, t. j. nájde koeficienty lineárnej funkcie tak, aby bol minimalizovaný súčet druhých mocnín rozdielov predikovanej a skutočnej hodnoty pre všetky pozorovania z tréningovej množiny. Lineárna funkcia sa použije na predikciu hodnoty spojitej cieľovej premennej pre pozorovania z testovacej množiny.

Metóda podporných vektorov (*Support Vector Machine* – SVM) je určená hlavne pre klasifikačné úlohy. Konštruuje nadrovinu na oddelenie pozorovaní z dvoch rôznych tried. Pritom maximalizuje hraničné pásmo definované podpornými vektormi, ktoré určujú vzdialenosť najbližších pozorovaní z tréningovej množiny od oddeľujúcej nadroviny. Šírku hraničného pásma možno ovplyvniť hyperparametrom C (*cost*) určujúcim penále za nesprávnu klasifikáciu. V prípade lineárne neseparovateľných dát metóda umožňuje použitie mapovania do viacrozmerného priestoru, v ktorom je už ich oddelenie možné. Existuje aj modifikácia metódy pre regresné úlohy.

Rozhodovací strom (*Decision Tree*) je prediktívny model založený na postupnom vytváraní stromu, v ktorom každý vrchol predstavuje delenie tréningovej množiny. V prípade spojitých vysvetľujúcich premenných sa pri každom delení vyberie jedna

premenná a tzv. deliaci bod, ktorý podľa hodnôt vybranej premennej rozdelí pozorovania na dve časti. Premenná a deliaci bod sú vybrané tak, aby rozdelením množiny došlo k maximálnemu zníženiu „nečistoty" podľa určeného kritéria, ako napr. entropia, rozptyl a pod. Listovým vrcholom je priradená predikovaná hodnota určená na základe pozorovaní z tréningovej množiny patriacich tomuto listu. Presnosť predikcie možno zvýšiť metódou náhodných lesov (*Random Forest*), ktorá konštruuje viacero rozhodovacích stromov. Každý z nich je vybudovaný nad náhodne vybranou vzorkou pozorovaní z tréningovej množiny, pričom sa pri každom delení množiny uvažuje iba náhodne vybraná podmnožina vysvetľujúcich premenných. Výsledná predikcia vznikne agregáciou predikcií jednotlivých stromov (väčšinové hlasovanie pri klasifikácii, priemer hodnôt pri regresii).

Základná myšlienka metódy k–NN (*k–Nearest Neighbours*) spočíva v tom, že metóda pre každé pozorovanie z testovacej množiny nájde k jeho najbližších susedov z tréningovej množiny, podľa ktorých určí predikovanú hodnotu. Podrobnejší popis vrátane rozširujúcich variantov metódy je v kapitole 3.2.1.



Obrázok 2 – 1 Krížová validácia

Pri hodnotení vplyvu výberu premenných na presnosť predikcie sa používajú rôzne validačné techniky. Medzi obľúbené patrí krížová validácia (*k–fold cross validation*) – obrázok 2 – 1 z [4]. Pozorovania sú rozdelené na k disjunktných podmnožín približne rovnakej veľkosti. Z nich $k-1$ bude spolu tvoriť tréningovú množinu, nad

ktorou sa vybuduje model. Zostávajúca k–ta bude testovacou množinou, na ktorej sa na základe modelu určí predikcia a odhadne sa jej chyba. Tento postup sa opakuje k–krát, pričom vždy iná podmnožina bude testovacou množinou. Priemer z takto získaných k odhadov chyby predikcie určí výslednú chybu.

Dosiahnutá presnosť predikcie môže byť porovnávaná rôznymi spôsobmi [4]. Obvykle je to na základe počtu percent správne klasifikovaných pozorovaní, ktorý je označovaný ako presnosť (*accuracy*). V prípade binárnej klasifikácie s rôznou početnosťou tried sa na hodnotenie často používa aj miera F_1 *score*, ktorá vystihuje rovnováhu medzi mierami *precision* a *recall* a je definovaná vzorcom:

$$F_1 = 2.\frac{precision.recall}{precision + recall}. \tag{2.18}$$

Precision určuje, koľko z pozitívne predikovaných prípadov je správnych a *recall* vyjadruje, aká časť pozitívnych prípadov bola správne predikovaná (*true positive rate*). Pri regresných úlohách sa používa priemerná štvorcová chyba (MSE – *Mean Squared Error*) alebo priemerná absolútna chyba (MAE – *Mean Absolute Error*).

2.4.3 Stabilita metód výberu premenných

Pri hodnotení metód výberu premenných je okrem ich vplyvu na presnosť predikcie algoritmov strojového učenia dôležitý aj aspekt ich stability, t. j. miera zmeny ich výstupu pri malej zmene vstupných dát. V prehľadovom článku [18] je definovaných a analyzovaných viacero mier stability pre rôzne formy výberu premenných, ktorých výstupom je poradie premenných (*ranking*), skóre premenných (váhy) alebo len vybraná podmnožina premenných.

Pre hodnotenie stability metód výberu premenných na základe podobnosti vybraných podmnožín premenných je možné využiť *Kuncheva index* [19] a vážený *consistency index* CW [20].

Nech m je celkový počet premenných a $S = \{S_1, .., S_k\}$ je systém k podmnožín, ktoré boli získané k–násobným aplikovaním metódy pre výber premenných na náhodné vzorky vytvorené zo základného súboru dát. Ak mohutnosť každej z pod-

množín je $d \leq m$, potom *Kuncheva index* pre systém S je definovaný vzorcom

$$\kappa(S) = \frac{2}{k(k-1)} \sum_{i=1}^{k-1} \sum_{j=i+1}^{k} \frac{|S_i \cap S_j| .m - d^2}{d(m-d)}. \tag{2.19}$$

Definícia indexu CW nevyžaduje rovnakú mohutnosť vyberaných podmnožín. Nech $N = \sum_{i=1}^{k} |S_i|$ je celkový počet výskytov ľubovoľnej premennej v systéme S a ψ_i je počet výskytov i–tej premennej v systéme S. Potom vážený *consistency index* je definovaný:

$$CW(S) = \sum_{i=1}^{m} \frac{\psi_i.(\psi_i - 1)}{N.(k-1)}. \tag{2.20}$$

Obe vyššie definované miery stability závisia od zvoleného počtu vyberaných premenných a nie sú vhodné pre analýzu vývoja stability metód v závislosti od rastúceho počtu vyberaných premenných. Na tento účel je však možné využiť relatívny vážený *consistency index* CW_{rel} definovaný úpravou indexu CW na základe jeho možnej minimálnej a maximálnej hodnoty. Ak $D = N \mod (m)$ a $H = N \mod (k)$, tak podľa [21]:

$$CW_{rel}(S) = \frac{m(N - D + \sum_{i=1}^{m} \psi_i.(\psi_i - 1)) - N^2 + D^2}{m(H^2 + k(N-H) - D) - N^2 + D^2}. \tag{2.21}$$

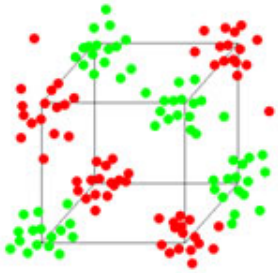
2.5 Súbory dát na overovanie metód výberu premenných

Hlavným cieľom hodnotenia efektívnosti metód na výber premenných je ich testovanie na reálnych dátach. Reálne dáta však nesú so sebou mnoho komplikácií, ako sú veľké množstvo irelevantných a redundantných premenných, malý počet pozorovaní v pomere k počtu premenných, nelineárnosť dát, prítomnosť šumu a podobne. Preto sa na overovanie metód často používajú najprv umelo skonštruované súbory dát.

2.5.1 Syntetické dáta

Výhodou použitia syntetických dát pri testovaní metód na výber premenných je možnosť riadenia experimentov zmenou parametrov pri generovaní súborov dát a známa množina relevantných premenných – TF. To umožňuje ľahko vyhodnotiť, nakoľko bola daná metóda pri ich identifikovaní úspešná.

V práci budú použité syntetické súbory dát so spojitými vysvetľujúcimi premennými, a to súbor dát Madelon pre binárnu klasifikačnú úlohu a súbory dát LinReg a Friedman pre regresnú úlohu.



Obrázok 2 – 2 Súbor dát Madelon v trojrozmernom priestore

Súbor dát Madelon je skonštruovaný tak, že vo vybraných vrcholoch hyperkocky v k–rozmernom priestore, kde k je počet TF, sú umiestnené zhluky náhodne vygenerovaných bodov s normálnym rozdelením so štandardnou odchýlkou 1. Všetkým bodom jedného zhluku je priradená rovnaká trieda, pričom body jednej polovice zhlukov sú triedy 0 a druhej polovice triedy 1 (obrázok 2 – 2 z [22]). Dĺžka hrany hyperkocky je štandardne 2, pre lepšie oddelenie zhlukov je možné voliť jej násobky. V práci je použitá dĺžka 2*2 a počet zhlukov 4 pre každú triedu.

V lineárnom regresnom súbore dát LinReg je cieľová premenná lineárnou kombináciou k nezávislých relevantných premenných:

$$Y = \alpha_0 + \alpha_1 X_1 + \alpha_2 X_2 + \ldots + \alpha_k X_k, \tag{2.22}$$

pričom koeficienty $\alpha_0, \alpha_1, \alpha_2, \ldots, \alpha_k$ sú generované náhodne.

V regresnom súbore dát Friedman je závislosť cieľovej premennej od vysvetľujúcich premenných nelineárna. Množina vysvetľujúcich premenných obsahuje 5 TF s rovnomerným rozdelením na intervale $\langle 0, 1 \rangle$ a cieľová premenná je určená vzťahom:

$$Y = 10sin(\pi.X_0.X_1) + 20(X_2 - 0.5)^2 + 10X_3 + 5X_4. \tag{2.23}$$

Na výpočet cieľovej premennej je použitých len 5 TF, ostatné premenné sú nezávislé od Y a sú náhodne vygenerované s Gaussovým normálnym rozdelením.

2.5.2 Reálne dáta

Jednou z oblastí reálneho života, kde boli v posledných rokoch úspešne aplikované metódy na výber premenných, je bioinformatika. Súbory dát pochádzajúce z DNA čipov (*DNA microarrays*) zhromažďujú veľké množstvo údajov zo vzoriek tkanív a buniek slúžiacich na odhaľovanie zmien v sekvencii DNA, ktoré by mohli byť užitočné pri diagnostikovaní chorôb, rozlišovaní typov nádorov a podobne. Ide o vysokodimenzionálne súbory dát s malým počtom pozorovaní. Obyčajne majú rádovo 10 000 premenných a 100 pozorovaní.

Tabuľka 2 – 1 Charakteristiky reálnych súborov dát

Názov [Zdroj]	Popis	Pacienti	Gény	Trieda 0	Trieda 1
Alon [23]	Nádor hrubého čreva	62	2 000	40	22
Pomeroy [24]	Nádor CNS	60	7 128	39	21
Golub [25]	Leukémia	72	7 129	47	25
Gordon [26]	Nádor pľúc	181	12 533	31	150
Singh [27]	Nádor prostaty	102	12 600	50	52
Tian [28]	Myelóm	173	12 625	36	137
Chin [29]	Nádor prsníka	118	22 215	43	75
Chowdaryi [30]	Nádor prsníka	104	22 283	62	42
Burczynski [31]	Crohnova choroba	127	22 283	85	42

Na overovanie, ako sa rôzne metódy na výber premenných správajú pri riešení úloh z reálneho sveta, bude v práci použitých 9 súborov dát pre binárnu klasifikačnú úlohu pochádzajúcich z DNA čipov. Súbory dát sú popísané v tabuľke 2 – 1, kde jednotliví pacienti predstavujú pozorovania a gény premenné. Interpretácia tried 0 a 1 je rôzna, napríklad či je tkanivo zdravé alebo choré, či bola liečba úspešná alebo nie (Pomeroy) alebo sú rozlíšené typy nádorov (Gordon).

3 Návrh a implementácia riešenia

V súlade s cieľmi uvedenými v kapitole 1 je časť práce venovaná analýze a implementácii metódy mRMR popísanej v kapitole 2.2. Kapitola 3.1 obsahuje teoretické výsledky týkajúce algoritmu mRMR, na základe ktorých bolo navrhnuté a implementované zovšeobecnenie algoritmu mRMR s doplnením podpory pre rôzne miery závislosti. Zároveň bol analyzovaný vzťah algoritmu mRMR a pažravého algoritmu pre *Max–Dependency*.

Hlavnú časť práce predstavuje návrh a implementácia novej metódy na výber premenných vhodnej na použitie na dátach veľkých rozmerov. Metóda rozpracovaná v kapitole 3.2 používa modifikovanú metódu k–NN (*k–Nearest Neighbours*) a metódu najväčšieho spádu (*gradient descent*) ako iteratívny optimalizačný algoritmus na hľadanie minima funkcie. V tejto časti práce sú popísané teoretické východiská, matematický základ navrhnutej metódy a spôsob jej parametrizácie, ktorý umožňuje v metóde použiť rôzne metriky a funkcie pre výpočet chyby predikcie.

Navrhnuté algoritmy sú implementované v jazyku Python verzia 3.6 [32]. Implementácia metódy mRMR bola overená voči výsledkom uvedeným v literatúre. Výsledky novej metódy pre rôzne nastavenia parametrov získané aplikáciou na vybrané syntetické a reálne súbory dát sú porovnané s výsledkami známych metód a vyhodnotené v kapitole 4. Táto kapitola obsahuje aj porovnanie stability jednotlivých metód. Popis implementácie navrhnutých algoritmov je obsahom Prílohy C.

3.1 Výber premenných použitím metódy mRMR

Jedným z teoretických výsledkov, ktoré vznikli pri štúdiu metódy mRMR, je dôkaz ekvivalencie inkrementálneho algoritmu mRMR uvedeného v [11] s pažravým algoritmom na maximalizáciu odvodenej účelovej funkcie (vzťah 3.1). Z jej definície vychádza návrh zovšeobecnenia algoritmu mRMR, v ktorom je navyše možné použiť rôzne miery závislosti. Ďalej bola pomocou syntetických aj reálnych súborov dát skúmaná ekvivalencia algoritmu mRMR a pažravého algoritmu pre *Max–Dependency*.

3.1.1 Zovšeobecnenie algoritmu mRMR

Možno si všimnúť, že definícia 2.7, resp. 2.8 inkrementálneho kroku v algoritme mRMR podľa [11] je odlišná od definície 2.6 funkcie Φ. Neobsahuje napríklad členy $I(X_i, X_j)$ pre $i = j$. Aj experimentálne bolo overené, že výsledky získané pomocou algoritmu mRMR založeného na inkrementálnej definícii sú odlišné od výsledkov pažravého algoritmu pre maximalizáciu funkcie Φ.

Nasledujúca veta uvádza funkciu Φ', o ktorej je matematickou indukciou dokázané, že pažravý algoritmus pre jej maximalizáciu vyberie rovnakú množinu premenných ako algoritmus mRMR v MID verzii (*mutual information difference criterion*) s inkrementálnou definíciou popísaný v [11].

Veta: Nech je funkcia Φ' definovaná nasledovne:

$$\Phi'(S) = \frac{1}{|S|} \sum_{X_i \in S} I(X_i, Y) - \frac{1}{2} . \frac{1}{|S|\,(|S|-1)} \sum_{\substack{X_i, X_j \in S, \\ i \neq j}} I(X_i, X_j) \tag{3.1}$$

Inkrementálna definícia algoritmu mRMR v MID verzii je ekvivalentná s pažravým algoritmom pre maximalizáciu funkcie Φ'.

Dôkaz:

Je zrejmé, že v prvom kroku inkrementálna definícia mRMR algoritmu rovnako ako pažravý algoritmus pre maximalizáciu funkcie Φ' vyberie premennú s maximálnou relevanciou voči cieľovej premennej. Ukážeme, že ak oba algoritmy v prvých m krokoch vybrali rovnaké premenné tvoriace množinu S, tak v kroku $m + 1$ vyberú rovnakú premennú X.

Nech $S' = S \cup \{X\}$. Množina S' má $m + 1$ prvkov. Hodnota optimalizovanej funkcie po pridaní premennej X je:

$$\Phi'(S') = \frac{1}{m+1} \sum_{X_i \in S'} I(X_i, Y) - \frac{1}{2(m+1)m} \sum_{\substack{X_i, X_j \in S', \\ i \neq j}} I(X_i, X_j) \tag{3.2}$$

Pažravý algoritmus pre maximalizáciu funkcie Φ' vyberie premennú X tak, že maximalizuje hodnotu funkcie Φ'. Jednotlivé členy vo výraze 3.2 možno rozpísať

nasledovne:

$$\frac{1}{m+1}\sum_{X_i\in S'} I(X_i,Y) = \frac{1}{m+1}\sum_{X_i\in S} I(X_i,Y) + \frac{1}{m+1}I(X,Y) \tag{3.3}$$

$$\begin{aligned}&\frac{1}{2(m+1)m}\sum_{\substack{X_i,X_j\in S',\\ i\neq j}} I(X_i,X_j) = \\ &\frac{1}{2(m+1)m}\sum_{\substack{X_i,X_j\in S,\\ i\neq j}} I(X_i,X_j) + \frac{1}{2(m+1)m}.2\sum_{X_i\in S} I(X_i,X)\end{aligned} \tag{3.4}$$

Po dosadení:

$$\begin{aligned}\Phi'(S') =& \frac{1}{m+1}\sum_{X_i\in S} I(X_i,Y) + \frac{1}{m+1}I(X,Y) \\ &-\frac{1}{2(m+1)m}\sum_{\substack{X_i,X_j\in S,\\ i\neq j}} I(X_i,X_j) - \frac{1}{(m+1)m}\sum_{X_i\in S} I(X_i,X)\end{aligned} \tag{3.5}$$

Tie členy vo výraze 3.5, ktoré neobsahujú premennú X, nemajú vplyv na maximalizáciu $\Phi'(S')$ vzhľadom na výber premennej X. Preto pre výber premennej X stačí maximalizovať výraz:

$$\frac{1}{m+1}I(X,Y) - \frac{1}{(m+1)m}\sum_{X_i\in S} I(X_i,X) \tag{3.6}$$

Výraz $\frac{1}{m+1}$ možno vybrať pred zátvorku a keďže ide o kladné číslo, je to ekvivalentné maximalizácii:

$$\max_{X\notin S}\left[I(X,Y) - \frac{1}{m}\sum_{X_i\in S} I(X_i,X)\right] \tag{3.7}$$

Výraz 3.7 je ekvivalentný inkrementálnemu kroku mRMR algoritmu v MID verzii, čiže pažravý algoritmus pre funkciu Φ' vyberie rovnakú premennú X ako mRMR algoritmus.

■

Funkcia Φ' je rozdielom priemernej korelácie premenných voči cieľovej premennej a priemernej korelácie medzi rôznymi premennými v množine S navzájom, ktorá je započítaná s váhou $\frac{1}{2}$. Možno ju zovšeobecniť nasledovne:

$$\Phi'(S) = \frac{1}{|S|}\sum_{X_i\in S} I(X_i,Y) - \lambda.\frac{1}{|S|.(|S|-1)}\sum_{\substack{X_i,X_j\in S,\\ i\neq j}} I(X_i,X_j) \tag{3.8}$$

V práci bol v jazyku Python implementovaný zovšeobecnený algoritmus mRMR založený na maximalizácii funkcie Φ' vo všeobecnom tvare podľa vzťahu 3.8. Okrem parametra λ je možné voliť aj mieru závislosti I, pričom štandardne sa používa vzájomná informácia. Pomocou vlastnej implementácie je realizovaný výpočet korelácie vzdialeností popísanej v kapitole 2.2.2. Podrobný popis všetkých parametrov a ďalšie implementačné detaily sa nachádzajú v Prílohe C.

Tabuľka 3 – 1 Súbory dát pre testovanie implementácie algoritmu mRMR

Názov	Pozorovania	Premenné
test_lung_s3	73	325
test_leukemia_s3	70	7 070
test_lymphoma_s3	96	4 026
test_colon_s3	62	2 000
test_nci9_s3	60	9 712

Za účelom overenia funkčnosti riešenia bol z [33] stiahnutý spustiteľný súbor `mrmr_win32.exe`, ktorý je implementáciou algoritmu mRMR podľa vzťahov 2.7 a 2.8 v jazyku C v skompilovanom tvare. Výsledky potvrdili, že implementované zovšeobecnené riešenie s nastavením parametra $\lambda = 0,5$ dáva na skúšobných súboroch dát uvedených v tabuľke 3 – 1 z [33] rovnaké výsledky ako použitý spustiteľný súbor. V implementácii v jazyku Python bolo však potrebné výraz pre výpočet vzájomnej informácie deliť hodnotou $\ln 2$, aby boli hodnoty rovnaké ako v algoritme implementovanom v jazyku C, ktorý používa na výpočet vzájomnej informácie logaritmus pri základe 2.

3.1.2 Algoritmus mRMR a *Max–Dependency*

Ako bolo uvedené v kapitole 2.2.1, implementácia algoritmu pre *Max–Dependency* je výpočtovo veľmi náročná, hlavne v prípade spojitých vysvetľujúcich premenných. Algoritmus mRMR predstavuje výpočtovo menej náročnú alternatívu. V [8] je uvedené z toho pohľadu dôležité tvrdenie o ekvivalencii mRMR algoritmu:

Veta:

Pri použití pažravého algoritmu sú mRMR a *Max–Dependency* ekvivalentné.

V [6] sa nachádza príklad dvoch vysvetľujúcich premenných, pre ktoré platí, že I(X_1,Y) = 0 a I(X_2,Y) = 0, a pritom táto dvojica premenných umožňuje presne rozlíšiť hodnoty cieľovej premennej. Tento príklad viedol k úvahám, či neexistuje kontrapríklad na spomínanú vetu uvedenú v [8].

Tabuľka 3 – 2 Súbor dát pre porovnanie algoritmov mRMR a Max–Dependency

	X_1	X_2	X_3	Y
1	0	0	0	0
2	0	0	0	0
3	0	1	0	1
4	0	1	0	1
5	1	0	0	1
6	1	0	0	1
7	1	1	0	0
8	2	1	0	0

	X_1	X_2	X_3	Y
9	0	0	1	0
10	0	0	1	0
11	0	1	1	1
12	0	1	1	1
13	1	0	1	1
14	1	0	1	1
15	1	1	1	0
16	2	1	1	0

V tabuľke 3 – 2 je popísaný súbor dát, ktorý má 16 pozorovaní. Premenná X_1 nadobúda 3 hodnoty, všetky ostatné premenné nadobúdajú 2 hodnoty. Možno si všimnúť, že prvých 8 pozorovaní z pohľadu premenných X_1, X_2 a Y je presne rovnakých ako druhých 8 pozorovaní. Líšia sa len tým, že prvých 8 pozorovaní má $X_3 = 0$ a druhých 8 pozorovaní má $X_3 = 1$. Dá sa overiť, že z toho vyplynie:

I(X_1,X_3) = 0, I(X_2,X_3) = 0.

Taktiež možno ľahko overiť, že

I(X_2,Y) = 0, I(X_3,Y)) = 0.

Hodnota I(X_1,Y) je kladná (približne 0,156), a preto v prvom kroku mRMR aj pažravý algoritmus pre *Max–Dependency* vyberú premennú X_1. (Pri výpočte sa používa logaritmus pri základe 2.) V druhom kroku mRMR vypočíta pre kandidátov:

X_2: I(X_2,Y) - I(X_2,X_1) = - I(X_2,X_1),

X_3: I(X_3,Y) - I(X_3,X_1) = 0 - 0 = 0.

Keďže hodnota I(X_2,X_1) je kladná (taktiež približne 0,156), tak maximum sa dosahuje pre X_3 a mRMR vyberie X_3 ako druhú premennú. Tento výber však nie je dobrý, nakoľko doplnenie X_3 neumožňuje lepšie rozlíšiť hodnoty cieľovej premennej.

Možno ukázať, že oproti tomu pažravý algoritmus pre *Max–Dependency* vyberie ako druhú premennú X_2. Na výber druhej zo zostávajúcich premenných X_2, X_3 týmto algoritmom treba určiť vzájomnú informáciu množín $\{X_1, X_2\}$ a $\{X_1, X_3\}$ s cieľovou premennou Y. Podľa definície vzájomnej informácie pre diskrétny prípad 2.2 uvedenej v kapitole 2.2.1 možno vypočítať hodnotu $I(\{X_1, X_3\}; Y)$ tak, že za Z sa dosadí (X_1, X_3) a vytvorí sa príslušná kontingenčná tabuľka.

Z pravej časti tabuľky 3 – 3 vyplýva, že I($\{X_1, X_3\}$, Y) = I(X_1,Y) ≅ 0,156.

Tabuľka 3 – 3 Kontingenčné tabuľky pre výpočet vzájomnej informácie

(X_1, X_2)	$Y = 0$	$Y = 1$	Σ
(0, 0)	4	0	4
(1, 0)	0	4	4
(2, 0)	0	0	0
(0, 1)	0	4	4
(1, 1)	2	0	2
(2, 1)	2	0	2
Σ	8	8	16

(X_1, X_3)	$Y = 0$	$Y = 1$	Σ
(0, 0)	2	2	4
(1, 0)	1	2	3
(2, 0)	1	0	1
(0, 1)	2	2	4
(1, 1)	1	2	3
(2, 1)	1	0	1
Σ	8	8	16

Podobne možno vypočítať aj vzájomnú informáciu množiny $\{X_1, X_2\}$ s cieľovou premennou Y (ľavá časť tabuľky 3 – 3). Dvojica X_1, X_2 presne určuje hodnotu cieľovej premennej a platí:

I($\{X_1, X_2\}$, Y) = 1 > I($\{X_1, X_3\}$, Y) = I(X_1,Y) ≅ 0,156.

Pre uvedený súbor dát teda pažravý algoritmus pre *Max–Dependency* vybral inú dvojicu premenných s oveľa vyššiou hodnotou závislosti ako algoritmus mRMR.

