Technická univerzita v Košiciach

Fakulta elektrotechniky a informatiky

Využitie strojového učenia v stávkovom systéme

Diplomová práca

2019 **Marek Ružička**

Technická univerzita v Košiciach

Fakulta elektrotechniky a informatiky

Využitie strojového učenia v stávkovom systéme

Diplomová práca

Študijný program:	Informatika
Študijný odbor:	Informatika
Školiace pracovisko:	Katedra počítačov a informatiky (KPI)
Školiteľ:	doc. Ing. Juraj Gazda PhD.
Konzultant:	

Košice 2019 **Marek Ružička**

Abstrakt v SJ

Základným cieľom záverečnej práce je analýza použiteľnosti algoritmov Extreme Gradient Boosting (XGBoost), náhodného lesa (RF), lineárnej a logistickej regresie a Support Vector Machines (SVM) implementovaných pomocou knižnice TensorFlow v jazyku python pri predikcii výsledkov futbalových zápasov v anglickej Premier League. Dôležitou časťou práce je tiež výber vhodných dát pre predikciu, vylepšovanie výkonu algoritmov pomocou nastavovania správnych parametrov a určenie najviac vyhovujúceho množstva dát ktoré sa použije pri trénovaní. Výstupom je webový portál využívajúci predikované hodnoty pre určenie kurzov zápasu.

Kľúčové slová

Stávkovanie, web, strojové učenie, šport, predikcia

Abstrakt v AJ

The main goal of this thesis is to analyze usefullness of prediction algithms logistic and linear regression, Extreme Gradient Boosting (XGBoost), Random Forest and Support Vector Machines (SVM) impleneted in TensorFlow library in python programming language in case of prediction of soccer match results in Premier League. Important part of the work is also to pick appropriate data set for prediction, improvement of algorithm parameters and estimating the most appropriate volume of data used for training. Output of the thesis is a web portal, which uses predicated values to determine betting odds.

Kľúčové slová v AJ

Betting, web, machine learning, sport, prediction

TECHNICKÁ UNIVERZITA V KOŠICIACH

FAKULTA ELEKTROTECHNIKY A INFORMATIKY

Katedra počítačov a informatiky

ZADANIE DIPLOMOVEJ PRÁCE

Študijný odbor: **Informatika**

Študijný program: **Informatika**

Názov práce:

Využitie strojového učenia v stávkovom systéme

The application of the machine learning principles in the sports betting systems

Študent: **Bc. Marek Ružička**

Školiteľ: **doc. Ing. Juraj Gazda, PhD.**

Školiace pracovisko: **Katedra počítačov a informatiky**

Konzultant práce:

Pracovisko konzultanta:

Pokyny na vypracovanie diplomovej práce:

1, V ľubovoľnom jazyku vytvorte webový portál pre online stávkovanie na futbalové zápasy, pričom pre určovanie stávkových kurzov využijete strojové učenie.

2, Dôraz nech je kladený na možnosti implementácie strojového učenia využitím knižnice TensorFlow v jazyku Python pomocou algoritmov lineárnej regresie, Support Vector Machines (SVM) a algoritmu Extreme Gradient Boosting (XGBoost).

3, Analyzujte úspešnosť jednotlivých algoritmov pri predpovedaní pravdepodobného výsledku zápasu a následného určovania kurzov podľa týchto hodnôt.

Jazyk, v ktorom sa práca vypracuje: slovenský

Termín pre odovzdanie práce: 26.04.2019

Dátum zadania diplomovej práce: 31.10.2018

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 26. 4. 2019 .....

Vlastnoručný podpis

Poďakovanie

Na tomto mieste by som chcel poďakovať doc. Ing. Jurajovi Gazdovi PhD. za vedenie mojej práce a Ing. Michalovi Kováčikovi za pomoc v oblasti stávkovania. Ďalej by som rád poďakoval Bc. Marekovi Kunštárovi za cenné rady o strojovom učení a Bc. Petrovi Dubóczimu za rady z oblasti futbalu. V neposlednom rade by som rád poďakoval svojej rodine za podporu počas celého štúdia.

Predhovor

Táto práca sa zaoberá predikciou zápasov pomocou strojového učenia. V práci využívame známe hodnoty získané za roky futbalových zápasov a vypočítavame z nich informácie, ktoré za pomoci strojového učenia vedú k lepšej predikcii zápasov ako v existujúcich riešeniach. V práci sa zaoberáme najvhodnejšou metódou predikcie, ako aj najvhodnejším algoritmom pre daný problém. Porovnávame viacero dátových sád a snažíme sa z nich vybrať takú, pri ktorej algoritmy vykazujú najlepšie výsledky.

Prvotná voľba témy tkvela v tom, že stávkovanie a hazard sú v dnešnej dobe všade prekvitajúcim odvetvím, ktoré potrebuje neustále vylepšovať svoje možnosti. Súčasné algoritmy určujúce kurzy stále nie sú vysoko presné, a preto je nutné túto oblasť stále skúmať.

Za pomoci rôznych metód, od základných úprav dátovej sady, cez vyhľadávanie najvhodnejších