Bolo overené, že takáto situácia nenastáva len u umelo vytvorených súborov dát, ale možno ju pozorovať aj u súborov dát z reálneho prostredia. Pri riešení binárnej klasifikačnej úlohy pre súbor Golub popísaný v kapitole 2.5.2 bolo vykonané zdiskrétnenie spojitých vysvetľujúcich premenných na 3 hodnoty spôsobom popísaným

v [11] s použitím hraníc $mean - std, mean + std$, kde $mean$ je priemerná hodnota a std štandardná odchýlka jednotlivých premenných. Následne bolo zistené, že algoritmus pre *Max–Dependency* vybral celkovo štyri premenné v poradí v3319, v803, v6942 a v48 s hodnotou vzájomnej informácie voči cieľovej premennej 0.93. Naproti tomu algoritmus mRMR vybral v prvých štyroch krokoch premenné v3319, v803, v4846 a v1828, teda už na treťom kroku sa výber líši. Zároveň vzájomná informácia vybranej množiny voči cieľovej premennej je len 0.84, to znamená, že cieľová premenná má vyššiu závislosť od množiny premenných vybranej algoritmom *Max–Dependency* ako od množiny premenných vybranej algoritmom mRMR.

Navyše algoritmus pre *Max–Dependency* vybral pre súbor dát Golub iba spomínané štyri premenné. Pri pridávaní ďalších sa hodnota vzájomnej informácie množiny vybraných premenných voči cieľovej premennej už nezväčšovala. Podobná situácia nastala v prípade súboru dát Gordon popísaného v kapitole 2.5.2, kde boli algoritmom vybrané dokonca len tri premenné, a to v12113, v3333 a v3249. Z kontingenčnej tabuľky 3 – 4 vidno, že z viac ako 12 500 premenných stačí vybrať tri premenné, na základe ktorých už možno presne určiť hodnotu cieľovej premennej Y.

Tabuľka 3 – 4 Kontingenčná tabuľka pre súbor dát Gordon

(v12113, v3333, v3249)	$Y = 0$	$Y = 1$	Σ
(0, 1, 0)	17	0	17
(0, 1, 2)	4	0	4
(0, 2, 0)	9	0	9
(0, 2, 2)	0	7	7
(1, 2, 0)	0	1	1
(1, 2, 1)	0	5	5
(1, 2, 2)	0	11	11
(2, 1, 2)	1	0	1
(2, 2, 0)	0	1	1
(2, 2, 1)	0	23	23
(2, 2, 2)	0	102	102
Σ	31	150	181

3.2 Návrh metódy výberu premenných na báze k–NN

V tejto kapitole je popísaný návrh a implementácia novej metódy na výber premenných (WkNN–FS), ktorá vychádza z metódy strojového učenia k najbližších susedov (*k–Nearest Neighbours*) a používa metódu najväčšieho spádu (*gradient descent*) ako iteratívny optimalizačný algoritmus pre hľadanie minima funkcie.

3.2.1 Metóda váženého k–NN

Algoritmus k–NN (k najbližších susedov) [4] je metóda strojového učenia, ktorá sa používa pre klasifikačné aj pre regresné úlohy. Algoritmus predikuje hodnotu cieľovej premennej pre pozorovanie z testovacej množiny na základe hodnôt cieľovej premennej jeho k najbližších susedov vybraných z tréningovej množiny. Parametre algoritmu sú prirodzené číslo k a definícia vzdialenosti. V prípade klasifikačných úloh je hodnotou cieľovej premennej označenie triedy, najčastejšie pomocou hodnôt 0, 1, atď. Predikcia sa robí podľa princípu väčšinového hlasovania, t. j. predikovaná hodnota bude rovná najčastejšie sa vyskytujúcej triede medzi vybranými susedmi. Pri regresných úlohách je predpovedaná hodnota reálne číslo, ktoré sa obvykle určuje ako aritmetický priemer hodnôt cieľovej premennej vybraných susedov.

Pri návrhu novej metódy na výber premenných bude uvažovaná regresná úloha. Nech $\{X_1, X_2, ..., X_m\}$ je množina m numerických vysvetľujúcich premenných, nech $\{x_1, x_2, ..., x_n\}$ je množina n pozorovaní a nech Y je závislá premenná (numerická) s hodnotami $y_1, y_2, ..., y_n$. Ak x_i je prvok testovacej množiny a N_i^k je množina indexov k najbližších susedov pozorovania x_i, možno predikciu p_i cieľovej premennej vyjadriť nasledovne:

$$p_i = \frac{1}{k} \sum_{j \in N_i^k} y_j. \tag{3.9}$$

Metóda k–NN je založená na meraní podobnosti objektov. Algoritmus hľadá k pozorovaní z tréningovej množiny v istom zmysle najviac podobných sledovanému pozorovaniu z testovacej množiny. Na vyjadrenie podobnosti používa vzdialenosť objektov. Vzdialenosť je chápaná ako metrika, teda funkcia, ktorá dvom objektom

priradí nezáporné reálne číslo, pričom musí spĺňať ďalšie vlastnosti popísané v [1]. Spôsob definície vzdialenosti použitej v algoritme k–NN závisí od typu premenných v súbore dát. V prípade spojitých premenných sa obvykle používa euklidovská vzdialenosť. Popis definícií vzdialenosti pre rôzne typy dát možno nájsť v [34].

V základnej metóde k–NN má každý z k najbližších susedov rovnaký vplyv na predikovanú hodnotu cieľovej premennej. Na základe myšlienky, že pozorovanie, ktoré je bližšie k sledovanému, je mu viac podobné, a preto by malo mať väčší vplyv na odhad jeho hodnoty cieľovej premennej [35], možno metódu prirodzene rozšíriť tak, že namiesto aritmetického priemeru hodnôt cieľovej premennej vybraných susedov sa využije vážený priemer, ktorý zvýhodňuje bližšie pozorovania. Toto rozšírenie sa nazýva k–NN vážené podľa vzdialenosti (*distance–weighted* k–NN). Váhy jednotlivých susedov vzniknú na základe transformácie vzdialenosti, ktorá je realizovaná hodnotiacou funkciou w. Potom možno vzťah 3.9 pre predikciu p_i nahradiť vzťahom:

$$p_i = \frac{1}{\sum_{j \in N_i^k} w(d_{ij})} \sum_{j \in N_i^k} w(d_{ij}).y_j, \tag{3.10}$$

pričom d_{ij} označuje vzdialenosť pozorovaní x_i a x_j a $w(d_{ij})$ jej hodnotu po transformácii pomocou funkcie w. Pritom hodnotiaca funkcia je zvolená tak, aby mali bližší susedia väčší vplyv a vzdialenejší menší vplyv na predikovanú hodnotu. Príkladom hodnotiacej funkcie môže byť funckia $w(d) = \frac{1}{d^2}$ [3].

Ďalším rozšírením metódy k–NN je atribútovo vážené k–NN (*attribute–weighted* k–NN) [3]. Každý atribút, t. j. vysvetľujúca premenná, je ohodnotený váhou, ktorá určuje, nakoľko je daný atribút užitočný pre určenie hodnoty cieľovej premennej. V prípade euklidovskej metriky váženie atribútov zodpovedá naťahovaniu osí v euklidovskom priestore. Osi, ktoré zodpovedajú menej relevantným premenným, sa skracujú a osi zodpovedajúce viac relevantným premenným sa predlžujú. Naťahovanie osí možno vyjadriť pomocou Hadamardovho (Schurovho) súčinu [36]:

Ak $\boldsymbol{v} = (v_1, v_2, \ldots..., v_m)$ je vektor váh jednotlivých vysvetľujúcich premenných a $x_i = (x_{i1}, x_{i2}, \ldots..., x_{im})$ je ľubovoľné pozorovanie, potom Hadamardov súčin vek-

tora $\boldsymbol{v}$ s pozorovaním x_i je definovaný nasledovne:

$$\boldsymbol{v} \circ x_i = (v_1.x_{i1}, v_2.x_{i2}, \ldots ..., v_m.x_{im}). \tag{3.11}$$

3.2.2 Návrh úpravy metódy k–NN

Navrhovaná metóda na výber premenných vychádza z metódy k–NN, pričom využíva kombináciu atribútovo váženého k–NN a k–NN váženého podľa vzdialenosti. Vychádza z toho, že hodnotu cieľovej premennej by mali najlepšie určovať najbližší susedia, pričom vzdialenosť susedov by mala najviac závisieť od relevantných premenných.

Transformácia vzdialenosti bude realizovaná funkciou $w : \Re_0^+ \rightarrow \langle 0, 1\rangle$ s nasledovnými vlastnosťami:

- $w(0) = 1$ – nulová vzdialenosť má hodnotu 1,
- funkcia w je klesajúca – čím väčšia vzdialenosť, tým menšia hodnota funkcie,
- $\lim_{d\rightarrow\infty} w(d) = 0$ – ak sa vzdialenosť zväčšuje do nekonečna, hodnota funkcie sa blíži k nule.

Príkladom funkcie s požadovanými vlastnosťami je funkcia $w(d) = e^{-d^2}$. Takéto funkcie transformujú vzdialenosť na mieru podobnosti (*similarity*). Hodnotiaca funkcia vzdialenosti sa nazýva aj *kernel*.

Zároveň bude každá vysvetľujúca premenná ohodnotená váhou, ktorá odráža, aký vplyv má daná premenná na hodnotu cieľovej premennej. Ak váha môže nadobúdať len hodnotu nula alebo jedna, tak určuje, či zodpovedajúcu vysvetľujúcu premennú treba uvažovať alebo nie, teda ide o výber premenných. Vo všeobecnosti však môže byť váha nezáporné reálne číslo určujúce relevanciu danej premennej.

V navrhovanej metóde sa váhy premenných použijú pri výpočte vzdialeností pozorovaní. Ak $\boldsymbol{v}$ je vektor váh, d je metrika a x_i a x_j je ľubovoľná dvojica pozorovaní, potom vzdialenosť pozorovaní x_i a x_j vážená vektorom $\boldsymbol{v}$ je definovaná nasledovne:

$$d_{ij}(\mathbf{v}) = d(\mathbf{v} \circ x_i, \mathbf{v} \circ x_j). \tag{3.12}$$

Na základe takto definovanej vzdialenosti budú vyberaní najbližší susedia. Pri zmene váh premenných však môže dôjsť aj k zmene množiny k najbližších susedov. Tým by sa predikovaná hodnota definovaná vzťahom 3.10 mohla stať nespojitou, a preto budú v navrhovanej metóde uvažované namiesto k najbližších susedov všetky ostatné pozorovania. Pri predikcii bude teda použitá celá tréningová množina s uplatnením princípu *leave–one–out* (LOO) [4].

Predikovaná hodnota $p_i(\boldsymbol{v})$ cieľovej premennej Y pre i–te pozorovanie v závislosti od vektora váh $\boldsymbol{v}$ s použitím hodnotiacej funkcie vzdialenosti w sa vypočíta ako vážený priemer hodnôt cieľovej premennej všetkých ostatných pozorovaní:

$$p_i(\mathbf{v}) = \frac{1}{\sum_{k \neq i} w(d_{ik}(\mathbf{v}))} \sum_{j \neq i} w(d_{ij}(\mathbf{v})).y_j. \tag{3.13}$$

Presnosť predikcie závislej premennej Y sa pri regresných úlohách meria pomocou veľkosti odchýlky predikovanej hodnoty od skutočnej hodnoty [2]. Na vyjadrenie chyby pre jedno pozorovanie sa používa funkcia chyby (*loss function*), čo je funkcia $l : \Re \times \Re \rightarrow \Re_0^+$. Ako príklad možno uviesť funkciu štvorcovej chyby (*square loss function*, resp. *squared error*) definovanú takto:

$$l(y_i, p_i(\mathbf{v})) = (y_i - p_i(\mathbf{v}))^2. \tag{3.14}$$

Pre celý súbor dát je potom definovaná účelová funkcia (*cost function*) ako priemerná chyba:

$$C(\mathbf{v}) = \frac{1}{n} \sum_{i=1}^{n} l(y_i, p_i(\mathbf{v})). \tag{3.15}$$

Vytvorený model definovaný vzťahom 3.13 umožňuje riešenie regresných úloh. Jeho chyba je vyjadrená ako funkcia váh premenných a cieľom bude nájsť taký vektor váh, aby model dával čo najmenšiu chybu. To znamená, že je potrebné riešiť optimalizačnú úlohu:

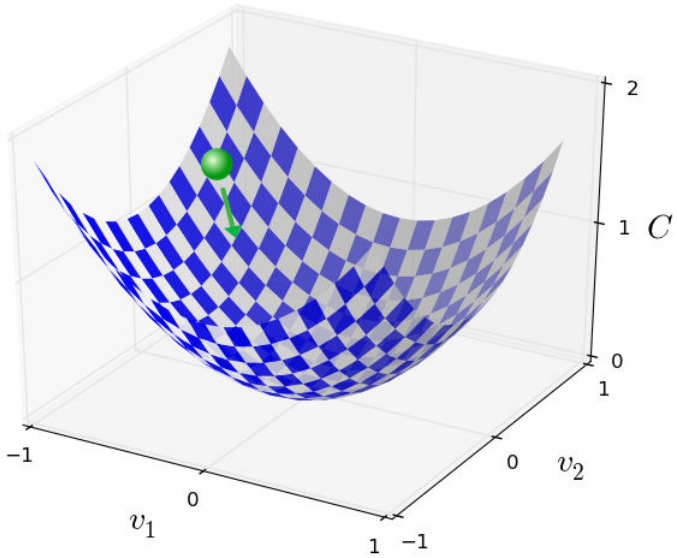
$$\mathbf{v}^{opt} = argmin_{\mathbf{v} \in \Re^m} C(\mathbf{v}). \tag{3.16}$$

V ďalšej kapitole bude popísaný spôsob hľadania optimálnych váh riešením vyššie uvedenej optimalizačnej úlohy.

3.2.3 Metóda najväčšieho spádu

Cieľom navrhovanej metódy na výber premenných je nájsť taký vektor váh $\boldsymbol{v}$, pre ktorý je priemerná chyba predikovanej hodnoty závislej premennej Y oproti skutočnej hodnote minimálna. Účelová funkcia je vo všeobecnosti funkcia $C : \Re^m \rightarrow \Re_0^+$. Ak je spojitá a diferencovateľná, možno na jej minimalizáciu použiť iteratívny optimalizačný algoritmus známy pod názvom metóda najväčšieho spádu (*gradient descent* – GD), ktorý na základe gradientu hľadá lokálne minimum funkcie.

Princíp hľadania minima funkcie pomocou metódy najväčšieho spádu [37] možno vysvetliť na príklade minimalizácie funkcie C dvoch premenných v_1 a v_2. Nech graf funkcie má tvar údolia. Hľadanie jej minima môže byť simulované pohybom loptičky, ktorá sa gúľa do údolia. Štartovací bod bude zvolený náhodne. Parciálne derivácie funkcie C v danom bode popisujú tvar údolia v okolí daného bodu, a tým určujú smer ďalšieho pohybu loptičky (obrázok 3 – 1 z [37]).



Obrázok 3 – 1 Metóda najväčšieho spádu

Ak sa loptička pohne o hodnotu Δv_1 v smere v_1 a o hodnotu Δv_2 v smere v_2, tak sa hodnota funkcie $C(\boldsymbol{v})$ zmení takto:

$$\Delta C(\boldsymbol{v}) \approx \frac{\partial C}{\partial v_1}\Delta v_1 + \frac{\partial C}{\partial v_2}\Delta v_2, \tag{3.17}$$

kde $\frac{\partial C}{\partial v_1}$ a $\frac{\partial C}{\partial v_2}$ sú parciálne derivácie funkcie C podľa premenných v_1 a v_2. Úlohou je nájsť vektor $\Delta \boldsymbol{v} = (\Delta v_1, \Delta v_2)$ taký, aby bolo $\Delta C(\boldsymbol{v})$ záporné. Vtedy sa bude hodnota $C(\boldsymbol{v})$ zmenšovať.

Vektor parciálnych derivácií funkcie C podľa premenných v_1 a v_2 sa nazýva gradient funkcie C a označuje sa

$$\nabla C \equiv \left(\frac{\partial C}{\partial v_1}, \frac{\partial C}{\partial v_2} \right)^T. \tag{3.18}$$

Dosadením možno ľahko ukázať, že ak sa za vektor $\Delta \boldsymbol{v}$ zvolí

$$\Delta \boldsymbol{v} = -\eta \nabla C, \tag{3.19}$$

kde η je dostatočne malé kladné číslo, tak je zaručené, že C sa bude zmenšovať. Krok η určuje rýchlosť konvergencie (*learning rate*).

Metóda najväčšieho spádu v každej iterácii vypočíta gradient a uskutoční posun o malú hodnotu z aktuálneho bodu v opačnom smere gradientu, teda v smere najväčšieho spádu funkcie. Teoreticky sa tento postup opakuje dovtedy, kým nie sú všetky parciálne derivácie nulové. V praxi sa ako podmienka na zastavenie (*stop condition*) používa hodnota parciálnych derivácií blízka nule [38].

Uvedený postup možno zovšeobecniť na minimalizáciu funkcie m premenných. Ak C je spojitá a diferencovateľná funkcia $C : \Re^m \to \Re$, potom v každom bode jej definičného oboru existuje gradient a má tvar:

$$\nabla C \equiv \left(\frac{\partial C}{\partial v_1}, \frac{\partial C}{\partial v_2}, \ldots, \frac{\partial C}{\partial v_m} \right)^T. \tag{3.20}$$

Metóda najväčšieho spádu štartuje z náhodne (alebo inak) zvoleného počiatočného bodu $\boldsymbol{v}_0 \in \Re^m$. V každej iterácii vypočíta gradient funkcie pre aktuálny bod $\boldsymbol{v}$ a určí nový bod $\boldsymbol{v}'$ podľa vzťahu:

$$\boldsymbol{v}' = \boldsymbol{v} - \eta \nabla C \tag{3.21}$$

pre vhodne zvolenú veľkosť kroku η.

Veľkosť kroku (*learning rate*) je možné voliť viacerými spôsobmi [38]. Často sa vyberie vhodná konštanta. Výkon algoritmu je závislý od správnej voľby tohto parametra. Ak je η príliš veľké, algoritmus osciluje (účelová funkcia skáče okolo minima), alebo dokonca diverguje. Naopak, ak je η príliš malé, algoritmus konverguje veľmi pomaly. Vhodným prístupom je prispôsobovanie veľkosti kroku počas optimalizácie (*adaptive learning rate*) tak, aby sa v jednej iterácii dosiahol väčší pokles optimalizovanej funkcie, a tak sa zrýchlila konvergencia, a zároveň aby nedošlo k oscilácii.

Metóda najväčšieho spádu konverguje k bodu, v ktorom sú všetky parciálne derivácie nulové. Vo všeobecnosti takýto bod nemusí byť globálnym minimom funkcie. Je to stacionárny bod, v ktorom sa nedá určiť smer ďalšieho pohybu. Môže to byť lokálne minimum, maximum alebo inflexný bod. Ak sú ako účelové funkcie volené konvexné funkcie, potom ich lokálne minimá sú zároveň globálnymi minimami a algoritmus konverguje ku globálnemu minimu.

Keďže účelová funkcia je definovaná ako priemer funkcií chyby pre jednotlivé pozorovania, v každej iterácii algoritmu treba vypočítať hodnotu parciálnych derivácií pre každé pozorovanie, čo môže byť niekedy výpočtovo náročné. V tom prípade je vhodné použiť stochastický variant metódy najväčšieho spádu (*stochastic gradient descent* – SGD), pri ktorom sa v každej iterácii náhodne vyberie dopredu zvolený počet k (*batch size*), $k << n$, pozorovaní a pomocou nich sa urobí odhad gradientu.

3.2.4 Hľadanie optimálnych váh

V navrhovanej metóde na výber premenných bude metóda najväčšieho spádu použitá na hľadenie vektora nezáporných váh $\boldsymbol{v} = (v_1, v_2, \ldots ..., v_m)$, v ktorom sa nadobúda minimum zvolenej účelovej funkcie $C : \Re^m \rightarrow \Re_0^+$ definovanej vzťahom 3.15.

V každej iterácii bude potrebné vypočítať gradient funkcie C pre aktuálny vektor váh $\boldsymbol{v}$ a určiť nový vektor váh posunom z pôvodného vektora v opačnom smere gradientu.

Výpočet gradientu znamená výpočet parciálnych derivácií funkcie C pre aktuálny

vektor váh podľa každej váhy v_l, kde $l \in \{1, 2, \ldots, m\}$ nasledovne:

$$\frac{\partial C(\mathbf{v})}{\partial v_l} = \frac{1}{n}\sum_{i=1}^{n}\frac{\partial l(y_i, p_i(\mathbf{v}))}{\partial v_l} = \frac{1}{n}\sum_{i=1}^{n}\frac{\partial l(y_i, p_i(\mathbf{v}))}{\partial p_i(\mathbf{v})}.\frac{\partial p_i(\mathbf{v})}{\partial v_l}. \tag{3.22}$$

Pre výpočet výrazu 3.22 je potrebné vypočítať parciálnu deriváciu predikovanej hodnoty $p_i(\mathbf{v})$ podľa váhy v_l. Pomocou vzťahov pre výpočet derivácie súčinu, podielu a zloženej funkcie možno derivovaním 3.13 vypočítať:

$$\begin{aligned}\frac{\partial p_i(\mathbf{v})}{\partial v_l} = \frac{-1}{\left[\sum_{k\neq i} w(d_{ik}(\mathbf{v}))\right]^2}.\sum_{k\neq i} w'(d_{ik}(\mathbf{v})).\frac{\partial d_{ik}(\mathbf{v})}{\partial v_l}.\sum_{j\neq i} w(d_{ij}(\mathbf{v}))y_j \quad + \\ + \frac{1}{\sum_{k\neq i} w(d_{ik}(\mathbf{v}))}.\sum_{j\neq i} w'(d_{ij}(\mathbf{v})).\frac{\partial d_{ij}(\mathbf{v})}{\partial v_l}.y_j\end{aligned} \tag{3.23}$$

Po úprave:

$$\begin{aligned}\frac{\partial p_i(\mathbf{v})}{\partial v_l} = \frac{-1}{\sum_{k\neq i} w(d_{ik}(\mathbf{v}))}.\sum_{k\neq i} w'(d_{ik}(\mathbf{v})).\frac{\partial d_{ik}(\mathbf{v})}{\partial v_l}.p_i(\mathbf{v}) \quad + \\ + \frac{1}{\sum_{k\neq i} w(d_{ik}(\mathbf{v}))}.\sum_{k\neq i} w'(d_{ik}(\mathbf{v})).\frac{\partial d_{ik}(\mathbf{v})}{\partial v_l}.y_k\end{aligned} \tag{3.24}$$

Z toho:

$$\frac{\partial p_i(\mathbf{v})}{\partial v_l} = \frac{1}{\sum_{k\neq i} w(d_{ik}(\mathbf{v}))}.\sum_{k\neq i} w'(d_{ik}(\mathbf{v})).\frac{\partial d_{ik}(\mathbf{v})}{\partial v_l}.(y_k - p_i(\mathbf{v})), \tag{3.25}$$

kde w' označuje prvú deriváciu hodnotiacej funkcie vzdialenosti w, čo je funkcia jednej premennej. Vzťahy 3.22 a 3.25 definujú spôsob výpočtu gradientu funkcie C vo všeobecnosti. Pre konkrétne prípady je potrebné do vzťahov dosadiť deriváciu zvolenej funkcie vzdialenosti, jej hodnotiacej funkcie a funkcie chyby.

Ďalej bude ukázané, ako sa vypočíta parciálna derivácia funkcie C podľa v_l v špeciálnom prípade, ak je použitá euklidovská vzdialenosť, hodnotiaca funkcia vzdialenosti $w(d) = e^{-d^2}$ a funkcia štvorcovej chyby definovaná vzťahom 3.14.

Euklidovská vzdialenosť (L2) dvoch pozorovaní x_i a x_k v závislosti od váhy $\boldsymbol{v}$ je vyjadrená vzťahom:

$$d_{ik}(\mathbf{v}) = L2(\mathbf{v} \circ x_i, \mathbf{v} \circ x_k) = \sqrt{\sum_{j=1}^{n} v_j^2.(x_{ij} - x_{kj})^2}. \tag{3.26}$$

Jej parciálna derivácia podľa v_l je

$$\frac{\partial d_{ik}(\mathbf{v})}{\partial v_l} = \frac{v_l.(x_{il} - x_{kl})^2}{\sqrt{\sum_{j=1}^{n} v_j^2.(x_{ij} - x_{kj})^2}} = \frac{v_l.(x_{il} - x_{kl})^2}{d_{ik}(\mathbf{v})} \tag{3.27}$$

Hodnotiaca funkcia vzdialenosti $w(d) = e^{-d^2}$ je funkcia jednej premennej a jej derivácia je

$$w'(d) = (e^{-d^2})' = -2d.e^{-d^2} = -2d.w(d). \tag{3.28}$$

Potom dosadením do vzťahu 3.25 možno odvodiť

$$\frac{\partial p_i(\mathbf{v})}{\partial v_l} = \frac{2v_l}{\sum_{k \neq i} w(d_{ik}(\mathbf{v}))} \sum_{k \neq i} w(d_{ik}(\mathbf{v})).(x_{il} - x_{kl})^2.(p_i(\mathbf{v}) - y_k). \tag{3.29}$$

Nakoniec pre funkciu $C(\mathbf{v})$, ktorá je aritmetickým priemerom zo štvorcových chýb pre každé pozorovanie, platí:

$$\frac{\partial C(\mathbf{v})}{\partial v_l} = \frac{1}{n} \sum_{i=1}^{n} \frac{\partial l(y_i, p_i(\mathbf{v}))}{\partial v_l} = \frac{2}{n} \sum_{i=1}^{n} (p_i(\mathbf{v}) - y_i) \frac{\partial p_i(\mathbf{v})}{\partial v_l}. \tag{3.30}$$

Vzťahy 3.29 a 3.30 definujú výpočet gradientu funkcie C v danom konkrétnom prípade. Z uvedených vzťahov vidno, že ak v niektorej iterácii váha v_l nadobudne hodnotu 0, bude parciálna derivácia funkcie $C(\mathbf{v})$ podľa tejto premennej nulová a váha v_l ostane nulová aj v každej ďalšej iterácii. Túto nežiadúcu vlastnosť možno odstrániť tak, že atribúty budú vážené druhou odmocninou váh (predpokladajú sa nezáporné váhy). Vzdialenosť $d_{ik}(\mathbf{v})$ bude potom modifikovaná nasledovne:

$$d_{ik}(\mathbf{v}) = L2(\sqrt{\mathbf{v}} \circ x_i, \sqrt{\mathbf{v}} \circ x_k) = \sqrt{\sum_{j=1}^{n} v_j.(x_{ij} - x_{kj})^2}. \tag{3.31}$$

Dá sa ukázať, že uvedená modifikácia má za následok zmenu vzťahu 3.29, ktorý bude upravený takto:

$$\frac{\partial p_i(\mathbf{v})}{\partial v_l} = \frac{1}{\sum_{k \neq i} w(d_{ik}(\mathbf{v}))} \sum_{k \neq i} w(d_{ik}(\mathbf{v})).(x_{il} - x_{kl})^2.(p_i(\mathbf{v}) - y_k). \tag{3.32}$$

Modifikácia vzdialenosti viedla k tomu, že z pravej strany rovnosti 3.29 bol vynechaný člen $2v_l$, ktorý spôsoboval, že v prípade $v_l = 0$ sa stala celá parciálna derivácia $C(\mathbf{v})$ podľa tejto premennej nulovou. Táto modifikácia zabraňuje uviaznutiu algoritmu v zložke s nulovou váhou.

3.2.5 Algoritmus na výber premenných

Na základe faktov popísaných v kapitolách 3.2.1 – 3.2.4 bol navrhnutý nový algoritmus pre selekciu premenných. Algoritmus používa metódu najväčšieho spádu na hľadanie optimálnych váh premenných, v ktorých účelová funkcia $C(\boldsymbol{v})$ nadobúda minimum. Umožňuje aj použitie stochastického variantu metódy.

Vstupom algoritmu je súbor dát X s m numerickými vysvetľujúcimi premennými $X_1, X_2, ..., X_m$ a s n pozorovaniami $x_1, x_2, ..., x_n$ a závislá premenná Y s hodnotami $y_1, y_2, ..., y_n$.

Ďalšie vstupné parametre sú:

a) definícia vzdialenosti,
b) hodnotiaca funkcia vzdialenosti,
c) funkcia chyby (*loss function*),
d) rýchlosť konvergencie (*learning rate*),
e) maximálny počet iterácií algoritmu (*stop condition*),
f) minimálna norma gradientu (*stop condition*),
g) počet náhodne vybraných pozorovaní pri SGD (*batch size*),
h) spôsob ošetrenia záporných váh,
i) príznak normalizácie gradientu.

Algoritmus bude vykonávať nasledovné kroky:

1. Najprv sa vykoná štandardizácia súboru dát, čiže úprava, po ktorej bude mať každá vysvetľujúca premenná strednú hodnotu 0 a štandardnú odchýlku 1.
2. Váhy vysvetľujúcich premenných sa nastavia na iniciálne hodnoty. Podporované hodnoty sú buď nulový vektor váh, vektor $\boldsymbol{v} = (\frac{1}{m}, ..., \frac{1}{m})$ alebo iný špecificky zvolený iniciálny vektor váh.
3. Ďalej sa budú vykonávať nasledovné kroky:
   (a) V prípade kladnej hodnoty parametra *batch_size* sa urobí náhodný výber *batch_size* pozorovaní zo súboru dát X (použitie SGD). Ak je hodnota *batch_size* ≤ 0, zoberie sa celý súbor dát X (použitie GD).

(b) Pri použití GD sa vypočíta matica vzdialeností pozorovaní so zohľadnením váh $d(\boldsymbol{v})$ podľa vybranej definície vzdialenosti. V prípade SGD sa počíta vzdialenosť pozorovaní vybraných v kroku 3(a) voči všetkým pozorovaniam súboru dát.

(c) Z matice vzdialeností sa vypočíta matica ohodnotených vzdialeností $w(d(\boldsymbol{v}))$ aplikovaním definovanej hodnotiacej funkcie.

(d) Podľa vzťahu 3.13 sa z matice $w(d(\boldsymbol{v}))$ a vektora hodnôt cieľovej premennej Y vypočíta vektor predikcií.

(e) Podľa zvolenej funkcie chyby sa pre aktuálnu hodnotu vektora váh $\boldsymbol{v}$ z predikovaných hodnôt a hodnôt cieľovej premennej vypočíta hodnota účelovej funkcie $C(\boldsymbol{v})$.

(f) Použitím vzťahov 3.22, 3.25 a derivácií funkcií zadaných ako parametre algoritmu sa vypočíta gradient, teda vektor parciálnych derivácií funkcie $C(\boldsymbol{v})$ podľa všetkých zložiek vektora váh pri aktuálnych váhach.

(g) Ak je nastavený príznak normalizácie gradientu, urobí sa L1–normalizácia gradientu. To znamená, že sa každá jeho zložka predelí súčtom absolútnych hodnôt všetkých zložiek.

(h) Po výpočte a prípadnej normalizácii gradientu funkcie C pre aktuálny vektor váh sa určí nový vektor váh podľa vzťahu 3.21 s použitím nastaveného parametra η označujúceho rýchlosť konvergencie.

(i) Ak sa medzi zložkami nového vektora váh $\boldsymbol{v}'$ vyskytujú záporné hodnoty, tak sa upravia podľa nastaveného spôsobu ošetrenia záporných váh. Napríklad, ak je nastavený spôsob *new_eta*, tak sa hodnota parametra η automaticky zmenší tak, aby nevznikli žiadne záporné váhy. Ďalšie možnosti ošetrenia záporných váh sú popísané v Prílohe C.

4. Kroky v bode 3 sa budú opakovať dovtedy, kým nebude splnená podmienka na zastavenie algoritmu, t. j. kým nebude dosiahnutý určený počet iterácií alebo kým sa norma gradientu nezmenší pod požadovanú hodnotu.

Výsledkom je vektor váh $\boldsymbol{v} = (v_1, v_2, \ldots ..., v_m)$ vypočítaný v poslednej iterácii algoritmu. V prípade konvergencie sa približuje k optimálnemu riešeniu, t. j. hodnota účelovej funkcie v tomto bode bude minimálna a vektor predikovaných hodnôt $(p_1(\boldsymbol{v}), \ldots, p_n(\boldsymbol{v}))$ bude pomerne presne odhadovať skutočnú hodnotu vektora Y. Vektor $\boldsymbol{v}$ nezáporných reálnych čísel zároveň určuje váhy vysvetľujúcich premenných $X_1, X_2, \ldots ..., X_m$, čo možno použiť na selekciu premenných. Premenné s väčšou váhou sú viac relevantné, teda majú väčší vplyv na hodnotu závislej premennej.

3.2.6 Príklad použitia algoritmu na výber premenných

Použitie a vlastnosti algoritmu na selekciu premenných popísaného v predchádzajúcej kapitole budú ilustrované na jednoduchom príklade. Ako vstupný súbor dát X bude použitý náhodne vygenerovaný súbor dát s Gaussovým normálnym rozdelením v 4–rozmernom priestore s 200 pozorovaniami. Cieľová premenná bude definovaná vzťahom $Y = X_0 + 4X_1$. Ako ďalšie vstupné parametre budú použité:

a) euklidovská vzdialenosť,
b) hodnotiaca funkcia vzdialenosti $w(d) = e^{-d^2}$,
c) funkcia štvorcovej chyby,
d) rýchlosť konvergencie $\eta = 0.1$,
e) počet iterácií algoritmu 1000,
f) minimálna norma gradientu $\epsilon = 10^{-6}$,
g) počet náhodne vybraných pozorovaní $batch\_size = 0$,
h) spôsob kontroly záporných váh – *new_eta*,
i) príznak normalizácie gradientu – *true*.

Po štandardizácii súboru dát sa ako iniciálny vektor váh nastaví vektor $(0, 0, 0, 0)$. Pri tomto nastavení je pre vybraný bod súboru dát vzdialenosť každého jeho suseda rovnaká (0) a po uplatnení hodnotiacej funkcie má potom každý sused ohodnotenú vzdialenosť rovnú 1. Prvá predikovaná hodnota cieľovej premennej sa vypočíta ako aritmetický priemer hodnôt cieľovej premennej pre všetky ostatné body súboru dát.

V každej iterácii sa opakuje výpočet gradientu pre aktuálny vektor váh a posun vektora o malú hodnotu opačným smerom ako je smer gradientu. Algoritmus končí po tisíc iteráciách. V tabuľke 3 – 5 sú zachytené hodnoty váh a účelovej funkcie MSE (*Mean Square Error*) vo vybraných iteráciách.

Tabuľka 3 – 5 Hodnoty váh vo vybraných iteráciách algoritmu WkNN–FS

Iterácia	v_0	v_1	v_2	v_3	MSE
1.	0.0	0.0	0.0	0.0	18.47510794
10.	0.13839425	0.76160575	0.0	0.0	3.35289128
100.	1.64683764	8.13239104	0.0	0.0	0.27747157
1000.	3.44704490	33.17725378	0.0	0.0	0.17592660

Algoritmus určil nenulové váhy relevantným premenným X_0 a X_1, premenné X_2 a X_3 majú nulové váhy. Pritom premenná X_1 má vyššiu váhu ako premenná X_0, lebo má väčší vplyv na hodnotu cieľovej premennej.

Štandardizácia súboru dát v prvom kroku algoritmu znamená transformáciu každej premennej X_i na

$$X_i' = \frac{X_i - mean(X_i)}{std(X_i)}. \tag{3.33}$$

Tým sa body súboru dát x_i transformujú na body x_i'.

Nájdenú dvojicu nenulových váh možno interpretovať ako transformáciu štvorrozmerného priestoru do dvojrozmerného priestoru definovanú vzťahmi:

$$\begin{aligned} X_0'' &= \sqrt{v_0}.X_0' = 1.85662191.X_0' \\ X_1'' &= \sqrt{v_1}.X_1' = 5.75996995.X_1' \end{aligned} \tag{3.34}$$

Uvedená transformácia znamená natiahnutie osí, pričom viac sa natiahne os X_1. Pri výpočte bola uvažovaná vzdialenosť bodov súboru dát definovaná takto:

$$d_{ij}(\mathbf{v}) = L2(\sqrt{\mathbf{v}} \circ x_i', \sqrt{\mathbf{v}} \circ x_j') = L2(x_i'', x_j''), \tag{3.35}$$

čo po transformácii 3.34 do dvojrozmerného priestoru zodpovedá štandardnej euklidovskej vzdialenosti.

Navrhovanú metódu možno označiť ako vloženú metódu v tom zmysle, že je založená na použití k–NN regresie s $k = n - 1$, kde n je počet prvkov súboru dát, a hodnotiacou funkciou vzdialenosti $w(d) = e^{-d^2}$. Bolo overené, že použitím triedy `KNeighborsRegressor` z knižnice `sklearn.neighbors` jazyka Python s vyššie uvedenými parametrami sa po uplatnení transformácie 3.34 získa predikcia, ktorá má pri krížovej validácii prostredníctvom LOO nízku chybu MSE = 0.17592660 zhodnú s výsledkom popísaného algoritmu. Uvedená metóda teda poskytuje nielen výber premenných, ale aj metódu predikcie a základný odhad chyby.

Z praktického hľadiska však nie je vhodné používať nastavenie $k = n - 1$, lebo pre každý prvok testovacej množiny by bolo treba počítať vzdialenosť od všetkých ostatných bodov tréningovej množiny. Keďže funkcia $w(d) = e^{-d^2}$ s rastúcou vzdialenosťou veľmi prudko klesá, tak vzdialenejší susedia majú veľmi nízke váhy, ktoré možno zanedbať. Preto stačí ako aproximáciu zvoliť k ako dosť veľkú konštantu s použitím klasického k–NN alebo použiť variant k–NN, pri ktorom sa uvažujú susedia vo vzdialenosti najviac r (*radius–based* k–NN), s nastavením dosť veľkej hodnoty r. Samozrejme navrhovanú metódu možno použiť len na výber premenných a na vybrané premenné aplikovať ľubovoľnú inú metódu strojového učenia.

3.2.7 Parametrizácia využitím rôznych funkcií pre výpočet chyby

Navrhnutý algoritmus na výber premenných má tri stupne voľnosti. Umožňuje voliť rôzne definície vzdialenosti, hodnotiace funkcie vzdialenosti a tiež rôzne funkcie chyby (*loss functions*).

Na vyjadrenie chyby predikcie v regresnej úlohe sa obvykle používajú funkcie chyby založené na vzdialenosti [39]. Pre tieto funkcie platí, že ich hodnota je tým menšia, čím viac sa predikovaná hodnota blíži skutočnej hodnote. Hodnota funkcie chyby je nulová práve vtedy, ak sa predikovaná hodnota rovná skutočnej.

Navrhnutý algoritmus na výber premenných podporuje viacero funkcií chyby, ktoré sú popísané nižšie.

Nech x_i je ľubovoľné pozorovanie z množiny $\{x_1, x_2, ..., x_n\}$, y_i jeho skutočná hodnota cieľovej premennej a $p_i(\boldsymbol{v})$ jej predikovaná hodnota v závislosti od váh $\boldsymbol{v}$.

1. Absolútna chyba (AE – *Absolute Error*)

   Hodnota absolútnej chyby pre dvojicu $(y_i, p_i(\boldsymbol{v}))$ je definovaná ako

   $$l(y_i, p_i(\boldsymbol{v})) = |y_i - p_i(\boldsymbol{v})| . \tag{3.36}$$

   Parciálna derivácia tejto funkcie chyby podľa v_l sa vypočíta takto:

   $$\frac{\partial l(y_i, p_i(\boldsymbol{v}))}{\partial v_l} = sgn(p_i(\boldsymbol{v}) - y_i)\frac{\partial p_i(\boldsymbol{v})}{\partial v_l}. \tag{3.37}$$

2. Štvorcová chyba (SE –*Squared Error*)

   Hodnota štvorcovej chyby je definovaná takto:

   $$l(y_i, p_i(\boldsymbol{v})) = (y_i - p_i(\boldsymbol{v}))^2. \tag{3.38}$$

   Parciálna derivácia funkcie štvorcovej chyby podľa v_l je rovná:

   $$\frac{\partial l(y_i, p_i(\boldsymbol{v}))}{\partial v_l} = 2(p_i(\boldsymbol{v}) - y_i)\frac{\partial p_i(\boldsymbol{v})}{\partial v_l}. \tag{3.39}$$

3. Huberova funkcia chyby (HL –*Huber loss function*)

   Hodnota Huberovej funkcie pre $\delta > 0$ je definovaná ako

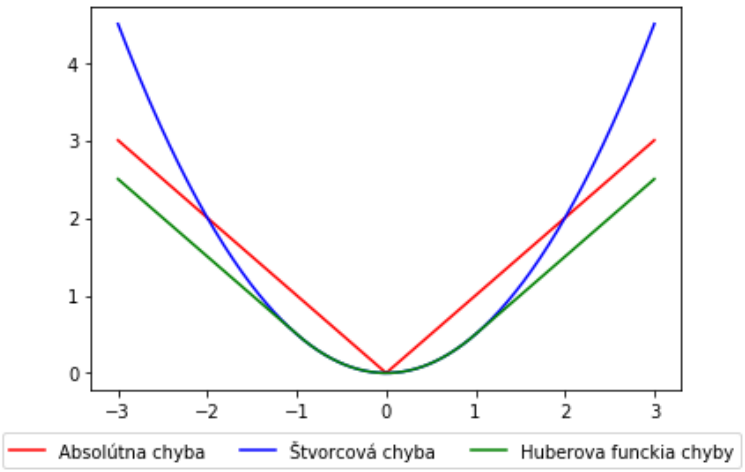
   $$l_\delta(y_i, p_i(\boldsymbol{v})) = \begin{cases} \frac{1}{2}(y_i - p_i(\boldsymbol{v}))^2, & \text{ak } |y_i - p_i(\boldsymbol{v})| \leq \delta, \\ \delta\,|y_i - p_i(\boldsymbol{v})| - \frac{1}{2}\delta^2, & \text{inak.} \end{cases} \tag{3.40}$$

   Pre parciálnu deriváciu Huberovej funkcie podľa v_l platí:

   $$\frac{\partial l_\delta(y_i, p_i(\boldsymbol{v}))}{\partial v_l} = \begin{cases} (p_i(\boldsymbol{v}) - y_i).\frac{\partial p_i(\boldsymbol{v})}{\partial v_l}, & \text{ak } |y_i - p_i(\boldsymbol{v})| \leq \delta, \\ \delta.sgn(p_i(\boldsymbol{v}) - y_i).\frac{\partial p_i(\boldsymbol{v})}{\partial v_l}, & \text{inak.} \end{cases} \tag{3.41}$$

   Z definície vyplýva, že ak je rozdiel medzi skutočnou a predikovanou hodnotou cieľovej premennej malý, Huberova funkcia zodpovedá štvorcovej chybe a naopak, ak je rozdiel veľký, zodpovedá absolútnej chybe.

Nevýhodou absolútnej chyby je, že nie je diferencovateľná v bodoch, pre ktoré platí $y_i - p_i(\boldsymbol{v}) = 0$. V praxi možno v týchto prípadoch ako hodnotu derivácie uvažovať 0, lebo predikovaná hodnota sa rovná skutočnej a dané pozorovanie neindikuje potrebu pohybu v opačnom smere gradientu. Štvorcová chyba je diferencovateľná vo všetkých bodoch definičného oboru. Jej nevýhodou je slabá odolnosť voči odľahlým bodom (*outliers*), čo je spôsobené práve použitím druhej mocniny chyby. Výsledná hodnota účelovej funkcie je viac ovplyvnená väčšími chybami ako priemernými chybami. Dá sa ukázať, že Huberova funkcia chyby je diferencovateľná v každom bode definičného oboru. Absolútna chyba a Huberova funkcia sú viac odolné voči odľahlým bodom ako štvorcová chyba.



Obrázok 3 – 2 Grafy funkcií chyby

Na obrázku 3 – 2 sú znázornené grafy funkcií: absolútnej chyby, štvorcovej chyby a Huberovej funkcie chyby s parametrom $\delta = 1$ ako funkcie jednej premennej – rozdielu skutočnej a predikovanej hodnoty.

V binárnych klasifikačných úlohách sa na určenie chyby často používa binárna krížová entropia (CE – *Cross–Entropy*) [37] definovaná vzťahom:

$$l(y_i, p_i(\boldsymbol{v})) = -(y_i \ln p_i(\boldsymbol{v}) + (1 - y_i) \ln(1 - p_i(\boldsymbol{v}))), \tag{3.42}$$

kde y_i je skutočná hodnota cieľovej premennej pozorovania x_i a $p_i(\boldsymbol{v})$ jej predikovaná hodnota.

Parciálna derivácia tejto funkcie sa vypočíta takto:

$$\frac{\partial l(y_i, p_i(\mathbf{v}))}{\partial v_l} = \left(\frac{1 - y_i}{1 - p_i(\mathbf{v})} - \frac{y_i}{p_i(\mathbf{v})} \right) \frac{\partial p_i(\mathbf{v})}{\partial v_l}. \tag{3.43}$$

Binárna krížová entropia meria výkonnosť klasifikačného modelu, ktorého výstupom je hodnota pravdepodobnosti z intervalu $(0, 1)$. Hodnota chyby rastie, ak sa predikovaná pravdepodobnosť vzdiaľuje od skutočnej hodnoty [37].

3.2.8 Použitie metrík a hodnotiacich funkcií vzdialenosti

V práci [34] je definovaných viacero funkcií vzdialenosti pre rôzne typy súborov dát. Pre numerické vysvetľujúce premenné uvažované v navrhovanej metóde môže byť použitá napríklad Minkowského metrika.

Nech x_i a x_j sú dve pozorovania. Každé pozorovanie je vektorom, ktorého zložky sú jeho hodnoty atribútov: $x_i = (x_{i1}, x_{i2}, \ldots..., x_{im})$ a $x_j = (x_{j1}, x_{j2}, \ldots..., x_{jm})$. Minkowského p–vzdialenosť (Lp) bodov x_i a x_j je definovaná vzťahom:

$$d_{ij} = \sqrt[p]{\sum_{k=1}^{m} |x_{ik} - x_{jk}|^p} \tag{3.44}$$

Špeciálnym prípadom Minkowského definície vzdialenosti je pre $p = 1$ manhattanská (*city–block*) vzdialenosť a pre $p = 2$ euklidovská vzdialenosť.

V navrhovanej metóde pre výber premenných sa používa vzdialenosť pozorovaní s uplatnením váh atribútov. Ako bolo popísané v 3.2.4 pre euklidovskú vzdialenosť ($p = 2$), pri vážení atribútov vektorom $\boldsymbol{v}$ môže dôjsť k uviaznutiu algoritmu pre zložky váh, ktoré v niektorej iterácii algoritmu nadobudli hodnotu nula. Tieto zložky vektora $\boldsymbol{v}$ budú mať hodnotu nula aj vo všetkých ďalších iteráciách algoritmu. Tento jav nastáva aj pre Minkovského vzdialenosť s ľubovoľným p. Preto je výhodnejšie použiť váženie atribútov p–tou odmocninou váh. Minkovského p–vzdialenosť pozorovaní x_i a x_j s upraveným vážením atribútov je vyjadrená nasledovne:

$$d_{ij}(\mathbf{v}) = Lp(\sqrt[p]{\mathbf{v}} \circ x_i, \sqrt[p]{\mathbf{v}} \circ x_j) = \sqrt[p]{\sum_{k=1}^{m} v_k |x_{ik} - x_{jk}|^p}. \tag{3.45}$$

Tento vzťah je alternatívou k definícii vzdialenosti podľa 3.12.

Parciálna derivácia vzdialenosti definovanej vzťahom 3.45 podľa v_l je

$$\frac{\partial d_{ij}(\mathbf{v})}{\partial v_l} = \frac{|x_{il} - x_{jl}|^p}{p.(\sqrt[p]{\sum_{k=1}^{m} v_k |x_{ik} - x_{jk}|^p})^{p-1}} = \frac{|x_{il} - x_{jl}|^p}{p.(d_{ij}(\mathbf{v}))^{p-1}} \tag{3.46}$$

Vzdialenosť d je v navrhovanej metóde transformovaná pomocou hodnotiacej funkcie w s vlastnosťami popísanými v kapitole 3.2.2.

V práci sú uvažované tieto funkcie:

$$w(d) = e^{-cd^\alpha}, \text{ kde } c, \alpha \in \Re^+, \tag{3.47}$$

$$w(d) = \frac{1}{1 + cd^\alpha}, \text{ kde } c, \alpha \in \Re^+. \tag{3.48}$$

Dá sa ukázať, že takto definované funkcie spĺňajú všetky tri požadované vlastnosti. Funkcia 3.47 pre $\alpha = 2$, ktorá bude v ďalšom označovaná aj *rbf*, zodpovedá funkcii *Radial Basis Function* (RBF *kernel*) používanej v metóde SVM (*Support Vector Machines*) popísanej v kapitole 2.4.2. Pre variant funkcie 3.47 s $\alpha = 1$ bude použité označenie *exp*.

Uvažované funkcie sú funkcie jednej premennej s deriváciami:

$$w'(d) = (e^{-cd^\alpha})' = -c\alpha.d^{\alpha-1}w(d), \tag{3.49}$$

$$w'(d) = \left(\frac{1}{1 + cd^\alpha}\right)' = -c\alpha.d^{\alpha-1}w(d)^2. \tag{3.50}$$

Ako vidno z vyššie uvedených vzťahov, derivácie hodnotiacich funkcií je možné vyjadriť pomocou hodnôt $w(d)$, ktoré sú pri realizácii algoritmu vypočítané v predchádzajúcom kroku.

3.2.9 Rekurzívna eliminácia premenných

Výsledkom algoritmu na výber premenných popísaného v kapitole 3.2.5 je vektor váh $\boldsymbol{v} = (v_1, v_2, \ldots..., v_m)$ vypočítaný v poslednej iterácii algoritmu. Všetky premenné s nulovou váhou možno vynechať. Pre ostatné premenné bude určené ich poradie (*ranking*) podľa váhy od najväčšej po najmenšiu.

Ak ostalo priveľa premenných s nenulovou váhou, potom sa dá postupovať tak, že ďalej sa bude postupne eliminovať vždy premenná s najmenšou váhou. Po vynechaní

premennej sa rekurzívne opakuje základný algoritmus so zostávajúcimi premennými. Keďže úvodné nastavenie váh sa zoberie z posledného optimálneho riešenia, postačí menší počet iterácií ako pri prvom behu algoritmu. Eliminácia sa opakuje po požadovaný počet vybraných premenných. Výsledkom bude tabuľka, v ktorej sa v každom riadku nachádza počet vybraných premenných, príslušná hodnota účelovej funkcie, zoznam premenných a zoznam ich váh po vykonaných elimináciách. Výsledkom riešenia je posledný riadok tabuľky.

Alternatívne je možné opakovať elimináciu až po poslednú premennú. Potom výsledná tabuľka poskytuje kompletný prehľad hodnôt účelovej funkcie pre rôzne počty vybraných premenných. Keďže premenné sa postupne vynechávajú, tak optimálne hodnoty účelovej funkcie sú neklesajúce a obvykle to platí aj o dosiahnutých hodnotách účelovej funkcie v tabuľke. Na základe hodnoty účelovej funkcie možno potom určiť vhodný počet vybraných premenných tak, aby nebol príliš veľký a aby hodnota účelovej funkcie bola dostatočne nízka.

Ako príklad možno uviesť podobný súbor dát ako v kapitole 3.2.6 avšak s 50 premennými, z ktorých prvé dve sú relevantné. V tabuľke 3 – 6 sa nachádza niekoľko riadkov výslednej tabuľky po rekurzívnej eliminácii po poslednú premennú.

Tabuľka 3 – 6 Výsledky rekurzívnej eliminácie premenných metódou WkNN–FS

P. prem.	Chyba	Premenné	Váhy
27	0.0076	1, 0, 38, 22, 3,...	10.25, 2.32, 0.36, 0.30,...
...	...	...	...
4	0.0113	1, 0, 38, 45	11.51, 1.90, 0.31, 0.18
3	0.0115	1, 0, 38	11.91, 1.98, 0.22
2	0.0121	1, 0	12.45, 2.35
1	0.0843	1	13.35

Ako vidno, je potrebné vybrať minimálne 2 premenné, aby sa dosiahla dostatočne nízka hodnota účelovej funkcie (chyba) a pridávanie ďalších premenných už neprináša výrazné zlepšenie.

3.2.10 Použitie metódy pre súbory dát s veľkým počtom pozorovaní

Pre súbory dát s veľkým počtom pozorovaní sa rýchlo zvyšuje výpočtová náročnosť metódy. Uvedená skutočnosť vyplýva už z toho, že v každej iterácii je potrebné vypočítať maticu vzdialeností, ktorá má n^2 prvkov, kde n je počet pozorovaní. Pri vyššom počte pozorovaní sa tak rýchlo narazí na obmedzenie veľkosti operačnej pamäte alebo doby výpočtu.

Riešením uvedeného problému je použitie stochastického variantu metódy najväčšieho spádu. Hlavná myšlienka je založená na tom, že gradient účelovej funkcie sa počíta ako aritmetický priemer hodnôt parciálnych derivácií vypočítaných pre jednotlivé pozorovania súboru dát. Pre urýchlenie výpočtu sa teda dá odhadnúť ako priemer hodnôt pre malú vzorku náhodne vybraných pozorovaní súboru dát.

Pri štandardnej implementácii SGD sa vzorky vyberajú tak, že sa zvolí náhodná permutácia pozorovaní súboru dát a potom sa postupne prechádza súbor dát po častiach nazývaných *mini–batch*. Jeden prechod sa nazýva epocha. Na začiatku ďalšej epochy sa urobí ďalšia náhodná permutácia súboru dát [37].

Navrhnutá metóda podporuje štandardnú implementáciu SGD, ale navyše kvôli dosiahnutiu vyššej stability odhadu gradientu bol podporený aj špeciálny spôsob generovania vzoriek. Najprv sa pozorovania rozdelia do skupín na základe hodnoty cieľovej premennej. Ak cieľová premenná nadobúda len malý počet rôznych hodnôt, tak jednotlivé skupiny sú priamo určené týmito hodnotami. V opačnom prípade sa pozorovania rozdelia do 10–tich rovnako veľkých skupín podľa percentilov cieľovej premennej. Potom sú pozorovania súboru dát náhodne rozdelené na vzorky tak, aby každá vzorka obsahovala rovnaké zastúpenie jednotlivých skupín ako celý súbor dát (obdoba tzv. *stratified folds*).

V implementácii navrhnutej metódy je použitie SGD indikované kladnou hodnotou parametra *batch_size*. Implicitne sa uplatní štandardná implementácia SGD, pre špeciálny spôsob generovania vzoriek je potrebné nastaviť parameter *stratified_SGD*.

Na overenie implementácie novej metódy s využitím SGD boli použité syntetické

regresné súbory dát LinReg a Friedman (kapitola 2.5.1) s 5 000 pozorovaniami a 500 premennými, z ktorých bolo 5 relevantných. Pri výpočtoch bol použitý parameter *batch_size*=200. Implementácia bola overovaná s rôznymi nastaveniami parametrov metódy, ako funkcia chyby, metrika a hodnotiaca funkcia vzdialenosti (*kernel*). Celkovo bolo overovaných 9 variantov uvedených v tabuľke 3 – 7.

Pri súbore dát LinReg identifikovala nová metóda všetkých 5 známych relevantných premenných v prípade všetkých testovaných variantov. Pri súbore dát Friedman boli pri použití štvorcovej chyby a euklidovskej vzdialenosti v kombinácii s funkciou e^{-d} identifikované len 4 známe relevantné premenné z piatich.

Tabuľka 3 – 7 Výsledné chyby pre súbor dát Friedman pre rôzne varianty WkNN–FS

Chyba	Metrika	Kernel	MSE	MAE	MHL
SE	L2	rbf	1.8468	1.0745	0.1026
SE	L1	exp	1.7984	1.0644	0.1015
SE	L2	exp	3.7868	1.5834	0.1534
AE	L2	rbf	1.8563	1.0777	0.1029
AE	L1	exp	1.7967	1.0630	0.1014
AE	L2	exp	1.8796	1.0829	0.1034
HL	L2	rbf	1.8565	1.0778	0.1029
HL	L1	exp	1.7969	1.0629	0.1014
HL	L2	exp	1.8768	1.0823	0.1033

Pre analýzu danej situácie bol spracovaný program, ktorý pre finálne nájdené váhy premenných vypočítal všetky uvažované priemerné chyby. Z tabuľky 3 – 7 vidno, že uvedený variant metódy dosiahol omnoho vyššiu hodnotu MSE (ale aj ďalších používaných účelových funkcií) ako ostatné varianty. V tomto prípade algoritmus SGD zrejme uviazol v nejakom lokálnom minime, pričom hodnota účelovej funkcie sa výrazne odlišuje od optimálnej hodnoty.

Tabuľka 3 – 7 zároveň ukazuje spôsob, ako porovnávať jednotlivé varianty novej metódy na výber premenných a ktorý variant vybrať, alebo naopak vylúčiť pri výbere premenných pre konkrétny súbor dát.

3.2.11 Regularizácia

Ak má vytvorený model veľa voľných parametrov, môže dôjsť k pretrénovaniu [37]. To znamená, že aj keď model dobre popisuje tréningovú množinu, môže zlyhať v prípade nových dát. Redukciu pretrénovania možno dosiahnuť zväčšením počtu pozorovaní tréningovej množiny alebo zmenšením počtu parametrov, čo ale niekedy nie je možné. Existujú však aj iné techniky na obmedzenie pretrénovania, ako napríklad regularizácia.

V prípade navrhnutej metódy na výber premenných použitie regularizácie znamená rozšírenie účelovej funkcie (*cost function*) definovanej vzťahom 3.15 na nasledovný tvar:

$$C(\mathbf{v}) = \frac{1}{n}\sum_{i=1}^{n} l(y_i, p_i(\mathbf{v})) + \lambda\sum_{j=1}^{m} |v_j|^q, \tag{3.51}$$

kde $\lambda \in \Re_0^+$ je regularizačný parameter a $\sum_{j=1}^{m} |v_j|^q$ je regularizačný člen zodpovedajúci Lq norme vektora váh premenných $\mathbf{v}$, pričom $q \in \{0, 1, 2\}$. V tomto prípade bude metóda na výber premenných hľadať taký vektor váh, v ktorom sa bude nadobúdať minimálna hodnota rozšírenej funkcie $C(\mathbf{v})$.

Vo všeobecnosti sa najčastejšie používa $L1$ a $L2$ regularizácia. Z pohľadu výberu premenných sa ako výhodnejšia javí $L1$ regularizácia, lebo má tendenciu produkovať riedke riešenia, t. j. riešenia s malým počtom nenulových váh premenných [40].

Implementácia novej metódy na výber premenných podporuje $L1$ regularizáciu, kde pri hľadaní minima účelovej funkcie $C(\mathbf{v})$ definovanej vzťahom 3.51 ($q = 1$) pomocou metódy najväčšieho spádu sa v každom kroku algoritmu vypočítajú parciálne derivácie funkcie $C(\mathbf{v})$ podľa každej váhy v_l, $l \in \{1, 2, \ldots, m\}$, nasledovne:

$$\frac{\partial C_(\mathbf{v})}{\partial v_l} = \frac{1}{n}\sum_{i=1}^{n} \frac{\partial l(y_i, p_i(\mathbf{v}))}{\partial v_l} + \lambda.sgn(v_l). \tag{3.52}$$

Aplikovaním metódy s použitím $L1$ regularizácie na súbor dát LinReg so 100 pozorovaniami a 500 premennými, z toho 5 TF, boli pre rôzne hodnoty parametra λ získané výsledky uvedené v tabuľke 3 – 8. Pritom bola uplatnená funkcia štvorcovej chyby, euklidovská metrika a hodnotiaca funkcia vzdialenosti $w(d) = e^{-d^2}$.

Bez použitia regularizácie, t. j. pre hodnotu $\lambda = 0$ bol dosiahnutý index úspešnosti identifikovania relevantných premenných (Suc.) 0,80 pri hodnote chyby MSE približne 24,66. Možno si všimnúť, že s rastúcou hodnotou λ chyba MSE rastie a hodnota regularizačného člena klesá. Pritom sa počet nenulových váh znižuje. Hodnota indexu úspešnosti sa pri použití $\lambda = 500$ zvýšila na 1,00, teda boli úspešne identifikované všetky TF, a tento stav pretrvával aj pre $\lambda = 800$. Avšak pri ďalšom zvyšovaní λ na 900 dochádza k zníženiu indexu úspešnosti na 0,80. Príklad ukazuje, že po prekročení určitej hodnoty λ v účelovej funkcii je sčítanec s regularizačným členom výrazne väčší ako hodnota chyby a algoritmus pri hľadaní minima funkcie $C(\mathbf{v})$ preferuje znižovanie hodnoty tohto sčítanca, teda sumy váh premenných.

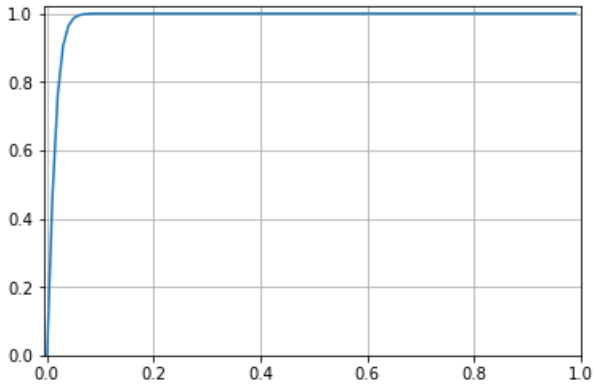
Tabuľka 3 – 8 Použitie metódy WkNN–FS s L1 regularizáciou

Lambda	MSE	Reg. člen	Suc.	Počet nenulových váh
0	24.66	12.0818	0.80	416
100	80.22	7.8640	0.80	348
500	665.08	5.4585	1.00	234
800	1322.35	4.4600	1.00	206
900	1692.85	4.0064	0.80	200

Z tabuľky 3 – 8 vidno, že aj pri hodnote indexu úspešnosti 1,00 bolo v nájdenom riešení veľa nenulových váh. Z hľadiska výberu premenných sa zdá vhodnejšie použitie L0 regularizácie, pri ktorej za predpokladu, že $0^0 \equiv 0$, regularizačný člen $\sum_{j=1}^{m} |v_j|^0$ vyjadruje počet nenulových váh. Avšak táto funkcia nie je spojitá, a teda neumožňuje použitie metódy najväčšieho spádu. Preto bola v práci navrhnutá modifikovaná L0 regularizácia s regularizačným členom vyjadreným pomocou súčtu spojitých sigmoidálnych funkcií, z ktorých každá približne indikuje nenulovosť váhy jednej premennej (graf jednej funkcie je na obrázku 3 – 3).

Pri navrhnutej úprave L0 regularizácie má funkcia $C(\mathbf{v})$ tvar:

$$C(\mathbf{v}) = \frac{1}{n}\sum_{i=1}^{n} l(y_i, p_i(\mathbf{v})) + \lambda \sum_{j=1}^{m} 2\left(\frac{1}{1+e^{-\alpha|v_j|}} - \frac{1}{2}\right). \tag{3.53}$$



Obrázok 3 – 3 Graf funkcie pre modifikáciu L0 regularizácie ($\alpha = 100$)

Parciálna derivácia funkcie $C(\mathbf{v})$ podľa váhy v_l je:

$$\frac{\partial C_(\mathbf{v})}{\partial v_l} = \frac{1}{n}\sum_{i=1}^{n}\frac{\partial l(y_i, p_i(\mathbf{v}))}{\partial v_l} + 2\lambda\alpha\frac{sgn(v_l).e^{-\alpha.|v_l|}}{(1+e^{-\alpha|v_l|})^2}. \tag{3.54}$$

Overovanie modifikovanej $L0$ regularizácie bolo vykonané na rovnakom súbore dát s použitím parametra $\alpha = 100$. Výsledky sú zobrazené v tabuľke 3 – 9. Oproti výsledkom dosiahnutým pomocou L1 regularizácie vidno, že hodnota regularizačného člena (okrem $\lambda = 0$) odhaduje pomerne presne počet nenulových váh, ktorý klesá omnoho rýchlejšie pri porovnateľnej veľkosti chyby. To dáva návod pre voľbu parametra λ – postupným zvyšovaním sa vyberie hodnota, pri ktorej počet nenulových váh približne zodpovedá požadovanému počtu vyberaných premenných.

Tabuľka 3 – 9 Použitie metódy WkNN–FS s modifikovanou L0 regularizáciou

Lambda	MSE	Reg. člen	Suc.	Počet nenulových váh
0	24.66	251.5831	0.80	416
30	872.11	25.9972	0.80	26
40	1423.17	13.9999	1.00	14
60	1704.55	7.9999	1.00	8
70	1801.85	5.9999	0.80	6

Regularizácia vedie k zníženiu počtu nenulových váh v nájdenom optimálnom riešení, a teda je alternatívou k rekurzívnej eliminácii premenných (kapitola 3.2.9).

4 Vyhodnotenie experimentálnych výsledkov

Navrhnutá metóda na výber premenných (WkNN–FS) bola hodnotená z hľadiska úspešnosti identifikovania známych relevantných premenných (TF – *True Features*), z hľadiska vplyvu výberu premenných na presnosť predikcie a z hľadiska stability. Merania boli uskutočnené na vybraných syntetických súboroch dát rôznych rozmerov a tiež na vysokodimenzionálnych reálnych dátach z biomedicínskeho prostredia. Výsledky meraní boli porovnané s výsledkami dosiahnutými pomocou známych metód na výber premenných.

Navrhnutý prístup umožňuje vytvárať rôzne varianty metódy WkNN–FS, ktoré závisia od definície vzdialenosti, hodnotiacej funkcie vzdialenosti a funkcie chyby. Pre hodnotenie bolo vybraných 12 variantov s použitím rôznych kombinácií parametrov:

- definícia vzdialenosti: euklidovská (L2) a manhattanská (L1),
- hodnotiaca funkcia vzdialenosti: $w(d) = e^{-d^2}$ (rbf) a $w(d) = e^{-d}$ (exp),
- funkcia chyby: štvorcová chyba (SE), absolútna chyba (AE), Huberova funkcia chyby (HL) a v prípade súborov dát pre binárne klasifikačné úlohy aj binárna krížová entropia (CE).

Vo výsledkoch sú parametre uvádzané pri označení WkNN v zátvorkách.

Metóda WkNN–FS bola porovnávaná s viacerými metódami na výber premenných, a to s vlastnou implementáciou zovšeobecnenej metódy mRMR s rôznymi mierami závislosti popísanou v kapitole 3.1 a s inými známymi metódami, ako f–score 2.3.1 a ReliefF 2.3.2. Navyše boli kvôli porovnaniu v práci implementované dve ďalšie jednoduché metódy na výber premenných, ktoré využívajú koreláciu vzdialeností (podľa 2.2.2). Prvá z nich, označovaná ako MaxDep/dCor, je založená na prístupe *Max–Dependency*. Je to pažravý algoritmus na výber takej množiny premenných, ktorá má s cieľovou premennou maximálnu hodnotu korelácie vzdialeností. Druhá metóda (označená ako Ranking/dCor) zoradí jednotlivé premenné podľa hodnoty ich korelácie vzdialeností s cieľovou premennou, pričom najvýznamnejšou je premenná s najväčšou hodnotou.

4.1 Výsledky pre syntetické dáta

Porovnávané metódy boli najprv aplikované na 5 syntetických súborov dát, a to tri súbory dát pre binárne klasifikačné úlohy, z toho jeden vysokodimenzionálny, a dva súbory dát pre regresné úlohy. Spôsob generovania súborov dát je popísaný v kapitole 2.5.1 a ich základné charakteristiky sa nachádzajú v tabuľke 4 – 1.

Tabuľka 4 – 1 Charakteristiky syntetických súborov dát

Názov	Pozorovania	Premenné	TF
Madelon100s500f	100	500	5
Madelon200s500f	200	500	5
MadelonHD	150	15 000	15
LinReg200s500f	200	500	5
Friedman200s500f	200	500	5

4.1.1 Binárna klasifikácia

Syntetické súbory dát pre binárnu klasifikáciu boli analyzované z pohľadu úspešnosti výberu TF aj z hľadiska presnosti predikcie. Pritom boli použité metódy pre riešenie klasifikačných úloh popísané v 2.4.2, a to naivný bayesovský klasifikátor (NB), metóda podporných vektorov (SVM), metóda náhodných lesov (RF) a vážené k–NN.

Pri metóde SVM bol použitý RBF *kernel* a štandardná hodnota hyperparametra *cost* (C=1). V prípade RF bolo v úlohe základného klasifikátora použitých 1 000 rozhodovacích stromov s parametrom *entropy* ako kritériom nečistoty uzla. Pri metóde k–NN bolo použité váženie 21 najbližších susedov podľa prevrátenej hodnoty vzdialenosti s výnimkou variantov metódy WkNN–FS, pri ktorých boli využité nájdené váhy premenných a príslušná hodnotiaca funkcia vzdialenosti spôsobom, ktorý bol popísaný v kapitole 3.2.6. Pri selekcii premenných metódou WkNN–FS bolo realizovaných vždy 1 000 iterácií, následne bol vybraný príslušný počet premenných a v ďalších 300 iteráciách boli upresnené ich váhy.

Dosiahnutá presnosť predikcie bola porovnávaná na základe percenta správne klasifikovaných pozorovaní (presnosť – *accuracy*), pričom bol použitý koncept krížovej validácie (*10-fold cross validation*) so stratifikovanými vzorkami.

Celkovo bolo porovnávaných 12 variantov metódy WkNN–FS a 7 iných metód pri výbere 15-tich premenných. V prípade metódy mRMR s mierami závislosti vzájomná informácia (MI) a chí–kvadrát (χ^2) boli spojité vysvetľujúce premenné zdiskrétnené spôsobom popísaným v [11] s použitím hraníc $mean - std, mean + std$.

Tabuľka 4 – 2 Výsledky pre súbor dát Madelon100s500f

Metóda	Suc.	Vybrané TF	Presnosť (Accuracy) NB	SVM	RF	k–NN
Všetky premenné			0.83	0.68	0.91	0.61
Všetky TF			0.98	1.00	0.98	0.99
mRMR/MI	0.60	5,3,2	0.99	0.98	0.97	0.92
mRMR/χ^2	0.60	5,3,2	0.99	0.99	0.98	0.94
mRMR/dCor	0.80	5,2,3,4	0.97	0.99	0.97	0.94
MaxDep/dCor	0.80	5,3,2,4	0.99	0.97	0.98	0.95
Ranking/dCor	0.80	5,3,2,4	0.94	0.98	0.97	0.95
ReliefF	0.80	5,3,2,4	0.98	0.98	0.98	0.93
f–score	0.80	5,3,2,4	0.98	0.94	0.97	0.95
WkNN(SE, L2, rbf)	0.80	5,2,3,4	0.96	0.97	0.99	**1.00**
WkNN(SE, L1, exp)	0.80	5,2,4,3	0.97	0.98	0.99	**1.00**
WkNN(SE, L2, exp)	**0.99**	5,2,4,3,1	0.98	0.99	0.97	**1.00**
WkNN(AE, L2, rbf)	0.80	5,2,4,3	0.93	0.98	0.98	**1.00**
WkNN(AE, L1, exp)	0.80	5,2,3,4	0.97	0.99	0.98	**1.00**
WkNN(AE, L2, exp)	0.80	5,2,4,3	0.98	1.00	0.99	**1.00**
WkNN(HL, L2, rbf)	0.80	5,2,3,4	0.96	0.96	0.99	**1.00**
WkNN(HL, L1, exp)	0.80	5,2,4,3	0.99	0.99	0.98	**1.00**
WkNN(HL, L2, exp)	**0.99**	5,2,4,3,1	0.97	0.97	0.99	**1.00**
WkNN(CE, L2, rbf)	0.80	5,2,4,3	0.96	0.95	0.97	**1.00**
WkNN(CE, L1, exp)	0.80	5,2,3,4	0.98	0.99	0.98	**1.00**
WkNN(CE, L2, exp)	0.80	5,2,4,3	0.98	1.00	0.98	**1.00**

Výsledky dosiahnuté pre súbor dát Madelon100s500f sú zhrnuté v tabuľke 4 – 2. Z hľadiska úspešnosti výberu TF dosiahli všetky varianty metódy WkNN–FS úspešnosť minimálne 80 % a v dvoch prípadoch až 99 %, t. j. bolo nájdených všetkých

5 TF. Ostatné metódy dosahovali úspešnosť 60–80 %, najhoršie výsledky boli u metódy mRMR/MI a mRMR/χ^2, čo mohlo byť spôsobené stratou časti informácií pri zdiskrétnení premenných.

Z pohľadu presnosti predikcie boli takmer pri všetkých metódach dosiahnuté veľmi vysoké hodnoty. Maximálna presnosť 100 % bola dosiahnutá pri použití metódy SVM v kombinácii s dvomi variantmi metódy WkNN–FS. Tieto výsledky boli dosiahnuté analogickou metodikou ako v [7], keď výber premenných bol realizovaný na celom súbore dát, a nie v rámci krížovej validácie. To spôsobuje pozitívne vychýlenie odhadu presnosti, avšak výsledky pre jednotlivé metódy výberu premenných a klasifikátory NB, SVM a RF sú porovnateľné. V prípade váženej verzie k–NN sú pre varianty metódy WkNN–FS využívané aj váhy premenných nájdené na celom súbore dát a vychýlenie odhadu môže byť väčšie. Získané výsledky však jasne ukazujú, že nájdené váhy výrazne zvýšili úspešnosť metódy váženého k–NN a viedli nielen k malej LOO chybe pri hľadaní váh, ale aj k 100 %–nej presnosti pri krížovej validácii. Z výsledkov je vidieť ešte jeden zaujímavý fakt – maximálna presnosť predikcie bola dosiahnutá v prípadoch, keď neboli nájdené všetky TF a dokonca aj pri nájdení len troch TF bola dosiahnutá presnosť predikcie veľmi vysoká.

Pri zvýšení počtu pozorovaní v súbore dát Madelon na 200 (tabuľka 1–2 v Prílohe B pre Madelon200s500f) sa úspešnosť výberu TF pri variantoch metódy WkNN–FS výrazne zvýšila – vo všetkých prípadoch bolo nájdených všetkých 5 TF. Je to pravdepodobne spôsobené tým, že pri zvýšenom počte pozorovaní už neboli nachádzané množiny premenných, ktoré by neobsahovali všetky TF a pritom by dosahovali vysokú presnosť predikcie pri váženom k–NN. Potvrdzuje to aj výraznejší odstup v presnosti predikcie medzi variantmi WkNN–FS a ostatnými metódami pri klasifikátoroch SVM a RF. Ostatné metódy totiž dosiahli pri výbere TF horšie výsledky v rozmedzí 40–80 %. Najvyššiu hodnotu pritom dosiahla metóda MaxDep/dCor.

V prípade vysokodimenzionálneho súboru dát MadelonHD bolo pri vyhodnocovaní úspešnosti výberu TF uvažovaných 450 premenných, t. j. 3 % z celkového počtu podľa metodiky z [7]. Dosiahnutá úspešnosť je nižšia v rozmedzí 40–60 %, najlepšia

Tabuľka 4 – 3 Výsledky pre súbor dát MadelonHD

Metóda	Suc. 450	Vybrané TF	Suc. 30	Presnosť (Accuracy) NB	SVM	RF	k–NN
Všetky premenné				0.59	0.56	0.62	0.58
Všetky TF				0.87	0.96	0.91	0.92
mRMR/MI	0.40	14,5,4,1,9,3	0.20	0.94	0.92	0.94	0.84
mRMR/χ^2	0.40	14,5,4,1,9,3	0.20	0.95	0.91	0.93	0.87
Ranking/dCor	0.53	5,14,4,1,13,11,9,7	0.27	0.93	0.95	0.92	0.94
ReliefF	0.40	14,5,4,13,1,7	0.33	0.91	0.87	0.90	0.88
f–score	0.47	5,14,4,1,13,11,9	0.27	0.96	0.98	0.94	0.97
WkNN(SE, L2, rbf)	0.53	5,14,4,1,13,11,7,9	0.27	0.97	0.96	0.93	**0.99**
WkNN(SE, L1, exp)	0.53	5,14,4,1,13,7,11,9	0.33	0.96	0.95	0.93	**0.99**
WkNN(SE, L2, exp)	**0.60**	14,5,4,1,7,10,8,2,6	**0.60**	0.95	0.99	0.95	**0.99**
WkNN(AE, L2, rbf)	0.53	5,14,4,1,13,11,7,9	0.27	0.95	0.96	0.93	0.98
WkNN(AE, L1, exp)	0.53	5,14,4,1,13,7,11,9	0.33	0.96	0.95	0.93	0.97
WkNN(AE, L2, exp)	0.40	5,14,4,1,7,10	0.40	0.89	0.93	0.92	0.94
WkNN(HL, L2, rbf)	0.53	5,14,4,1,13,11,7,9	0.27	0.95	0.96	0.93	**0.99**
WkNN(HL, L1, exp)	0.53	5,14,4,1,13,7,11,9	0.33	0.96	0.95	0.93	0.97
WkNN(HL, L2, exp)	0.40	5,14,4,1,7,10	0.40	0.90	0.95	0.91	0.94
WkNN(CE, L2, rbf)	0.53	5,14,4,1,13,11,7,9	0.27	0.97	0.97	0.93	0.97
WkNN(CE, L1, exp)	0.53	5,14,4,1,13,7,11,9	0.33	0.96	0.95	0.92	**0.99**
WkNN(CE, L2, exp)	0.47	14,5,4,1,7,10,8	0.47	0.94	0.99	0.95	0.96

hodnota bola dosiahnutá pri jednom z variantov metódy WkNN–FS.

Pri vyhodnocovaní presnosti klasifikátorov bol v tomto prípade použitý výber 30–tich premenných. Podobne ako v prípade Madelon200s500f aj v tomto prípade sa pri použití metód pre výber premenných výrazne zvýšila presnosť klasifikátorov. Najlepšie hodnoty boli dosiahnuté pri dvoch variantoch metódy WkNN–FS, a to až 99 %. Prekvapujúci je však fakt, že pre metódu f-score sú výsledky všetkých štyroch klasifikátorov lepšie ako v prípade použitia všetkých 15–tich TF. Pritom táto metóda v rámci 30–tich vybraných premenných našla len 4 TF. Podobný fenomén bol popísaný aj v [7] pri vyhodnocovaní presnosti predikcie na súbore dát CorrAL, keď pri malom počte pozorovaní bola najvyššia presnosť predikcie dosiahnutá pri výbere s jedinou TF. Podľa autorov je to spôsobené tým, že v prípade malého počtu

pozorovaní a veľkého počtu náhodne generovaných premenných môžu pri vytváraní súboru dát náhodne vzniknúť ďalšie premenné, ktoré obsahujú informačnú hodnotu pre klasifikátory. Tým zároveň môže byť vysvetlený neúspech metódy WkNN–FS pri hľadaní ďalších TF, nakoľko zrejme iné náhodne vzniknuté premenné umožnili dosiahnuť pri váženom k–NN vyššiu presnosť predikcie ako pri výbere ďalších TF.

4.1.2 Regresia

V prípade syntetických súborov dát s numerickou cieľovou premennou boli pre odhad presnosti predikcie použité metódy popísané v kapitole 2.4.2, a to základná metóda pre regresné úlohy – lineárna regresia (LR) a pre regresiu upravená verzia metódy podporných vektorov (SVM), metódy náhodných lesov (RF) a váženého k–NN. Tieto metódy boli použité s rovnakými parametrami ako v prípade binárnej klasifikácie s výnimkou metódy SVM, kde bola modifikovaná hodnota hyperparametra *cost* nastavením na hodnotu rozptylu cieľovej premennej.

Pri meraní presnosti predikcie boli ako miery chyby použité hodnoty RMSE (*Root Mean Squared Error*) a MAE. Vo výsledkoch je kvôli prehľadnosti uvedená len hodnota RMSE prepočítaná na koeficient R^2, ktorý určuje, aké percento rozptylu cieľovej premennej bolo vysvetlené danou predikciou (maximálna hodnota je 1.00, t. j. 100 %) [4]. Pre určenie presnosti predikcie bol použitý koncept krížovej validácie (*10–fold cross validation*) pričom stratifikácia sa dosiahla tak, že pozorovania boli rozdelené do piatich skupín podľa percentilov pre hodnoty 20, 40, 60, 80 a 100.

Celkovo bolo porovnávaných 9 variantov metódy WkNN–FS a 10 iných metód pre výber 15–tich premenných. V prípade metódy mRMR/MI a mRMR/χ^2 boli spojité vysvetľujúce premenné aj cieľová premenná zdiskrétnené rovnako ako pri binárnej klasifikácii. Rovnaká úprava cieľovej premennej bola uplatnená aj v prípade metódy ReliefF a f–score, ktoré sú pôvodne určené pre klasifikáciu. Navyše bolo zaradené jednoduché zoradenie premenných podľa Pearsonovho korelačného koeficientu (Ranking/Pearson). Podrobné výsledky dosiahnuté na regresných syntetických súboroch dát sú zhrnuté v tabuľkách 1–4 a 1–5 v Prílohe B.

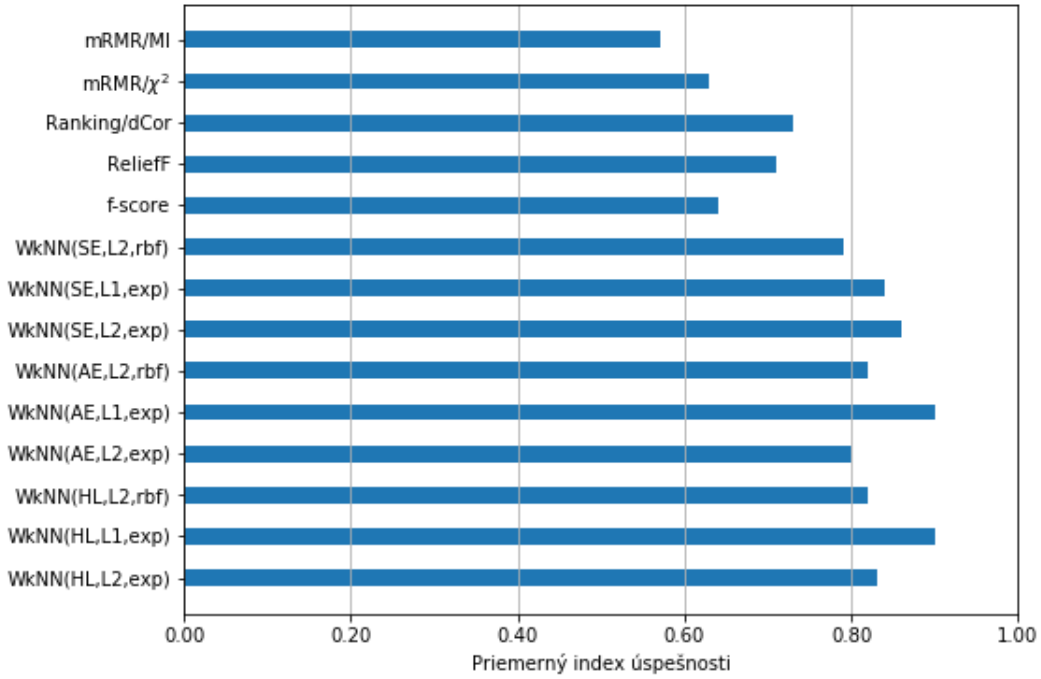
Z hľadiska úspešnosti výberu TF v prípade súboru dát LinReg200s500f dosiahli najlepšie výsledky varianty metódy WkNN–FS, ktoré vo všetkých prípadoch okrem jedného našli 5 TF. Z ostatných metód sa im vyrovnala len metóda MaxDep/dCor, najslabší výsledok naopak dosiahla metóda ReliefF. Presnosť predikcie bez výberu premenných bola veľmi nízka s výnimkou metódy RF, použitím metód pre výber premenných sa výrazne zvýšila. Vzhľadom na lineárnu závislosť cieľovej premennej boli dosiahnuté najlepšie výsledky pri LR, ktorá pri výbere všetkých piatich relevantných premenných dosiahla nulovú chybu ($R^2 = 1.0$). Z nelineárnych metód bola dosahovaná najlepšia presnosť v prípade SVM v kombinácii s niektorými variantmi metódy WkNN–FS.

Pre súbor dát Friedman200s500f pri výbere premenných až 5 metód identifikovalo všetkých 5 TF, z toho 3 varianty metódy WkNN–FS na prvých piatich miestach. Úspešnosť výberu relevantných premenných bola vysoká, všetky metódy vybrali aspoň 4 TF. Metóda ReliefF bola na rozdiel od predchádzajúceho regresného súboru dát veľmi úspešná. V prípade bez výberu premenných bola presnosť predikcie na súbore dát Friedman200s500f mimoriadne nízka, ako použiteľnú možno označiť len metódu RF. Prípad použitia všetkých TF ukazuje, že ako najvhodnejšia sa pre daný súbor dát javí metóda SVM. Pri výbere 15-tich premenných sa však nepodarilo priblížiť s SVM k danej presnosti, a to ani v prípadoch, že výber obsahoval všetky TF – prítomnosť ďalších irelevantných premenných znižovala presnosť predikcie. Na druhej strane, v prípade LR sa podarilo dosiahnuť presnosť dokonca o málo vyššiu ako v prípade použitia všetkých TF (pritom boli vybrané len 4 TF).

Pri porovnávaní dvoch prístupov s použitím Pearsonovho korelačného koeficientu, a to Ranking/Pearson a mRMR/Pearson, je vidieť, že na oboch skúmaných regresných súboroch dát našli rovnaký počet TF a dosiahli veľmi podobnú presnosť predikcie. Uvedená skutočnosť pravdepodobne súvisí s tým, že jednotlivé TF sú v oboch prípadoch nezávislé.

4.1.3 Porovnanie metód z hľadiska indexu úspešnosti

Obrázok 4 – 1 zobrazuje porovnanie jednotlivých metód výberu premenných z hľadiska úspešnosti pri výbere známych relevantných premenných. Pre každú metódu bol vypočítaný priemerný index úspešnosti z indexov úspešnosti dosiahnutých na piatich vyššie uvedených syntetických súboroch dát, a to Madelon100s500f, Madelon200s500f, LinReg200s500f, Friedman200s500f a MadelonHD.



Obrázok 4 – 1 Porovnanie z hľadiska indexu úspešnosti

V celkovom porovnaní sa ako najúspešnejšie javia varianty metódy WkNN–FS, z ostatných metód najlepšie výsledky dosiahli metódy Ranking/dCor a ReliefF. Treba poznamenať, že ďalšie metódy výberu premenných s využitím korelácie vzdialeností nie sú v prehľade uvedené kvôli ich veľkej výpočtovej náročnosti pre výber 450 premenných v prípade súboru dát MadelonHD.

4.2 Výsledky pre reálne dáta

Okrem syntetických súborov dát bola navrhnutá metóda hodnotená na deviatich vysokodimenzionálnych reálnych súboroch dát pre binárnu klasifikáciu, ktorých charakteristiky sú popísané v tabuľke 2 – 1. V prípade reálnych súborov dát však nie je možné vyhodnotiť úspešnosť výberu TF, preto bol hodnotený iba vplyv výberu premenných na presnosť binárnej klasifikácie.

Hlavnou odlišnosťou oproti binárnej klasifikácii na umelých súboroch dát je skutočnosť, že niektoré zo súborov dát nie sú vybalansované, t. j. početnosť jednotlivých tried je veľmi rozdielna. To platí predovšetkým pre súbory dát Tian a Alon, v menšej miere aj Burczynski, Chin a Golub. Vzhľadom na to bola okrem vyhodnocovania percentuálnej presnosti predikcie (*accuracy*) hodnotená aj miera F_1 *score*.

Pri meraní bola použitá rovnaká metodika založená na krížovej validácii a rovnaké metódy pre binárnu klasifikáciu s identickým nastavením hyperparametrov ako pre syntetické súbory dát. V prípade metódy WkNN–FS bola zvolená eliminačná metóda popísaná v kapitole 3.2.9 s 300 iteráciami v prvej fáze, orezaním na 1 000 najvýznamnejších premenných a ich postupnou elimináciou na 30 výsledných premenných. Porovnávaných bolo celkom 12 variantov metódy WkNN–FS a 7 ostatných metód pre výber premenných. Oproti syntetickým súborom dát bol kvôli porovnaniu s mRMR/MI pridaný variant s *Max–Dependency* prístupom a vzájomnou informáciou (MaxDep/MI). Na rozdiel od MaxDep/dCor boli vyberané premenné len dovtedy, kým zvyšovali hodnotu vzájomnej informácie vybranej množiny premenných s cieľovou premennou. V prípade metód s použitím MI boli vysvetľujúce premenné zdiskrétnené rovnakou metódou ako v prípade syntetických súborov dát.

Výsledky pre súbor dát Alon sú uvedené v tabuľke 4 – 4. Ako vidno, uplatnenie metód výberu premenných výrazne zvyšuje presnosť predikcie vo všetkých prípadoch. Najlepšie výsledky dosiahli varianty metódy WkNN–FS v kombinácii s klasifikátorom SVM, a to z pohľadu oboch mier: presnosť (*Acc.*) aj F_1 *score*. Z ostatných metód dosiahla najlepšie výsledky metóda MaxDep/dCor v kombinácii s klasifiká-

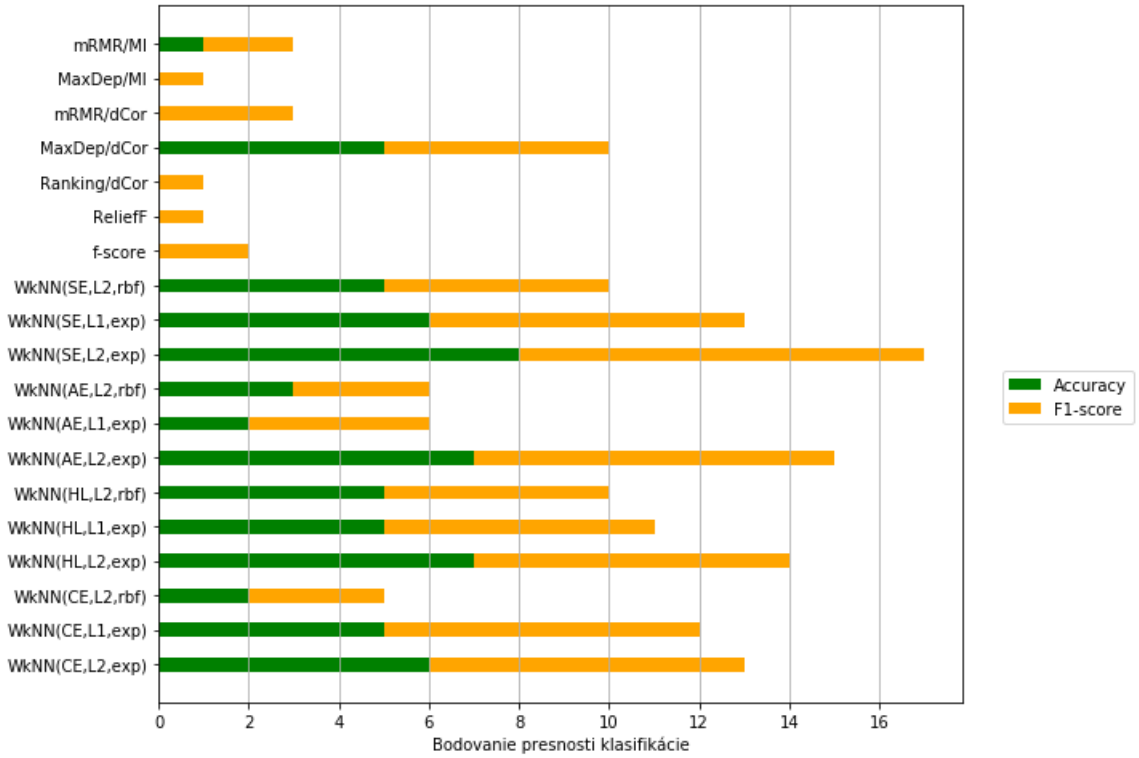
Tabuľka 4 – 4 Výsledky pre súbor dát Alon

Metóda	NB		SVM		RF		k–NN	
	Acc.	F_1	Acc.	F_1	Acc.	F_1	Acc.	F_1
Všetky premenné	0.61	0.59	0.79	0.53	0.82	0.69	0.68	0.10
mRMR/MI	0.89	0.83	0.89	0.83	0.87	0.81	0.84	0.73
MaxDep/MI	0.79	0.72	0.82	0.69	0.85	0.79	0.80	0.58
mRMR/dCor	0.90	0.86	0.89	0.84	0.87	0.81	0.87	0.80
MaxDep/dCor	0.92	0.89	0.90	0.86	0.87	0.81	0.89	0.76
Ranking/dCor	0.87	0.81	0.87	0.80	0.85	0.78	0.85	0.76
ReliefF	0.85	0.78	0.87	0.80	0.87	0.81	0.85	0.72
f–score	0.87	0.81	0.87	0.80	0.87	0.81	0.85	0.76
WkNN(SE, L2, rbf)	0.86	0.82	0.90	0.84	0.87	0.79	0.98	0.97
WkNN(SE, L1, exp)	0.84	0.78	0.89	0.81	0.89	0.84	0.98	0.97
WkNN(SE, L2, exp)	0.87	0.85	0.92	0.88	0.85	0.77	**1.00**	**1.00**
WkNN(AE, L2, rbf)	0.78	0.74	0.89	0.81	0.87	0.80	0.98	0.97
WkNN(AE, L1, exp)	0.81	0.75	0.90	0.84	0.87	0.80	0.97	0.95
WkNN(AE, L2, exp)	0.86	0.84	0.94	0.91	0.89	0.82	0.98	0.97
WkNN(HL, L2, rbf)	0.83	0.77	0.92	0.85	0.87	0.80	0.98	0.97
WkNN(HL, L1, exp)	0.84	0.81	0.89	0.82	0.89	0.84	0.98	0.97
WkNN(HL, L2, exp)	0.87	0.85	0.92	0.88	0.87	0.76	0.98	0.97
WkNN(CE, L2, rbf)	0.91	0.88	0.89	0.81	0.84	0.76	0.98	0.97
WkNN(CE, L1, exp)	0.78	0.72	0.94	0.90	0.87	0.80	0.98	0.97
WkNN(CE, L2, exp)	0.89	0.83	0.90	0.84	0.85	0.78	0.98	0.97

tormi NB a čiastočne aj SVM. Metóda váženého k–NN ukazuje výrazný vplyv nájdených váh premenných pri WkNN–FS na zvýšenie presnosti predikcie, v jednom prípade bola dosiahnutá pri krížovej validácii dokonca úspešnosť 100 %. Vzhľadom na to, že pri vyhľadávaní váh premenných bol použitý celý súbor dát, tieto výsledky nemožno považovať za priamo porovnateľné s výsledkami ostatných klasifikátorov.

Podrobné výsledky pre všetky reálne súbory dát sú vzhľadom na ich rozsah uvedené v Prílohe B. U niektorých súborov dát boli dosiahnuté veľmi vysoké presnosti predikcie, a to najmä u súborov Gordon, Golub a Burczynski. V prípade súboru Gordon to vzhľadom na pozorovanie uvádzané v kapitole 3.1.2 pre *Max-Dependency* prístup s použitím vzájomnej informácie nie je prekvapujúce. Na druhej strane boli výsledky podstatne horšie pre nevybalansované súbory dát Tian, Alon a Pomeroy.

Výsledky uvedené pri súbore dát Alon sa viac–menej potvrdili aj u ďalších reálnych súborov dát. Kvôli prehľadnosti bolo spracované sumárne porovnanie metód tak, že bolo uvažovaných všetkých 9 súborov dát a 3 klasifikátory – NB, SVM a RF, t. j. celkom 27 prípadov. Pre každý prípad bol metóde pre výber premenných pridelený bod vtedy, ak sa pri danej kombinácii súbor dát/klasifikátor umiestnila na prvom mieste (pritom mohlo byť aj viacero víťazov). Body boli počítané osobitne pre mieru presnosť (*accuracy*) a mieru F_1 *score*. Sumárne výsledky sú zobrazené na obrázku 4 – 2. Z tohto porovnania vychádzajú víťazne niektoré varianty metódy WkNN–FS, a to hlavne s použitím metriky L2 a hodnotiacej funkcie vzdialenosti *exp*. Na druhej strane štandardné metódy mRMR/MI, ReliefF a f–score sa nepresadili.



Obrázok 4 – 2 Porovnanie z hľadiska presnosti predikcie na reálnych súboroch dát

Pri porovnávaní mRMR/MI a MaxDep/MI si možno všimnúť, že až na súbory Gordon a Chin boli výsledky metódy MaxDep/MI horšie ako mRMR/MI a ostatných vyhodnocovaných metód. Problémom pri použití MaxDep/MI je, že pomocou

kontingenčnej tabuľky sa odhaduje pravdepodobnosť možných stavov. Už v prípade použitia piatich premenných má úplná kontingenčná tabuľka $3^5 * 2 = 486$ položiek, čo je viac ako počet pozorovaní v súbore dát. Počet pozorovaní v skúmaných reálnych súboroch dát je teda príliš malý pre získanie spoľahlivého odhadu pravdepodobnosti jednotlivých stavov a následného výpočtu vzájomnej informácie. Tieto výsledky potvrdzujú výhrady uvedené v [8], že metóda MaxDep/MI je použiteľná len pre výber malého množstva relevantných premenných a vyžaduje väčší počet pozorovaní.

Prekvapivým súperom pre metódu WkNN–FS bola metóda MaxDep/dCor, ktorá bola najlepšia z ostatných metód a dosiahla najvyššiu presnosť predikcie na súboroch dát Tian a Singh. Pritom pôvodne bolo zamýšľané použitie tejto metódy len v prípade regresie. Pri porovnaní troch prístupov k využitiu korelácie vzdialeností – Ranking, mRMR a *Max-Dependency* sa v zhode s očakávaním ukazuje ako najslabší Ranking a ako najlepší prístup *Max-Dependency*. Istou slabinou tejto metódy je jej horšia využiteľnosť v prípade väčšieho počtu pozorovaní kvôli potrebe výpočtu dištančnej matice s kvadratickou zložitosťou. Možno preto uvažovať o jej využití v tvare skupinovej metódy s opakovaným náhodným výberom menších vzoriek pozorovaní.

4.3 Porovnanie stability metód pre výber premenných

Porovnanie stability bolo vykonané pre 3 vybrané štandardné metódy pre výber premenných a 3 vybrané varianty metódy WkNN. Výpočty boli realizované na štyroch syntetických súboroch dát — Madelon200s500f (skrátene Mad), LinReg200s500f (LReg), Friedman200s500f (Fried), MadelonHD (MadHD) a štyroch reálnych súboroch dát s rôznym počtom premenných – Alon s 2 000 premennými, Pomeroy (Pom) so 7 128 premennými, Singh s 12 600 premennými a Chin s 22 215 premennými. Pre každú metódu a súbor dát bolo vykonaných 100 spustení. Keďže bol vzhľadom na výpočtovú náročnosť zvolený menší počet meraní ako napr. v [41], bolo potrebné venovať zvýšenú pozornosť výberu vzoriek. Preto bola využitá analogická metodológia ako v [18], keď vzorky boli generované s využitím krížovej validácie (*10-fold strati-*

fied cross validation). Tento prístup zabezpečil, že každá vzorka obsahovala 90 % z celkového počtu pozorovaní a každé pozorovanie sa vyskytlo vo vzorke 90–krát.

V prvom kroku bola porovnaná stabilita pri výbere malého počtu premenných – bola zvolená hodnota 5, čo v prípade niektorých syntetických súborov dát zodpovedalo počtu TF. V tabuľke 4 – 5 sú uvádzané hodnoty *Kuncheva indexu* popísaného v kapitole 2.4.3 pre jednotlivé metódy a súbory dát. Možno si všimnúť, že varianty WkNN–FS dosahujú vysokú stabilitu na syntetických dátach. Na reálnych dátach sú výsledky veľmi rozdielne, kým napríklad stabilita na súbore dát Singh je vysoká, na súbore Alon je omnoho nižšia ako v prípade metód ReliefF a f–score.

Tabuľka 4 – 5 Kuncheva index pre výber 5 premenných

Metóda/Súbor dát	Mad	LReg	Fried	MdHD	Alon	Pom	Singh	Chin
ReliefF	0.7104	0.6413	0.8530	0.6181	0.8298	0.4319	0.6880	0.6647
f–score	0.5118	0.6931	0.8292	0.5858	0.8214	0.5605	0.7665	0.8681
mRMR/MI	0.3704	0.6586	0.7872	0.4941	0.4129	0.1630	0.3764	0.6839
WkNN(SE, L2, rbf)	0.9960	0.8369	0.8091	0.6783	0.4428	0.4826	0.8382	0.6655
WkNN(HL, L1, exp)	1.0000	0.9386	0.9579	0.7640	0.4821	0.4114	0.7740	0.6636
WkNN(HL, L2, exp)	1.0000	0.8054	0.8164	0.7567	0.6537	0.3519	0.8306	0.6703

V druhom kroku boli vypočítané hodnoty *Kuncheva indexu* pre väčší počet premenných, ktorý zodpovedal počtu použitému pri vyhodnocovaní úspešnosti výberu a presnosti predikcie – 15, resp. 30 v prípade vysokodimenzionálnych súborov dát. Výsledky sú uvedené tabuľke 4 – 6 a možno si všimnúť, že pri výbere 15 premenných na syntetických dátach stabilita WkNN–FS prudko klesla a väčšinou bola prekonaná metódami f–score a ReliefF. To znamená, že kým varianty WkNN–FS pri malom počte premenných vyberajú stabilné množiny obsahujúce väčšinou TF, v prípade výberu ďalších väčšinou irelevantných premenných je stabilita nízka. Táto vlastnosť nemusí byť nutne nevýhodou, teoreticky by mohla byť využitá na odlíšenie relevantných a irelevantných premenných v prípade skupinových metód. Tiež si možno všimnúť, že v prípade súboru dát MadelonHD a WkNN(HL, L2, exp) nie je hodnota indexu vypočítaná. Dôvodom je, že táto metóda v niektorých prípadoch vybrala

menej ako 30 premenných (všetky ostatné váhy boli nulové) a definícia *Kuncheva indexu* predpokladá len porovnávanie množín s rovnakým počtom premenných.

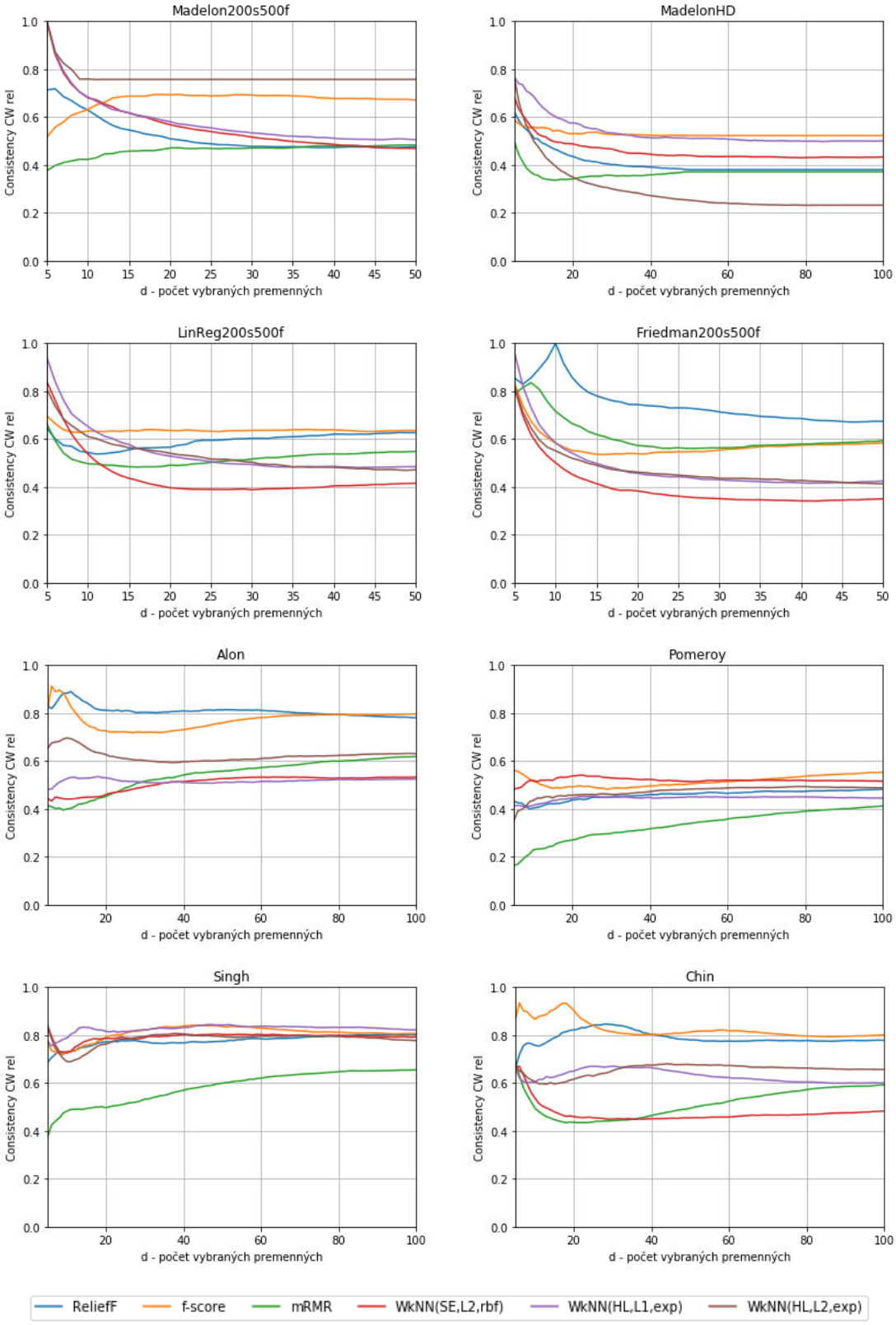
Tabuľka 4 – 6 Kuncheva index pre výber 15/30 premenných

Metóda/Súbor dát	Mad	LReg	Fried	MdHD	Alon	Pom	Singh	Chin
ReliefF	0.5415	0.5510	0.7785	0.4002	0.8029	0.4469	0.7712	0.8443
f–score	0.6843	0.6326	0.5317	0.5263	0.7183	0.4821	0.8218	0.8138
mRMR/MI	0.4520	0.4794	0.6151	0.3553	0.5130	0.2947	0.5318	0.4422
WkNN(SE, L2, rbf)	0.6132	0.4306	0.4077	0.4660	0.4909	0.5272	0.7891	0.4489
WkNN(HL, L1, exp)	0.6142	0.5731	0.4934	0.5333	0.5088	0.4482	0.8225	0.6698
WkNN(HL, L2, exp)	N/A	0.5591	0.4858	N/A	0.6005	0.4585	0.7941	0.6507

Hodnoty indexu uvedené v prvej a druhej tabuľke nie sú priamo porovnateľné, nakoľko hodnoty *Kuncheva indexu* sú ovplyvňované počtom vyberaných premenných. Pre rôzne počty vyberaných premenných však možno využiť relatívnu verziu váženého *consistency* indexu (CW_{rel}). Keďže všetky uvažované metódy môžu byť interpretované aj ako *ranking*, možno urobiť výber prvých 50 alebo 100 premenných podľa jednotlivých metód a potom vyhodnotiť stabilitu výberu z pohľadu podobnosti vybraných podmnožín pre rôzny počet vyberaných premenných. Výsledky sú zobrazené v tvare grafov v tabuľke 4 – 7. V grafickom zobrazení je možné lepšie identifikovať a potvrdiť vyššie uvedené pozorovania. Zároveň sa ukazuje niekoľko zaujímavých faktov, napr. konštantná časť grafu pri Madelon200s500f a WkNN(HL, L2, exp) spôsobená tým, že táto metóda vyberala len množiny s malým počtom premenných. V prípade súboru dát Friedman200s500f metóda ReliefF vyberala vždy rovnakých 10 premenných ($CW_{rel} = 1.0$).

V prípade niektorých reálnych súborov dát ako Pomeroy a Singh mala metóda mRMR/MI výrazne nižšiu stabilitu ako ostatné metódy. Je otázne, či to nesúvisí s faktom, že na rozdiel od ostatných porovnávaných metód sa u nej vykonávalo zdiskrétnenie vysvetľujúcich premenných. V tejto súvislosti treba poznamenať, že zdiskrétnenie premenných rovnako ako $\langle 0, 1\rangle$–normalizácia pri ReliefF a štandardizácia pri WkNN–FS bolo vykonávané na každej vzorke nezávisle.

Tabuľka 4 – 7 Porovnanie stability metód pre vybrané súbory dát



5 Záver

Hlavným prínosom práce je návrh novej metódy pre výber premenných, ktorá vychádza zo špeciálnej verzie známej metódy strojového učenia označovanej ako k–NN (k najbližších susedov). Navrhnutá úprava k–NN využíva ohodnotenie susedov podľa vzdialenosti, váženie premenných a koncept *leave–one–out* (LOO) pre určenie chyby. Hľadanie optimálnych váh premenných je inšpirované prístupmi známymi z neurónových sietí a realizované metódou najväčšieho spádu v štandardnom alebo stochastickom variante. Pre získanie finálneho výberu premenných a ich váh je možné použiť rekurzívnu elimináciu alebo vhodný spôsob regularizácie.

Metóda predstavuje všeobecný rámec so širokými možnosťami parametrizácie:

- použitím rôznych metrík (euklidovská, manhattanská, atď.),
- využitím rôznych funkcií chyby (MSE, MAE, Huberova funkcia a pod.),
- rôznymi hodnotiacimi funkciami vzdialenosti (RBF *kernel* a iné).

Okrem výberu premenných metóda poskytuje váhy premenných, ktoré možno interpretovať ako transformáciu priestoru, resp. úpravu definície vzdialenosti. Nájdené váhy sa dajú využiť na zvýšenie presnosti predikcie, napríklad pri použití metódy k–NN pre klasifikáciu alebo regresiu.

Navrhnutá metóda je použiteľná pri regresii aj binárnej klasifikácii v kombinácii s rôznymi metódami strojového učenia. Možno ju použiť pre súbory dát s vysokým počtom premenných a malým počtom pozorovaní, ktoré sa často vyskytujú v oblasti biomedicíny. Je využiteľná aj pre súbory dát s väčším počtom pozorovaní aplikovaním stochastického variantu metódy najväčšieho spádu.

Práca obsahuje podrobné porovnanie novej metódy s niekoľkými známymi metódami pre výber premenných na syntetických aj reálnych súboroch dát. Výsledky získané pre viacero variantov novej metódy ukazujú, že z pohľadu úspešnosti výberu relevantných premenných aj vplyvu na presnosť predikcie sú plne porovnateľné s existujúcimi metódami. Experimentálne výsledky obsahujú aj analýzu stability vybraných metód.

Ďalším prínosom práce je implementácia zovšeobecnej verzie algoritmu mRMR s možnosťou použitia rôznych mier závislosti premenných, ako aj nový pohľad na formuláciu mRMR ako optimalizačnej úlohy a na vzťah mRMR a tzv. *Max-Dependency* prístupu. Pritom sa javia ako zaujímavé dosiahnuté výsledky pri použití pomerne novej miery závislosti, ktorou je korelácia vzdialeností.

Nevýhodami novej metódy sú hlavne vyššia výpočtová náročnosť v porovnaní s jednoduchšími filtrami pre výber premenných a možnosť uviaznutia v lokálnom minime pri metóde najväčšieho spádu.

Pre navrhnutú metódu možno hľadať ďalšie vhodné konfigurácie pre rôzne typy reálnych súborov dát s prípadným použitím iných metrík, nových funkcií pre výpočet chyby a pod. Zaujímavá by bola podrobnejšia analýza dosiahnutého zvýšenia presnosti predikcie pri použití špeciálnych verzií metódy k–NN a nájdených optimálnych váh. Ďalším smerom výskumu by mohlo byť hľadanie takých účelových funkcií alebo úprav algoritmu, ktoré by odstránili alebo znížili riziko uviaznutia metódy najväčšieho spádu v lokálnom minime. Popísaný koncept regularizácie umožňuje ďalší rozvoj návrhom nových regularizačných funkcií a metód pre optimálne nastavenie regularizačného parametra.

Literatúra

[1] HAND, D., MANNILA, H., SMYTH, P. *Principles of Data Mining.* Cambridge : Massachusetts Institute of Technology, 2001. 546 p. ISBN 0–262–08290–X.

[2] HAN, J., KAMBER, M. *Data Mining.* San Francisco : Elsevier Inc., 2006. 743 p. ISBN 978–1–55860–901–3.

[3] MITCHELL, T. M. *Machine Learning.* Maidenhead U. K. : McGraw–Hill, 1997. 414 p. ISBN 0–07–115467–1.

[4] RASCHKA, S. *Python Maschine Learning.* Birmingham : Packt Publishing Ltd., 2016. 425 p. ISBN 978–1–78355–513–0.

[5] ZAKI, M. J., MEIRA, W. *Data Mining and Analysis* : Fundamental Concepts and Algorithms. Cambridge : University Press, 2014. 593 p. ISBN 978–0–521–76633–3.

[6] GUYON, I., ELISEFF, A. An Introduction to Variable and Feature Selection. In *Journal of Machine Learning Research.* USA : JMLR, Inc. and Microtome Publishing. ISSN 1533–7928, 2003, vol. 3, p. 1157–1182.

[7] BOLÓN-CANEDO, V., SÁNCHEZ-MARONO, N., ALONSO-BETANZOS, A. *Feature Selection for High-Dimensional Data.* Heidelberg : Springer International Publishing, 2015. 147 p. ISBN 978–3–319–21857–1.

[8] PENG, H., LONG, F., DING, CH. Feature Selection Based on Mutual Information: Criteria of Max-Dependency, Max-Relevance, and Min-Redundancy. In *IEEE Transactions on Pattern Analysis and Machine Intelligence.* ISSN 0162–8828, 2005, vol. 27, no. 8, p. 1226–1238.

[9] LAZAR, C. et al. A Survey on Filter Techniques for Feature Selection in Gene Expression Microarray Analysis. In *IEEE/ACM Transactions on Computati-*

onal Biology and Bioinformatics. Los Alamos, CA, USA : IEEE Computer Society Press. ISSN 1545–5963, 2012, vol. 9, no. 4, p. 1106–1117.

[10] ANG, J. C., MIRZAL, A., HARON, H., HAMED, H. N. Supervised, Unsupervised, and Semi-Supervised Feature Selection: A Review on Gene Selection. In *IEEE/ACM Transactions on Computational Biology and Bioinformatics.* Los Alamos, CA, USA : IEEE Computer Society Press. ISSN 1545–5963, 2016, vol. 13, no. 5, p. 971–989.

[11] DING, CH., PENG, H. Minimum Redundancy Feature Selection from Microarray Gene Expression Data. In *Journal of Bioinformatics and Computational Biology.* Berkeley : Imperial College Press. ISSN 1757–6334, 2004, Vol. 3, No. 2, p. 185–205.

[12] BERRENDERO, J. R., CUEVAS, A., TORRECILLA, J. L. The mRMR Variable Selection Method: a Comparative Study for Functional Data. In *Journal of Statistical Computation and Simulation.* USA : Taylor & Francis. ISSN 1563–5163, 2016, Vol. 86, No. 5, p. 891–907.

[13] SZÉKELY, G. J., RIZZO, M. L., BAKIROV, N. K. Measuring and Testing Dependence by Correlation of Distances. In *The Annals of Statistics.* The Institute of Mathematical Statistics. ISSN 0090–5364, 2007, Vol. 35, No. 6, p. 2769–2794.

[14] DASH, M., LIU, H. Feature Selection for Classification. In *Intelligent Data Analysis.* Amsterdam, The Netherlands : IOS Press. ISSN 1088–467X, 1997, vol. 1, No. 3, p. 131–156.

[15] WEIR, B. Estimating F-Statistics: A Historical View. In *Philosophy of Science.* Chicago : The University of Chicago Press Journals. ISSN 0031-8248, 2003, vol. 79, no. 5, p. 637–643.

[16] ROBNIK–ŠIKONJA, M., KONONENKO, I. Theoretical and Empirical Ana-

lysis of ReliefF and RReliefF. In *Machine Learning.* The Netherlands : Kluwer Academic Publishers. ISSN 0885-6125, 2003, vol. 53, no. 1, p. 23–69.

[17] RISH, I. An Empirical Study of the Naive Bayes Classifier. In *IJCAI 2001 Workshop on Empirical Methods in Artificial Intelligence.* New York : IBM Research Division. 2001, vol. 3, p. 41–46.

[18] KALOUSIS, A., PRADOS, J., HILARIO, M. Stability of Feature Selection Algorithms: a Study on High–dimensional Spaces. In *Knowl. Inf. Syst.* 2007, vol. 12, no. 1 p. 95–116.

[19] KUNCHEVA, L. I. A Stability Index for Feature Selection. In *Proceedings of the 25th IASTED International Multi–Conference on Artificial Intelligence and Applications, AIAP07.* Anaheim, CA, USA : ACTA Press. 2007 p. 390–395.

[20] SOMOL, P., NOVOVIČOVÁ, J. Evaluating Stability and Comparing Output of Feature Selectors that Optimize Feature Subset Cardinality. In *IEEE Trans. Pattern Anal. Mach. Intell.* 2010, vol. 32, no. 11 p. 1921–1939.

[21] SOMOL, P., NOVOVIČOVÁ, J. Evaluating the Stability of Feature Selectors that Optimize Feature Subset Cardinality. In *N. da Vitoria Lobo, T. Kasparis, F. Roli, J. Kwok, M. Georgiopoulos, G. Anagnostopoulos, M. Loog(Eds.), Structural, Syntactic, and Statistical Pattern Recognition, Lecture Notes in Computer Science,.* Berlin, Heidelberg : Springer. 2008, vol. 5342 p. 956–966.

[22] DUA, D. and KARRA TANISKIDOU, E. *UCI Machine Learning Repository* [online]. 2017. [cit. 2017–12–19]. Irvine, CA: University of California, School of Information and Computer Science. 2017. Dostupné na internete: `http://archive.ics.uci.edu/ml`

[23] ALON, U. et al. Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays. In *Proceedings of the National Academy of Sciences.* 1999, vol. 96, no. 12, p. 6745–6750.

[24] POMEROY, S. L. et al. Prediction of central nervous system embryonal tumour outcome based on gene expression. In *Nature.* 2002, vol. 415, no. 6870, p. 436–442.

[25] GOLUB, T. R. et al. Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. In *Science.* 1999, vol. 286, no. 5439, p. 531–537.

[26] GORDON, G. J. G. et al. Translation of Microarray Data into Clinically Relevant Cancer Diagnostic Tests Using Gene Expression Ratios in Lung Cancer and Mesothelioma. In *Cancer Research.* 2002, vol. 62, no. 17, p. 4963–4967.

[27] SINGH, D. et al. Gene expression correlates of clinical prostate cancer behavior. In *ISRN Cancer Cell.* 2002, vol. 1, no. 2, p. 203–209.

[28] TIAN, E. et al. The Role of the Wnt-Signaling Antagonist DKK1 in the Development of Osteolytic Lesions in Multiple Myeloma. In *New England Journal of Medicine.* 2003, vol. 349, no. 26, p. 2483–2494.

[29] CHIN, K. et al. Genomic and transcriptional aberrations linked to breast cancer pathophysiologies. In *Cancer Cell.* 2006, vol. 10, no. 6, p. 529–541.

[30] CHOWDARYI, D. et al. Prognostic Gene Expression Signatures Can Be Measured in Tissues Collected in RNAlater Preservative. In *The Journal of Molecular Diagnostics.* 2006, vol. 8, no. 1, p. 31–39.

[31] BURCZYNSKI, M. E. et al. Molecular Classification of Crohnś Disease and Ulcerative Colitis Patients Using Transcriptional Profiles in Peripheral Blood Mononuclear Cells. In *The Journal of Molecular Diagnostics.* 2006, vol. 8, no. 1, p. 51–61.

[32] Python Software Foundation. *Python Language Reference* [online]. Verzia 3.6. [cit. 2017–12–19]. Dostupné na internete: `http://www.python.org`

[33] PENG, H. *mRMR (minimum Redundancy Maximum Relevance Feature Selection)* [online]. 2017. [cit. 2017–12–19]. Dostupné na internete: http://home.penglab.com/proj/mRMR/

[34] WILSON, D. R., MARTINEZ, T. R. Improved heterogeneous distance functions. In *Journal of Artificial Intelligence Research.* USA : AI Access Foundation. ISSN 1076–9757, 1997, vol. 6, p. 1–34.

[35] HECHENBICHLER, K., SCHLIEP, K. Weighted k–Nearest–Neighbor Techniques and Ordinal Classification. In *Discussion Paper 399, SFB 386* Munich : Ludwigs–Maximilians University. 2004, vol. 399, p. 1–16.

[36] DAVIS, CH. The Norm of the Schur Product Operation. In *Numerische Mathematik.* Gottingen : Digi Zeitschriften e. V. 1962, vol. 4, p. 343—344.

[37] NIELSEN, M. A. *Neural Networks and Deep Learning* [online]. Determination Press, 2015. [cit. 2017–11–18]. Dostupné na internete: http://neuralnetworksanddeeplearning.com

[38] GOODFELLOW, I. et al. *Deep Learning* [online]. MIT Press, 2016. [cit. 2017–11–18]. Dostupné na internete: http://www.deeplearningbook.org

[39] STEINWART, I., CHRISTMANN, A. *Support Vector Machine.* New York : Springer Verlag, 2008. 601 p. ISBN 978–0–387–77241–7.

[40] PERKINS, S., LACKER, K., THEILER, J. Grafting: Fast, Incremental Feature Selection by Gradient Descent in Function Space. In *Journal of Machine Learning Research.* ISSN 1532-4435, 2003, vol. 3 p. 1333–1356.

[41] DROTÁR, P., GAZDA, J., SMÉKAL, Z. An Experimental Comparison of Feature Selection Methods on Two-class Biomedical Datasets. In *Computers in Biology and Medicine.* The Netherlands : Elsevier B. V. 2015, vol. 66 p. 1–10.

Zoznam príloh

Príloha A CD médium – diplomová práca v elektronickej podobe

Príloha B Experimentálne výsledky

Príloha C Používateľská príručka

Obsah priloženého CD

Názov adresára:	Popis obsahu:
\doc	
	Diplomová práca a prílohy vo formáte pdf.
\tex	
	Zdrojové súbory diplomovej práce a príloh.
\src	
\src\WkNN_FS	Zdrojové kódy metódy WkNN–FS: - základný algoritmus WkNN–FS, - rekurzívna eliminácia s využitím WkNN–FS.
\src\mRMR	Zdrojové kódy zovšeobecneného algoritmu mRMR: - zovšeobecnený algoritmus mRMR, - utility pre mRMR: kritériá MID a MIQ, korelačné miery.
\src\utils	Zdrojové kódy podporných nástrojov pre realizáciu výberu premenných a vyhodnocovanie výsledkov
\data	
\data\syntetic_data	Syntetické súbory dát použité pri hodnotení metód.
\data\real_data	Reálne súbory dát použité pri hodnotení metód.

Technická univerzita v Košiciach

Fakulta elektrotechniky a informatiky

Katedra počítačov a informatiky

Návrh metódy výberu príznakov na báze k–NN algoritmu

Diplomová práca

Príloha B

Experimentálne výsledky

Vedúci diplomovej práce:

Doc. Ing. Peter Drotár, PhD.

Diplomant:

Peter Bugata

Košice 2018

Obsah

1 Výsledky pre syntetické dáta

1.1 Výsledky pre súbor dát Madelon100s500f

500 premenných, z toho 5 TF

100 pozorovaní, 50 z triedy 0 / 50 z triedy 1

Tabuľka 1 – 1 Výsledky pre súbor dát Madelon100s500f

Metóda	Suc.	Vybrané TF	Presnosť (Accuracy) NB	SVM	RF	k–NN
Všetky premenné			0.83	0.68	0.91	0.61
Všetky TF			0.98	1.00	0.98	0.99
mRMR/MI	0.60	5,3,2	0.99	0.98	0.97	0.92
mRMR/χ^2	0.60	5,3,2	0.99	0.99	0.98	0.94
mRMR/dCor	0.80	5,2,3,4	0.97	0.99	0.97	0.94
MaxDep/dCor	0.80	5,3,2,4	0.99	0.97	0.98	0.95
Ranking/dCor	0.80	5,3,2,4	0.94	0.98	0.97	0.95
ReliefF	0.80	5,3,2,4	0.98	0.98	0.98	0.93
f–score	0.80	5,3,2,4	0.98	0.94	0.97	0.95
WkNN(SE, L2, rbf)	0.80	5,2,3,4	0.96	0.97	0.99	**1.00**
WkNN(SE, L1, exp)	0.80	5,2,4,3	0.97	0.98	0.99	**1.00**
WkNN(SE, L2, exp)	**0.99**	5,2,4,3,1	0.98	0.99	0.97	**1.00**
WkNN(AE, L2, rbf)	0.80	5,2,4,3	0.93	0.98	0.98	**1.00**
WkNN(AE, L1, exp)	0.80	5,2,3,4	0.97	0.99	0.98	**1.00**
WkNN(AE, L2, exp)	0.80	5,2,4,3	0.98	1.00	0.99	**1.00**
WkNN(HL, L2, rbf)	0.80	5,2,3,4	0.96	0.96	0.99	**1.00**
WkNN(HL, L1, exp)	0.80	5,2,4,3	0.99	0.99	0.98	**1.00**
WkNN(HL, L2, exp)	**0.99**	5,2,4,3,1	0.97	0.97	0.99	**1.00**
WkNN(CE, L2, rbf)	0.80	5,2,4,3	0.96	0.95	0.97	**1.00**
WkNN(CE, L1, exp)	0.80	5,2,3,4	0.98	0.99	0.98	**1.00**
WkNN(CE, L2, exp)	0.80	5,2,4,3	0.98	1.00	0.98	**1.00**

1.2 Výsledky pre súbor dát Madelon200s500f

500 premenných, z toho 5 TF

200 pozorovaní, 100 z triedy 0 / 100 z triedy 1

Tabuľka 1 – 2 Výsledky pre súbor dát Madelon200s500f

			Presnosť (Accuracy)			
Metóda	Suc.	Vybrané TF	NB	SVM	RF	k–NN
Všetky premenné			0.58	0.54	0.70	0.53
Všetky TF			0.71	0.96	0.95	0.99
mRMR/MI	0.40	3,2	0.77	0.74	0.78	0.69
mRMR/χ^2	0.40	3,2	0.76	0.73	0.80	0.71
mRMR/dCor	0.40	3,1	0.78	0.77	0.82	0.77
MaxDep/dCor	0.80	3,1,4,2	0.75	0.79	0.80	0.75
Ranking/dCor	0.60	3,1,2	0.77	0.79	0.82	0.78
ReliefF	0.60	3,1,2	0.72	0.80	0.79	0.78
f–score	0.60	3,1,2	0.76	0.80	0.82	0.80
WkNN(SE, L2, rbf)	0.99	3,5,1,4,2	0.72	0.84	0.88	0.99
WkNN(SE, L1, exp)	0.99	3,5,1,4,2	0.75	0.89	0.91	**1.00**
WkNN(SE, L2, exp)	**1.00**	3,4,5,1,2	0.72	0.94	0.94	0.99
WkNN(AE, L2, rbf)	**1.00**	3,5,4,1,2	0.73	0.86	0.92	0.99
WkNN(AE, L1, exp)	0.99	3,5,4,1,2	0.74	0.84	0.91	0.99
WkNN(AE, L2, exp)	**1.00**	3,4,1,5,2	0.71	0.96	0.95	0.99
WkNN(HL, L2, rbf)	0.99	3,5,4,1,2	0.74	0.84	0.89	0.99
WkNN(HL, L1, exp)	0.99	3,5,1,4,2	0.72	0.86	0.89	**1.00**
WkNN(HL, L2, exp)	**1.00**	3,4,1,5,2	0.69	0.94	0.95	0.99
WkNN(CE, L2, rbf)	0.99	3,5,1,4,2	0.76	0.85	0.93	0.99
WkNN(CE, L1, exp)	0.99	3,1,5,4,2	0.73	0.85	0.91	**1.00**
WkNN(CE, L2, exp)	**1.00**	3,4,5,1,2	0.72	0.94	0.94	0.99

1.3 Výsledky pre súbor dát MadelonHD

15 000 premenných, z toho 15 TF

150 pozorovaní, 75 z triedy 0 / 75 z triedy 1

Tabuľka 1 – 3 Výsledky pre súbor dát MadelonHD

Metóda	Suc. 450*	Vybrané TF	Suc. 30*	Presnosť (Accuracy) NB	SVM	RF	k–NN
Všetky premenné				0.59	0.56	0.62	0.58
Všetky TF				0.87	0.96	0.91	0.92
mRMR/MI	0.40	14,5,4,1,9,3	0.20	0.94	0.92	0.94	0.84
mRMR/χ^2	0.40	14,5,4,1,9,3	0.20	0.95	0.91	0.93	0.87
Ranking/dCor	0.53	5,14,4,1,13,11,9,7	0.27	0.93	0.95	0.92	0.94
ReliefF	0.40	14,5,4,13,1,7	0.33	0.91	0.87	0.90	0.88
f–score	0.47	5,14,4,1,13,11,9	0.27	0.96	0.98	0.94	0.97
WkNN(SE, L2, rbf)	0.53	5,14,4,1,13,11,7,9	0.27	0.97	0.96	0.93	**0.99**
WkNN(SE, L1, exp)	0.53	5,14,4,1,13,7,11,9	0.33	0.96	0.95	0.93	**0.99**
WkNN(SE, L2, exp)	**0.60**	14,5,4,1,7,10,8,2,6	**0.60**	0.95	0.99	0.95	**0.99**
WkNN(AE, L2, rbf)	0.53	5,14,4,1,13,11,7,9	0.27	0.95	0.96	0.93	0.98
WkNN(AE, L1, exp)	0.53	5,14,4,1,13,7,11,9	0.33	0.96	0.95	0.93	0.97
WkNN(AE, L2, exp)	0.40	5,14,4,1,7,10	0.40	0.89	0.93	0.92	0.94
WkNN(HL, L2, rbf)	0.53	5,14,4,1,13,11,7,9	0.27	0.95	0.96	0.93	**0.99**
WkNN(HL, L1, exp)	0.53	5,14,4,1,13,7,11,9	0.33	0.96	0.95	0.93	0.97
WkNN(HL, L2, exp)	0.40	5,14,4,1,7,10	0.40	0.90	0.95	0.91	0.94
WkNN(CE, L2, rbf)	0.53	5,14,4,1,13,11,7,9	0.27	0.97	0.97	0.93	0.97
WkNN(CE, L1, exp)	0.53	5,14,4,1,13,7,11,9	0.33	0.96	0.95	0.92	**0.99**
WkNN(CE, L2, exp)	0.47	14,5,4,1,7,10,8	0.47	0.94	0.99	0.95	0.96

*počet vyberaných premenných použitých pri vyhodnotení indexu úspešnosti

1.4 Výsledky pre súbor dát LinReg200s500f

500 premenných, z toho 5 TF

200 pozorovaní

Tabuľka 1 – 4 Výsledky pre súbor dát LinReg200s500f

			R^2			
Metóda	Suc.	Vybrané TF	LR	SVM	RF	k–NN
Všetky premenné			0.45	0.18	0.71	0.09
Všetky TF			1.00	0.97	0.88	0.80
mRMR/MI	0.60	4,2,5	0.94	0.80	0.82	0.51
mRMR/χ^2	0.80	4,2,5,1	0.99	0.83	0.83	0.49
mRMR/Pearson	0.80	4,5,2,1	0.99	0.88	0.84	0.56
mRMR/Spearman	0.80	4,5,2,1	0.99	0.86	0.83	0.59
mRMR/dCor	0.80	4,5,2,1	0.99	0.86	0.83	0.59
MaxDep/dCor	0.99	4,2,5,1,3	1.00	0.87	0.84	0.55
Ranking/dCor	0.80	4,2,5,3	0.96	0.85	0.83	0.58
ReliefF	0.40	4,5	0.70	0.60	0.68	0.35
f–score	0.60	4,2,5	0.94	0.80	0.82	0.54
Ranking/Pearson	0.80	4,2,5,1	0.99	0.87	0.83	0.57
WkNN(SE, L2, rbf)	**1.00**	4,2,5,1,3	1.00	0.88	0.84	0.94
WkNN(SE, L1, exp)	0.80	4,2,5,1	0.99	0.86	0.84	0.92
WkNN(SE, L2, exp)	**1.00**	4,2,5,1,3	1.00	0.92	0.84	0.94
WkNN(AE, L2, rbf)	**1.00**	4,2,5,1,3	1.00	0.87	0.85	0.93
WkNN(AE, L1, exp)	**1.00**	4,2,5,1,3	1.00	0.91	0.85	0.92
WkNN(AE, L2, exp)	0.99	4,2,5,3,1	1.00	0.90	0.84	0.93
WkNN(HL, L2, rbf)	**1.00**	4,2,5,1,3	1.00	0.87	0.85	0.93
WkNN(HL, L1, exp)	**1.00**	4,2,5,1,3	1.00	0.91	0.85	0.92
WkNN(HL, L2, exp)	0.99	4,2,5,3,1	1.00	0.90	0.84	0.93

1.5 Výsledky pre súbor dát Friedman200s500f

500 premenných, z toho 5 TF

200 pozorovaní

Tabuľka 1 – 5 Výsledky pre súbor dát Friedman200s500f

			R^2			
Metóda	Suc.	Vybrané TF	LR	SVM	RF	k–NN
Všetky premenné			-0.31	0.15	0.59	0.06
Všetky TF			0.78	0.91	0.79	0.77
mRMR/MI	0.80	4,1,2,5	0.78	0.71	0.72	0.55
mRMR/χ^2	0.80	1,4,2,5	0.77	0.71	0.72	0.50
mRMR/Pearson	0.80	4,2,5,1	0.79	0.69	0.74	0.59
mRMR/Spearman	0.80	4,2,5,1	0.79	0.73	0.74	0.60
mRMR/dCor	0.99	4,5,2,1,3	0.78	0.78	0.74	0.57
MaxDep/dCor	0.80	4,2,5,1	0.78	0.73	0.73	0.64
Ranking/dCor	0.80	4,2,5,1	0.78	0.67	0.73	0.60
ReliefF	0.99	4,5,2,1,3	0.77	0.73	0.73	0.49
f–score	0.80	4,1,2,5	0.76	0.66	0.71	0.55
Ranking/Pearson	0.80	4,5,2,1	0.77	0.71	0.73	0.62
WkNN(SE, L2, rbf)	0.80	2,1,4,5	0.78	0.73	0.73	0.80
WkNN(SE, L1, exp)	**1.00**	2,3,4,1,5	0.76	0.80	0.75	**0.83**
WkNN(SE, L2, exp)	0.80	2,4,1,5	0.78	0.77	0.73	**0.83**
WkNN(AE, L2, rbf)	0.80	2,5,4,1	0.80	0.77	0.73	0.81
WkNN(AE, L1, exp)	**1.00**	3,2,4,1,5	0.78	0.78	0.74	**0.83**
WkNN(AE, L2, exp)	0.80	2,4,5,1	0.77	0.74	0.73	0.82
WkNN(HL, L2, rbf)	0.80	2,5,4,1	0.79	0.78	0.73	0.81
WkNN(HL, L1, exp)	**1.00**	3,4,2,1,5	0.78	0.77	0.74	0.82
WkNN(HL, L2, exp)	0.80	2,4,5,1	0.77	0.74	0.73	0.82

2 Výsledky pre reálne dáta

2.1 Výsledky pre súbor dát Alon

2 000 premenných

62 pozorovaní, 40 z triedy 0 / 22 z triedy 1

Tabuľka 2 – 1 Výsledky pre súbor dát Alon

	NB		SVM		RF		k–NN	
Metóda	Acc.	F_1	Acc.	F_1	Acc.	F_1	Acc.	F_1
Všetky premenné	0.61	0.59	0.79	0.53	0.82	0.69	0.68	0.10
mRMR/MI	0.89	0.83	0.89	0.83	0.87	0.81	0.84	0.73
MaxDep/MI	0.79	0.72	0.82	0.69	0.85	0.79	0.80	0.58
mRMR/dCor	0.90	0.86	0.89	0.84	0.87	0.81	0.87	0.80
MaxDep/dCor	0.92	0.89	0.90	0.86	0.87	0.81	0.89	0.76
Ranking/dCor	0.87	0.81	0.87	0.80	0.85	0.78	0.85	0.76
ReliefF	0.85	0.78	0.87	0.80	0.87	0.81	0.85	0.72
f–score	0.87	0.81	0.87	0.80	0.87	0.81	0.85	0.76
WkNN(SE, L2, rbf)	0.86	0.82	0.90	0.84	0.87	0.79	0.98	0.97
WkNN(SE, L1, exp)	0.84	0.78	0.89	0.81	0.89	0.84	0.98	0.97
WkNN(SE, L2, exp)	0.87	0.85	0.92	0.88	0.85	0.77	**1.00**	**1.00**
WkNN(AE, L2, rbf)	0.78	0.74	0.89	0.81	0.87	0.80	0.98	0.97
WkNN(AE, L1, exp)	0.81	0.75	0.90	0.84	0.87	0.80	0.97	0.95
WkNN(AE, L2, exp)	0.86	0.84	0.94	0.91	0.89	0.82	0.98	0.97
WkNN(HL, L2, rbf)	0.83	0.77	0.92	0.85	0.87	0.80	0.98	0.97
WkNN(HL, L1, exp)	0.84	0.81	0.89	0.82	0.89	0.84	0.98	0.97
WkNN(HL, L2, exp)	0.87	0.85	0.92	0.88	0.87	0.76	0.98	0.97
WkNN(CE, L2, rbf)	0.91	0.88	0.89	0.81	0.84	0.76	0.98	0.97
WkNN(CE, L1, exp)	0.78	0.72	0.94	0.90	0.87	0.80	0.98	0.97
WkNN(CE, L2, exp)	0.89	0.83	0.90	0.84	0.85	0.78	0.98	0.97

Vysvetlivky: Acc. – presnosť (*accuracy*), F_1 – F_1 *score*

2.2 Výsledky pre súbor dát Burczynski

22 283 premenných

127 pozorovaní, 85 z triedy 0 / 42 z triedy 1

Tabuľka 2 – 2 Výsledky pre súbor dát Burczynski

	NB		SVM		RF		k–NN	
Metóda	Acc.	F_1	Acc.	F_1	Acc.	F_1	Acc.	F_1
Všetky premenné	0.86	0.79	0.93	0.84	0.93	0.89	0.81	0.69
mRMR/MI	0.96	0.94	0.98	0.97	0.98	0.97	0.97	0.94
MaxDep/MI	0.92	0.88	0.97	0.96	0.94	0.92	0.90	0.82
mRMR/dCor	0.98	0.97	0.99	0.99	0.99	0.99	0.99	0.99
MaxDep/dCor	0.99	0.99	0.99	0.99	0.98	0.97	0.99	0.99
Ranking/dCor	0.96	0.94	0.98	0.97	0.96	0.94	0.97	0.95
ReliefF	0.83	0.76	0.96	0.93	0.93	0.86	0.89	0.78
f–score	0.94	0.92	0.98	0.97	0.97	0.95	0.98	0.96
WkNN(SE, L2, rbf)	0.98	0.97	1.00	1.00	0.99	0.99	**1.00**	**1.00**
WkNN(SE, L1, exp)	1.00	1.00	1.00	1.00	0.98	0.97	**1.00**	**1.00**
WkNN(SE, L2, exp)	0.99	0.99	1.00	1.00	0.99	0.99	**1.00**	**1.00**
WkNN(AE, L2, rbf)	0.98	0.97	1.00	1.00	0.98	0.97	**1.00**	**1.00**
WkNN(AE, L1, exp)	0.99	0.99	1.00	1.00	0.99	0.99	**1.00**	**1.00**
WkNN(AE, L2, exp)	1.00	1.00	1.00	1.00	0.98	0.97	**1.00**	**1.00**
WkNN(HL, L2, rbf)	0.99	0.99	1.00	1.00	0.99	0.99	**1.00**	**1.00**
WkNN(HL, L1, exp)	1.00	1.00	1.00	1.00	0.99	0.99	**1.00**	**1.00**
WkNN(HL, L2, exp)	1.00	1.00	1.00	1.00	0.99	0.99	**1.00**	**1.00**
WkNN(CE, L2, rbf)	0.98	0.97	1.00	1.00	0.99	0.99	**1.00**	**1.00**
WkNN(CE, L1, exp)	0.98	0.98	1.00	1.00	0.98	0.97	**1.00**	**1.00**
WkNN(CE, L2, exp)	0.98	0.98	1.00	1.00	0.99	0.99	**1.00**	**1.00**

2.3 Výsledky pre súbor dát Chin

22 215 premenných

118 pozorovaní, 43 z triedy 0 / 75 z triedy 1

Tabuľka 2 – 3 Výsledky pre súbor dát Chin

	NB		SVM		RF		k–NN	
Metóda	Acc.	F_1	Acc.	F_1	Acc.	F_1	Acc.	F_1
Všetky premenné	0.86	0.89	0.85	0.89	0.88	0.92	0.80	0.87
mRMR/MI	0.89	0.92	0.90	0.93	0.91	0.93	0.88	0.92
MaxDep/MI	0.91	0.93	0.89	0.92	0.92	0.94	0.91	0.93
mRMR/dCor	0.93	0.95	0.92	0.94	0.92	0.94	0.91	0.93
MaxDep/dCor	0.93	0.95	0.97	0.97	0.92	0.94	0.90	0.93
Ranking/dCor	0.88	0.91	0.90	0.93	0.90	0.93	0.90	0.93
ReliefF	0.88	0.91	0.90	0.93	0.89	0.92	0.89	0.92
f–score	0.89	0.92	0.91	0.93	0.90	0.93	0.90	0.93
WkNN(SE, L2, rbf)	0.92	0.94	0.97	0.97	0.91	0.93	0.99	0.99
WkNN(SE, L1, exp)	0.95	0.96	0.96	0.97	0.91	0.93	**1.00**	**1.00**
WkNN(SE, L2, exp)	0.98	0.98	0.98	0.99	0.93	0.95	0.98	0.99
WkNN(AE, L2, rbf)	0.91	0.93	0.93	0.95	0.90	0.93	0.99	0.99
WkNN(AE, L1, exp)	0.91	0.93	0.95	0.96	0.91	0.94	0.98	0.99
WkNN(AE, L2, exp)	0.96	0.97	0.97	0.98	0.92	0.94	0.97	0.98
WkNN(HL, L2, rbf)	0.90	0.92	0.94	0.96	0.89	0.92	**1.00**	**1.00**
WkNN(HL, L1, exp)	0.93	0.95	0.95	0.96	0.91	0.93	0.99	0.99
WkNN(HL, L2, exp)	0.98	0.98	0.98	0.98	0.93	0.95	0.98	0.98
WkNN(CE, L2, rbf)	0.90	0.92	0.91	0.93	0.92	0.94	0.99	0.99
WkNN(CE, L1, exp)	0.94	0.96	0.97	0.97	0.91	0.94	**1.00**	**1.00**
WkNN(CE, L2, exp)	0.97	0.97	0.98	0.99	0.92	0.94	**1.00**	**1.00**

2.4 Výsledky pre súbor dát Chowdaryi

22 283 premenných

104 pozorovaní, 62 z triedy 0 / 42 z triedy 1

Tabuľka 2 – 4 Výsledky pre súbor dát Chowdaryi

	NB		SVM		RF		k–NN	
Metóda	Acc.	F_1	Acc.	F_1	Acc.	F_1	Acc.	F_1
Všetky premenné	0.91	0.88	0.96	0.95	0.96	0.95	0.90	0.85
mRMR/MI	0.97	0.97	0.98	0.97	0.98	0.97	0.94	0.91
MaxDep/MI	0.88	0.84	0.92	0.91	0.91	0.88	0.85	0.80
mRMR/dCor	0.99	0.99	0.99	0.99	0.98	0.97	0.99	0.99
MaxDep/dCor	0.97	0.97	0.99	0.99	0.98	0.97	0.99	0.99
Ranking/dCor	0.94	0.92	0.98	0.97	0.96	0.95	0.98	0.97
ReliefF	0.98	0.97	0.98	0.97	0.98	0.97	0.96	0.95
f–score	0.98	0.97	0.98	0.97	0.98	0.97	0.99	0.99
WkNN(SE, L2, rbf)	0.97	0.97	1.00	1.00	0.97	0.96	**1.00**	**1.00**
WkNN(SE, L1, exp)	0.98	0.97	1.00	1.00	0.98	0.97	**1.00**	**1.00**
WkNN(SE, L2, exp)	0.97	0.97	0.99	0.99	0.98	0.97	**1.00**	**1.00**
WkNN(AE, L2, rbf)	0.94	0.93	0.98	0.97	0.98	0.97	0.99	0.99
WkNN(AE, L1, exp)	0.98	0.97	0.99	0.99	0.98	0.97	**1.00**	**1.00**
WkNN(AE, L2, exp)	0.98	0.97	0.99	0.99	0.98	0.97	**1.00**	**1.00**
WkNN(HL, L2, rbf)	0.97	0.97	1.00	1.00	0.97	0.96	**1.00**	**1.00**
WkNN(HL, L1, exp)	0.98	0.97	0.99	0.99	0.98	0.97	**1.00**	**1.00**
WkNN(HL, L2, exp)	0.98	0.97	0.98	0.98	0.98	0.97	**1.00**	**1.00**
WkNN(CE, L2, rbf)	0.98	0.97	0.98	0.98	0.98	0.97	**1.00**	**1.00**
WkNN(CE, L1, exp)	1.00	1.00	0.99	0.99	0.98	0.97	**1.00**	**1.00**
WkNN(CE, L2, exp)	0.96	0.95	0.98	0.98	0.98	0.97	**1.00**	**1.00**

2.5 Výsledky pre súbor dát Golub

7 129 premenných

72 pozorovaní, 47 z triedy 0 / 25 z triedy 1

Tabuľka 2 – 5 Výsledky pre súbor dát Golub

	NB		SVM		RF		k–NN	
Metóda	Acc.	F_1	Acc.	F_1	Acc.	F_1	Acc.	F_1
Všetky premenné	0.99	0.99	0.84	0.90	0.99	0.99	0.73	0.83
mRMR/MI	0.95	0.94	0.98	0.97	0.98	0.96	0.95	0.90
MaxDep/MI	0.85	0.76	0.92	0.87	0.89	0.83	0.89	0.79
mRMR/dCor	0.96	0.96	0.98	0.97	0.99	0.98	0.99	0.98
MaxDep/dCor	0.97	0.97	1.00	1.00	0.98	0.96	0.75	0.38
Ranking/dCor	0.95	0.94	0.98	0.97	0.99	0.98	0.95	0.90
ReliefF	0.95	0.94	0.98	0.97	0.99	0.98	0.94	0.88
f–score	0.95	0.94	0.98	0.97	0.96	0.94	0.95	0.90
WkNN(SE, L2, rbf)	1.00	1.00	1.00	1.00	0.97	0.96	**1.00**	**1.00**
WkNN(SE, L1, exp)	1.00	1.00	0.99	0.98	0.99	0.98	**1.00**	**1.00**
WkNN(SE, L2, exp)	1.00	1.00	1.00	1.00	1.00	1.00	**1.00**	**1.00**
WkNN(AE, L2, rbf)	1.00	1.00	0.99	0.98	0.99	0.98	**1.00**	**1.00**
WkNN(AE, L1, exp)	0.99	0.99	0.99	0.98	0.99	0.98	**1.00**	**1.00**
WkNN(AE, L2, exp)	1.00	1.00	1.00	1.00	0.99	0.98	**1.00**	**1.00**
WkNN(HL, L2, rbf)	1.00	1.00	1.00	1.00	0.99	0.98	**1.00**	**1.00**
WkNN(HL, L1, exp)	1.00	1.00	0.99	0.98	0.99	0.98	**1.00**	**1.00**
WkNN(HL, L2, exp)	1.00	1.00	1.00	1.00	0.99	0.98	**1.00**	**1.00**
WkNN(CE, L2, rbf)	0.99	0.99	0.99	0.98	0.99	0.98	**1.00**	**1.00**
WkNN(CE, L1, exp)	1.00	1.00	1.00	1.00	0.99	0.98	**1.00**	**1.00**
WkNN(CE, L2, exp)	1.00	1.00	1.00	1.00	0.97	0.98	**1.00**	**1.00**

2.6 Výsledky pre súbor dát Gordon

12 533 premenných

181 pozorovaní, 31 z triedy 0 / 150 z triedy 1

Tabuľka 2 – 6 Výsledky pre súbor dát Gordon

	NB		SVM		RF		k–NN	
Metóda	Acc.	F_1	Acc.	F_1	Acc.	F_1	Acc.	F_1
Všetky premenné	0.98	0.99	0.98	0.99	0.99	1.00	0.84	0.91
mRMR/MI	1.00	1.00	0.99	0.99	0.99	1.00	0.99	0.99
MaxDep/MI	0.99	0.99	0.99	1.00	0.99	0.99	0.99	1.00
mRMR/dCor	0.99	1.00	0.99	1.00	0.99	1.00	0.98	0.99
MaxDep/dCor	0.99	0.99	1.00	1.00	0.99	0.99	0.98	0.99
Ranking/dCor	0.98	0.99	0.99	0.99	0.99	1.00	0.98	0.99
ReliefF	0.99	0.99	0.99	1.00	0.98	0.99	0.94	0.97
f–score	0.99	0.99	0.99	1.00	0.99	1.00	0.97	0.98
WkNN(SE, L2, rbf)	0.98	0.99	0.98	0.99	0.99	1.00	**1.00**	**1.00**
WkNN(SE, L1, exp)	1.00	1.00	0.99	1.00	1.00	1.00	**1.00**	**1.00**
WkNN(SE, L2, exp)	1.00	1.00	0.99	1.00	0.99	1.00	**1.00**	**1.00**
WkNN(AE, L2, rbf)	0.99	0.99	0.99	0.99	1.00	1.00	**1.00**	**1.00**
WkNN(AE, L1, exp)	1.00	1.00	0.99	1.00	0.99	1.00	**1.00**	**1.00**
WkNN(AE, L2, exp)	1.00	1.00	0.99	1.00	0.99	1.00	**1.00**	**1.00**
WkNN(HL, L2, rbf)	0.99	0.99	0.98	0.99	0.99	1.00	**1.00**	**1.00**
WkNN(HL, L1, exp)	1.00	1.00	0.99	1.00	1.00	1.00	**1.00**	**1.00**
WkNN(HL, L2, exp)	1.00	1.00	0.99	1.00	0.99	1.00	**1.00**	**1.00**
WkNN(CE, L2, rbf)	0.98	0.99	0.99	1.00	0.99	1.00	**1.00**	**1.00**
WkNN(CE, L1, exp)	1.00	1.00	0.99	1.00	0.99	1.00	**1.00**	**1.00**
WkNN(CE, L2, exp)	1.00	1.00	0.99	1.00	1.00	1.00	**1.00**	**1.00**

2.7 Výsledky pre súbor dát Pomeroy

7 128 premenných

60 pozorovaní, 39 z triedy 0 / 21 z triedy 1

Tabuľka 2 – 7 Výsledky pre súbor dát Pomeroy

	NB		SVM		RF		k–NN	
Metóda	Acc.	F_1	Acc.	F_1	Acc.	F_1	Acc.	F_1
Všetky premenné	0.70	0.46	0.65	0.00	0.65	0.05	0.65	0.05
mRMR/MI	0.92	0.86	0.82	0.66	0.82	0.65	0.77	0.65
MaxDep/MI	0.75	0.54	0.80	0.66	0.78	0.63	0.72	0.37
mRMR/dCor	0.88	0.80	0.92	0.81	0.87	0.78	0.92	0.84
MaxDep/dCor	0.93	0.90	0.95	0.92	0.85	0.71	0.80	0.81
Ranking/dCor	0.85	0.78	0.94	0.85	0.87	0.76	0.89	0.79
ReliefF	0.80	0.64	0.83	0.66	0.83	0.66	0.72	0.27
f–score	0.89	0.81	0.90	0.80	0.83	0.72	0.88	0.79
WkNN(SE, L2, rbf)	0.88	0.81	0.95	0.91	0.80	0.57	**1.00**	**1.00**
WkNN(SE, L1, exp)	0.77	0.66	0.89	0.80	0.80	0.55	0.98	0.97
WkNN(SE, L2, exp)	0.90	0.79	0.95	0.92	0.80	0.53	**1.00**	**1.00**
WkNN(AE, L2, rbf)	0.89	0.81	0.94	0.88	0.75	0.43	0.98	0.97
WkNN(AE, L1, exp)	0.72	0.64	0.82	0.62	0.80	0.57	0.98	0.97
WkNN(AE, L2, exp)	0.87	0.79	0.95	0.91	0.83	0.63	**1.00**	**1.00**
WkNN(HL, L2, rbf)	0.91	0.85	0.95	0.87	0.82	0.65	**1.00**	**1.00**
WkNN(HL, L1, exp)	0.82	0.68	0.90	0.78	0.85	0.65	**1.00**	**1.00**
WkNN(HL, L2, exp)	0.90	0.84	0.93	0.88	0.87	0.73	**1.00**	**1.00**
WkNN(CE, L2, rbf)	0.74	0.64	0.91	0.80	0.78	0.57	**1.00**	**1.00**
WkNN(CE, L1, exp)	0.89	0.83	0.94	0.85	0.85	0.71	**1.00**	**1.00**
WkNN(CE, L2, exp)	0.88	0.76	0.94	0.85	0.80	0.62	**1.00**	**1.00**

2.8 Výsledky pre súbor dát Singh

12 600 premenných

102 pozorovaní, 50 z triedy 0 / 52 z triedy 1

Tabuľka 2 – 8 Výsledky pre súbor dát Singh

	NB		SVM		RF		k–NN	
Metóda	Acc.	F_1	Acc.	F_1	Acc.	F_1	Acc.	F_1
Všetky premenné	0.64	0.71	0.89	0.88	0.92	0.92	0.79	0.82
mRMR/MI	0.95	0.95	0.95	0.95	0.96	0.96	0.95	0.95
MaxDep/MI	0.96	0.96	0.97	0.97	0.95	0.95	0.93	0.94
mRMR/dCor	0.94	0.94	0.96	0.96	0.95	0.95	0.98	0.98
MaxDep/dCor	0.96	0.96	0.99	0.99	0.96	0.96	0.96	0.97
Ranking/dCor	0.93	0.93	0.93	0.93	0.92	0.92	0.94	0.94
ReliefF	0.94	0.94	0.94	0.94	0.95	0.95	0.95	0.95
f–score	0.94	0.94	0.93	0.93	0.93	0.93	0.94	0.94
WkNN(SE, L2, rbf)	0.95	0.94	0.98	0.98	0.94	0.94	**1.00**	**1.00**
WkNN(SE, L1, exp)	0.93	0.93	0.98	0.98	0.97	0.97	0.99	0.99
WkNN(SE, L2, exp)	0.95	0.94	0.96	0.96	0.95	0.95	**1.00**	**1.00**
WkNN(AE, L2, rbf)	0.95	0.95	0.98	0.98	0.95	0.95	0.98	0.98
WkNN(AE, L1, exp)	0.93	0.93	0.98	0.98	0.96	0.96	0.98	0.98
WkNN(AE, L2, exp)	0.96	0.95	0.98	0.98	0.96	0.96	0.99	0.99
WkNN(HL, L2, rbf)	0.97	0.97	0.98	0.98	0.95	0.95	**1.00**	**1.00**
WkNN(HL, L1, exp)	0.93	0.92	0.96	0.95	0.93	0.93	0.99	0.99
WkNN(HL, L2, exp)	0.96	0.95	0.95	0.94	0.95	0.95	**1.00**	**1.00**
WkNN(CE, L2, rbf)	0.95	0.95	0.98	0.98	0.96	0.96	**1.00**	**1.00**
WkNN(CE, L1, exp)	0.95	0.94	0.96	0.95	0.94	0.94	**1.00**	**1.00**
WkNN(CE, L2, exp)	0.96	0.95	0.96	0.96	0.95	0.95	**1.00**	**1.00**

2.9 Výsledky pre súbor dát Tian

12 625 premenných

173 pozorovaní, 36 z triedy 0 / 137 z triedy 1

Tabuľka 2 – 9 Výsledky pre súbor dát Tian

	NB		SVM		RF		k–NN	
Metóda	Acc.	F_1	Acc.	F_1	Acc.	F_1	Acc.	F_1
Všetky premenné	0.76	0.85	0.79	0.88	0.79	0.88	0.79	0.88
mRMR/MI	0.87	0.91	0.86	0.91	0.87	0.92	0.82	0.90
MaxDep/MI	0.78	0.87	0.80	0.88	0.86	0.92	0.80	0.89
mRMR/dCor	0.86	0.91	0.90	0.94	0.86	0.92	0.82	0.90
MaxDep/dCor	0.90	0.94	0.96	0.97	0.82	0.90	0.79	0.88
Ranking/dCor	0.83	0.88	0.84	0.90	0.85	0.91	0.82	0.90
ReliefF	0.85	0.90	0.85	0.91	0.81	0.89	0.80	0.89
f–score	0.84	0.89	0.85	0.91	0.84	0.91	0.81	0.89
WkNN(SE, L2, rbf)	0.83	0.89	0.83	0.90	0.81	0.89	0.95	0.97
WkNN(SE, L1, exp)	0.81	0.87	0.87	0.92	0.82	0.90	0.98	0.99
WkNN(SE, L2, exp)	0.89	0.93	0.91	0.95	0.86	0.92	0.94	0.97
WkNN(AE, L2, rbf)	0.76	0.84	0.81	0.89	0.82	0.90	0.97	0.98
WkNN(AE, L1, exp)	0.76	0.84	0.82	0.89	0.83	0.90	0.95	0.97
WkNN(AE, L2, exp)	0.87	0.92	0.91	0.95	0.83	0.90	0.91	0.95
WkNN(HL, L2, rbf)	0.83	0.89	0.84	0.91	0.82	0.90	0.98	0.99
WkNN(HL, L1, exp)	0.88	0.93	0.88	0.93	0.86	0.92	**1.00**	**1.00**
WkNN(HL, L2, exp)	0.84	0.89	0.91	0.95	0.84	0.91	0.93	0.96
WkNN(CE, L2, rbf)	0.85	0.90	0.91	0.95	0.81	0.89	0.99	**1.00**
WkNN(CE, L1, exp)	0.84	0.90	0.89	0.94	0.84	0.91	0.99	**1.00**
WkNN(CE, L2, exp)	0.89	0.93	0.93	0.96	0.84	0.91	0.96	0.97

Zoznam tabuliek

Technická univerzita v Košiciach

Fakulta elektrotechniky a informatiky

Katedra počítačov a informatiky

Návrh metódy výberu príznakov na báze k–NN algoritmu

Diplomová práca

Príloha C

Používateľská príručka

Vedúci diplomovej práce:

Doc. Ing. Peter Drotár, PhD.

Diplomant:

Peter Bugata

Košice 2018

Obsah

1 Implementácia metód na výber premenných

Príručka obsahuje popis implementácie navrhnutej metódy na výber premenných založenej na algoritme k–NN (WkNN–FS) a popis implementácie zovšeobecnenej metódy mRMR. Zdrojové kódy programov sú obsahom Prílohy A.

2 Použité programové prostriedky

Všetky navrhnuté algoritmy boli implementované v programovacom jazyku Python verzia 3.6. Pritom boli použité algoritmy a nástroje strojového učenia z knižnice `scikit-learn` [1], základný balík pre operácie s maticami a vedecké výpočty `Numpy` [2], balík `Pandas` s funkciami na spracovanie dát, knižnica `SciPy` obsahujúca matematické a štatistické nástroje [3] a tiež balík `Matplotlib` na vytváranie grafických výstupov [4].

Programovací jazyk Python je multiparadigmový jazyk, podporuje objektovo orientované, štruktúrované aj funkcionálne programovanie. Je to dynamicky typovaný jazyk so stredne prísnou typovou kontrolou. Podporuje veľké množstvo vysokoúrovňových dátových typov a má automatickú správu pamäte. Je to interpretovaný jazyk, ktorý zahŕňa podporu pre viacvláknové a viacprocesové paralelné programovanie. Pretože je interpretovaný, je pomalší ako kompilované vyššie programovacie jazyky. Umožňuje však použitie rozširujúcich modulov napísaných v iných jazykoch, napr. C, C++ a pod.

Na vývoj bola použitá voľne dostupná distribúcia jazyka Python Anaconda 3 s integrovaným vývojovým prostredím SPYDER (*Scientific PYthon Development EnviRonment*) 3.2.6. Anaconda 3 integruje `Numpy`, `SciPy`, `Matplotlib`, `scikit-learn` a ďalšie voľne dostupné programové prostriedky na vedecké výpočty. SPYDER umožňuje interaktívnu prácu. Má rozhranie pre ladenie, prezeranie atribútov objektov, prieskumníkov dokumentácie, premenných, súborov a adresárov. Ponúka konzoly rôznych typov.

3 Implementácia metódy WkNN–FS

V práci bol implementovaný nový algoritmus na selekciu premenných založený na metóde k–NN. Algoritmus používa metódu najväčšieho spádu na hľadanie optimálnych váh premenných, v ktorých účelová funkcia nadobúda minimálnu hodnotu (kapitola 3.2.5 práce). Na zníženie počtu nenulových váh premenných umožňuje využiť regularizáciu (kapitola 3.2.11 práce) alebo rekurzívnu elimináciu premenných (kapitola 3.2.9). Kvôli časovej náročnosti bola implementácia optimalizovaná vektorizáciou výpočtu gradientu a predikovaných hodnôt cieľovej premennej a je použiteľná na súboroch dát s veľkým počtom pozorovaní aj veľkým počtom premenných.

3.1 Implementácia základného algoritmu

Implementácia základného algoritmu metódy WkNN–FS je realizovaná triedou:

class `NeighborsVarSelector`(*lambda_const, eta, batch_size, n_iters, metric, p, dist_weights='exp', error_type='mse', delta=0.1, eps=0.0001, prec=1e-10, c=1.0, check_neg_var_weights='zero', normalize_grad=False, stratified_SGD=False*)

Parametre konštruktora:

`lambda_const` : reálne číslo

Regularizačný parameter pre L0 alebo L1 regularizáciu.

`eta` : reálne číslo

Rýchlosť konvergencie (*learning rate*).

`batch_size` : celé číslo

Veľkosť vzorky pre SGD (ak je kladné), inak sa berie celý súbor dát.

`n_iters` : celé číslo

Maximálny počet iterácií algoritmu (*stop condition*).

`metric` : reťazec

Definícia vzdialenosti, možné hodnoty:

- 'minkowski' : Minkowského vzdialenosť Lp,

- 'cityblock' : manhattanská vzdialenosť (L1),
- 'euclidean' : euklidovská vzdialenosť (L2),
- 'sqeuclidean' : štvorcová euklidovská vzdialenosť (druhá mocnina L2),
- ďalšie hodnoty (modifikované): 'minkowski_mod', 'euclidean_mod' a 'sqeuclidean_mod' sa od pôvodných odlišujú tým, že pri výpočte vzdialenosti sa uplatní odmocnina váh (p–ta odmocnina pre Lp). Odporúča sa použitie modifikovaných vzdialeností (kapitola 3.2.4).

`p` : celé číslo

Hodnota p pre vzdialenosť 'minkowski', pre iné vzdialenosti sa nepoužíva.

`dist_weights` : reťazec, implicitne 'exp'

Hodnotiaca funkcia vzdialenosti d (*kernel*), možné hodnoty:

- 'exp' : funkcia e^{-cd},
- 'expp' : funkcia e^{-d^c},
- 'inv' : funkcia $\frac{1}{1+cd}$,
- 'invp' : funkcia $\frac{1}{1+d^c}$.

`error_type` : reťazec, implicitne 'mse'

Funkcia chyby (*loss function*), možné hodnoty:

- 'mse' : štvorcová chyba,
- 'mae' : absolútna chyba,
- 'huber' : Huberova funkcia chyby,
- 'cross_entropy' : binárna krížová entropia.

`delta` : reálne číslo, implicitne 0.1

Hodnota parametra δ pre Huberovu funkciu chyby, inak sa nepoužíva.

`eps` : reálne číslo, implicitne 0.0001

Minimálna norma gradientu (stop condition). Norma gradientu sa počíta ako súčet absolútnych hodnôt jeho zložiek (L1 norma).

`prec` : reálne číslo, implicitne 1e-15

Presnosť zaokrúhľovania.

c : reálne číslo, implicitne 1.0

Parameter hodnotiacej funkcie vzdialenosti.

check_neg_var_weights : reťazec, implicitne 'zero'

Spôsob ošetrenia záporných váh premenných, možné hodnoty:

- 'zero' : vynulovanie všetkých záporných váh,
- 'abs' : nahradenie záporných váh ich absolútnou hodnotou,
- 'new_eta' : zmenšenie hodnoty parametra *eta* tak, aby nedošlo k vzniku záporných váh.

normalize_grad : boolovská hodnota, implicitne False

Príznak normalizácie gradientu.

stratified_SGD : boolovská hodnota, implicitne False

Príznak použitia stratifikovaného výberu vzoriek pri SGD.

Metódy triedy `NeighborsVarSelector` zodpovedajú hlavným krokom algoritmu výberu premenných popísaným v kapitole 3.2.5 práce a sú zhrnuté v tabuľke 3 – 1.

Tabuľka 3 – 1 Metódy triedy `NeighborsVarSelector`

`init_var_weights`(*var_weights*)	nastavenie počiatočných hodnôt váh vysvetľujúcich premenných
`choose_subsample`(*X, y*)	výber dátovej vzorky veľkosti *batch_size*
`compute_distances`(*X_subsample, X, indices*)	výpočet matice vzdialeností a matice ohodnotených vzdialeností
`predict`(*y, indices*)	výpočet vektora predikovaných hodnôt cieľovej premennej
`error_function`(*y*)	výpočet chyby
`dcost_dvl`(*indices, X, y, l*)	výpočet gradientu účelovej funkcie
`update_var_weights`(*gradient_vector*)	aktualizácia váh premenných
`correct_var_weights()`	úprava záporných váh premenných
`select_vars`(*X, y, xcols, initial_weights*)	samotný výber premenných

V ďalšej časti sa nachádza podrobnejší popis jednotlivých metód triedy.

init_var_weights(*var_weights*)

Metóda sa používa na inicializáciu váh vysvetľujúcich premenných. Ak parameter *var_weights* nie je zadaný, počiatočné váhy premenných sa nastavia na hodnoty $\frac{1}{m}$, kde m je počet vysvetľujúcich premenných v súbore dát. Inak sa použije zadaný špecifický vektor, napríklad nulový vektor.

Parametre:

var_weights : numpy vektor alebo zoznam
Počiatočný vektor váh premenných (štartovací bod algoritmu).

choose_subsample(X, y)

Metóda realizuje náhodný výber vzorky, t. j. podmnožiny pozorovaní zo súboru dát. Ak je parameter *batch_size* ≤ 0, metóda vráti pôvodný súbor dát X a vektor hodnôt cieľovej premennej y bezo zmeny. Ak je parameter *batch_size* kladné číslo, metóda náhodne vyberie tento počet pozorovaní. Implementované sú dve stratégie výberu vzoriek popísané v kapitole 3.2.10 práce. Dajú sa prepínať parametrom *stratified_SGD*. Ak je nastavený na hodnotu *False*, použije sa štandardný spôsob. V opačnom prípade sa budú vzorky vyberať špeciálnym – stratifikovaným spôsobom.

Parametre:

X : numpy matica
Súbor dát – matica, ktorej riadky zodpovedajú pozorovaniam.

y : numpy vektor
Vektor hodnôt cieľovej premennej.

compute_distances(*X_subsample, X, indices*)

Metóda sa používa na výpočet matice vzdialeností a matice ohodnotených vzdialeností pozorovaní súboru dát *X_subsample* a súboru dát X. Riadky vstupných matíc sa najskôr prenásobia pôvodným alebo modifikovaným vektorom váh premenných Hadamardovým súčinom. V prípade, ak je parameter konštruktora *metric* nastavený na niektorú zo štandardných metrík, použijú sa pôvodné váhy premenných. Pri ich modifikovaných verziách (s príponou *_mod*) sa násobí príslušnou odmocninou váh premenných. Z takto upravených vstupných matíc sa vypočíta matica vzdialeností

uplatnením zadanej metriky. Aplikovaním vybranej hodnotiacej funkcie vzdialenosti určenej pomocou parametra konštruktora *dist_weights* na maticu vzdialeností bude potom vypočítaná matica ohodnotených vzdialeností.

Parametre:	**`X_subsample`** : numpy matica Matica, ktorej riadky sú v prípade použitia SGD náhodne vybrané pozorovania súboru dát, inak obsahuje všetky pozorovania. **`X`** : numpy matica Súbor dát – matica, ktorej riadky sú všetky pozorovania. **`indices`** : numpy vektor alebo zoznam Indexy vybraných pozorovaní v prípade SGD (inak všetky indexy).

`predict`(*y, indices*)

Metóda vypočíta vektor predikovaných hodnôt cieľovej premennej a uloží ich do atribútu *predictions*. Predikované hodnoty sa počítajú z matice ohodnotených vzdialeností podľa vzťahu 3.14, ktorý je uvedený v kapitole 3.2.2 práce.

Parametre:	**`y`** : numpy vektor Vektor hodnôt cieľovej premennej. **`indices`** : numpy vektor alebo zoznam Indexy vybraných pozorovaní v prípade SGD (inak všetky indexy).

`error_function`(y)

Metóda vypočíta hodnotu priemernej chyby použitú pre informatívne výpisy. Pomocou funkcie chyby nastavenej v parametri konštruktora *error_type* vypočíta vektor chýb predikcie pre jednotlivé pozorovania a následne určí z neho priemer.

Parametre:	**`y`** : numpy vektor Vektor hodnôt cieľovej premennej.
Návratové hodnoty:	**`error`** : reálne číslo Hodnota priemernej chyby predikcie.

dcost_dvl(*indices, X, y, l*)

Metóda realizuje výpočet parciálnych derivácií účelovej funkcie podľa vybraných váh v_l určených parametrom *l*. Pomocou vektorového výpočtu umožňuje naraz vypočítať celý gradient účelovej funkcie. Spôsob výpočtu parciálnych derivácií závisí od nastavenej funkcie chyby (*error_type*), definície vzdialenosti (*metric*) a hodnotiacej funkcie vzdialenosti (*dist_weights*). Použité vzťahy sú uvedené v kapitole 3.2.4.

Parametre:	**indices** : numpy vektor alebo zoznam Indexy vybraných pozorovaní v prípade SGD (inak všetky indexy). **X** : numpy matica Súbor dát – matica, ktorej riadky sú všetky pozorovania. **y** : numpy vektor Vektor hodnôt cieľovej premennej. **l** : celé číslo alebo zoznam celých čísel Zoznam (vektor) poradových čísel váh premenných, podľa ktorých budú vypočítané parciálne derivácie.
Návratové hodnoty:	**gradient_vector** : vektor reálnych čísel Gradient – vektor parciálnych derivácií účelovej funkcie podľa váh vysvetľujúcich premenných.

update_var_weights(*gradient_vector*)

Metóda aktualizuje vektor váh vysvetľujúcich premenných podľa vzťahu 3.22 z kapitoly 3.2.3 práce na základe gradientu odovzdávaného pomocou parametra *gradient_vector* a parametra konštruktora *eta*, ktorým je určená rýchlosť konvergencie metódy najväčšieho spádu.

Parametre:	**gradient_vector** : vektor reálnych čísel Gradient účelovej funkcie – vektor jej parciálnych derivácií podľa váh vysvetľujúcich premenných.

correct_var_weights()

V procese výpočtu môžu vznikať záporné váhy premenných. Tento nežiadúci jav umožňuje metóda ošetriť tromi spôsobmi. Výber spôsobu ošetrenia závisí od nastavenia parametra konštruktora *check_neg_var_weights*:

zero – všetky záporné váhy sa vynulujú,

abs – záporné váhy budú nahradené ich absolútnou hodnotou,

new_eta – krok *eta* bude zmenšený tak, aby nevznikli záporné váhy.

select_vars(*X*, *y*, *xcols*, *initial_weights*)

Metóda je implementáciou samotného algoritmu pre výber premenných. Vyššie uvedené metódy volá v iteračnej slučke. Postupnosť volaní sa v cykle opakuje maximálne *n_iters*–krát. Algoritmus môže skončiť aj skôr, ak je norma vypočítaného gradientu menšia ako *eps*.

Pomocou atribútu *debug* možno povoliť alebo potlačiť informatívne výpisy na obrazovku a do logovacieho súboru. Vo výstupnom súbore sú informácie o sumách absolútnych hodnôt váh vysvetľujúcich premenných, o hodnotách chyby a trvaní výpočtu pre jednotlivé iterácie algoritmu. Na konci súboru je uvedený zoznam premenných s ich finálnymi váhami usporiadaný podľa váh v nerastúcom poradí.

Parametre:	**X** : numpy matica Súbor dát – matica, ktorej riadky sú všetky pozorovania. **y** : numpy vektor Vektor hodnôt cieľovej premennej. **xcols** : zoznam čísel alebo reťazcov Zoznam názvov vysvetľujúcich premenných. **initial_weights** : numpy vektor alebo zoznam reálnych čísel Štartovací bod, počiatočné váhy vysvetľujúcich premenných.

Návratové hodnoty:

col_names : zoznam celých čísel alebo reťazcov

Názvy premenných zoradené podľa finálnych váh.

col_numbers : zoznam celých čísel

Poradové čísla premenných zoradené podľa finálnych váh.

col_weights : zoznam reálnych čísel

Finálne váhy premenných zoradené v nerastúcom poradí.

final_error : reálne číslo

Finálna hodnota chyby.

Príklad použitia:

```
import numpy as np
from sklearn.preprocessing import StandardScaler
from neighbors_var_sel import NeighborsVarSelector

X = np.loadtxt('friedman200s500f_X.txt')
y = np.loadtxt('friedman200s500f_y.txt')

scl = StandardScaler()
X = scl.fit_transform(X)

selector = NeighborsVarSelector(lambda_const=0, eta=0.1,
           batch_size=0, n_iters=1000, metric='minkowski', p=1,
           dist_weights='exp', error_type='mse', eps=0.0001,
           check_neg_var_weights='zero', normalize_grad=True)

var_names, col_nrs, var_weights, error = selector.select_vars (X=X,
           y=y, xcols=np.arange(len(X[0])).tolist())

print('Column numbers: ', col_nrs)
print('Variable weights: ', var_weights)
```

3.2 Rekurzívna eliminácia premenných

Rekurzívna eliminácia premenných pomocou WkNN–FS je implementovaná triedou:

class `NeighborsVarSelElim`(*selector, maxvars_after_first=None, n_iters_exact=100, scaling=True*)

Parametre konštruktora:

`selector` : objekt triedy NeighborsVarSelector

Základná metóda výberu premenných, ktorá sa bude opakovane spúšťať.

`maxvars_after_first` : celé číslo, implicitne None

Počet premenných, ktoré sa budú brať do úvahy po prvom behu algoritmu.

`n_iters_exact` : celé číslo, implicitne 100

Maximálny počet iterácií pri jednej eliminácii.

`scaling` : boolovská hodnota, implicitne True

Príznak použitia štandardizácie súboru dát.

Metódy triedy `NeighborsVarSelElim` sú zhrnuté v tabuľke 3 – 2.

Tabuľka 3 – 2 Metódy triedy `NeighborsVarSelElim`

`select_vars`(*X, y, xcols, maxvars, initial_weights*)	výber premenných
`get_transformed_dataset`(*X, cols, col_weights*)	výpočet transformovaného súboru dát

`select_vars`(*`X, y, xcols, maxvars, initial_weights`*)

Metóda realizuje výber premenných rekurzívnou eliminációu. Pritom opakovane spúšťa základnú metódu WkNN–FS. Ak je príznak použitia štandardizácie nastavený na *True*, metóda najprv štandardizuje vstupný súbor dát *X*. Potom odovzdá dáta základnej metóde `select_vars`(*X, y, xcols, initial_weights*), ktorú vyvolá na objekte základného selektora premenných. Po prvom behu základnej metódy získa názvy premenných, ich poradové čísla a finálne váhy, ako aj finálnu hodnotu chyby. Premenné s nulovou váhou sa vynechajú. Ak zadaná hodnota parametra *maxvars_after_first* je menšia ako počet zostávajúcich nenulových váh, vynechajú

sa ďalšie premenné s najmenšími váhami tak, aby ostal špecifikovaný počet. Potom sa opakovane spúšťa základná metóda s menším počtom (*n_iters_exact*) iterácií, pričom pri každom behu sa eliminuje jedna premenná s najmenšou váhou. Eliminácia sa opakuje dovtedy, kým počet nenulových váh je väčší ako počet požadovaných premenných (*maxvars*).

Parametre:	**`X`** : numpy matica Súbor dát – matica, ktorej riadky sú pozorovania. **`y`** : numpy vektor Vektor hodnôt cieľovej premennej. **`xcols`** : zoznam celých čísel alebo reťazcov Zoznam názvov vysvetľujúcich premenných. **`maxvars`** : celé číslo Požadovaný počet vybraných premenných. **`initial_weights`** : numpy vektor alebo zoznam reálnych čísel Štartovací bod, počiatočné váhy vysvetľujúcich premenných.
Návratové hodnoty:	**`col_names`** : zoznam celých čísel alebo reťazcov Názvy vybraných premenných zoradené podľa finálnych váh. **`col_numbers`** : zoznam celých čísel Poradové čísla vybraných premenných zoradené podľa váh. **`col_weights`** : zoznam reálnych čísel Zoznam finálnych váh vybraných premenných v nerastúcom poradí. **`final_error`** : reálne číslo Finálna hodnota chyby.

`get_transformed_dataset(`*`X, cols, col_weights`*`)`

Metóda slúži na výpočet transformovaného súboru dát. Najprv štandardizuje pôvodný súbor dát *X*, potom z neho vyberie stĺpce špecifikované parametrom *cols* a vynásobí ich váhami *col_weights* zodpovedajúcich premenných, prípadne ich vhodnou modifikáciou.

Parametre:	**X** : numpy matica Súbor dát – matica, ktorej riadky sú pozorovania. **cols** : zoznam celých čísel Poradové čísla vybraných premenných. **col_weights** : zoznam reálnych čísel Zoznam váh vybraných premenných.
Návratové hodnoty:	**transformed_X** : numpy matica Transformovaný súbor dát.

Príklad použitia:

```
import numpy as np
from neighbors_var_sel import NeighborsVarSelector
from neighbors_var_sel_elimination import NeighborsVarSelElim

X = np.loadtxt('chin_X.txt')
y = np.loadtxt('chin_y.txt')

selector = NeighborsVarSelector(lambda_const=0, eta=0.1,
       batch_size=0, n_iters=1000, metric='sqeuclidean_mod',
       dist_weights='exp', error_type='cross_entropy', eps=0.0001,
       check_neg_var_weights='zero', normalize_grad=True)

selector_elim = NeighborsVarSelElim(selector=selector,
       maxvars_after_first=1000, n_iters_exact=10, scaling=True)

var_names, col_nrs, var_weights, error = selector_elim.select_vars
      (X=X, y=y-1, xcols=np.arange(len(X[0])).tolist(), maxvars=30,
       init_var_weights=np.zeros(shape=len(X[0])))

print('Column numbers: ', col_nrs)
print('Variable weights: ', var_weights)
```

```
22
23 transformed_dataset = selector_elim.get_transformed_dataset(X=X,
24         cols=col_nrs, col_weights=var_weights)
25
26 np.savetxt(X=transformed_dataset, fname='chin_transformed_X.txt')
```

4 Zovšeobecnená metóda mRMR

V práci bol implementovaný zovšeobecnený algoritmus mRMR, založený na maximalizácii funkcie Φ' vyjadrenej vzťahom 3.8 v kapitole 3.1.1 práce. Implementácia umožňuje použitie rôznych mier závislosti. Metóda je implementovaná triedou:

class `FilterMRMR`(*mrmr_scheme=MRMRScheme.MID, rel_function=None, with_self=False, lambdaweight=0.5*)

Parametre konštruktora:

`mrmr_scheme` : objekt enumeračnej triedy `MRMRScheme`

Kritérium mRMR, možné hodnoty:

- MRMRScheme.MID : kritérium MID (implicitná hodnota),
- MRMRScheme.MIQ : kritérium MIQ.

`rel_function` : funkcia so signatúrou relevance_function(x,y)

Miera závislosti premenných *x,y*, možné hodnoty:

- sklearn.metrics.mutual_info_score(*x,y*) : vzájomná informácia (impl.),
- mrmr_util.chi_square(*x,y*) : chí–kvadrát test,
- mrmr_util.pearson(*x,y*) : Pearsonov korelačný koeficient,
- mrmr_util.spearman(*x,y*) : Spearmanov korelačný koeficient,
- mrmr_util.d_cor(*x,y*) : korelácia vzdialeností.

Vstupom tejto funkcie je dvojica premenných, návratová hodnota je hodnota miery závislosti, resp. korelácie (reálne číslo).

with_self : boolovská hodnota, implicitne False

Príznak započítania korelácií premenných samých so sebou.

lambdaweight : reálne číslo, implicitne 0.5

Váha pre započítanie korelácie medzi vysvetľujúcimi premennými.

Metódy triedy `FilterMRMR` sú zhrnuté v tabuľke 4 – 1.

Tabuľka 4 – 1 Metódy triedy `FilterMRMR`

`select_vars`(*X, y, xcols, maxvars*)	výber premenných

`select_vars`(*`X, y, xcols, maxvars`*)

Metóda je implementáciou pažravého algoritmu na maximalizáciu funkcie Φ' definovanej v kapitole 3.1.1 práce. Do množiny vybraných premenných sa v každom kroku pridá jedna vysvetľujúca premenná, ktorá najviac zväčší hodnotu funkcie.

Algoritmus je implementovaný pre rôzne miery závislosti. Na výpočet hodnoty vzájomnej informácie je použitá funkcia `mutual_info_score()` zabudovaná v knižnici jazyka Python `sklearn.metrics`. Hodnota je počítaná s použitím prirodzeného logaritmu. Ďalšie korelačné miery sú počítané pomocou funkcií z knižnice `scipy.stats` okrem korelácie vzdialeností, ktorá je realizovaná vlastnou implementáciou. Implementácia algoritmu na výber premenných bola optimalizovaná tak, aby sa odstránilo opakované počítanie korelácie tej istej dvojice premenných.

Parametre:

X : numpy matica

Súbor dát – matica, ktorej riadky sú pozorovania.

y : numpy vektor

Vektor hodnôt cieľovej premennej.

xcols : zoznam celých čísel alebo reťazcov

Zoznam názvov vysvetľujúcich premenných.

maxvars : celé číslo

Maximálny počet vybraných premenných.

Návratové hodnoty:	**col_names** : zoznam celých čísel alebo reťazcov Zoznam názvov vybraných vysvetľujúcich premenných **col_numbers** : zoznam celých čísel Zoznam čísel stĺpcov, ktoré prislúchajú vybraným premenným

Príklad použitia:

```
import numpy as np
from filter_mrmr import FilterMRMR
from mrmr_util import MRMRScheme

X = np.loadtxt('madelon200s500f_discr_X.txt')
y = np.loadtxt('madelon200s500f_y.txt')

selector = FilterMRMR(mrmr_scheme=MRMRScheme.MID, with_self=False,
    lambdaweight=0.5)

var_names, col_nrs = selector.select_vars(X=X, y=y, xcols=np.arange
    (len(X[0])).tolist(), maxvars=15)
```

Zoznam tabuliek

Zoznam použitej literatúry

[1] PEDREGOSA, F. et al. Scikit-learn: Machine Learning in Python. In *Journal of Machine Learning Research.* Cambridge : MIT Press. ISSN 1533–7928, 2011, vol. 12, p. 2825–2830.

[2] OLIPHANT, T. E. Python for Scientific Computing. In *Computing in Science & Engineering.* ISSN 1521–9615, 2007, vol. 9, no. 3 p. 10–20.

[3] JONES, E., OLIPHANT, T. E., PETERSON, P. et al. *SciPy: Open Source Scientific Tools for Python* [online]. Verzia 3.6. [cit. 2017–12–19]. Dostupné na internete: http://www.scipy.org/

[4] HUNTER, J. D. Matplotlib: A 2D Graphics Environment. In *Computing in Science & Engineering.* ISSN 1521–9615, 2007, vol. 9, no. 3 p. 90–95.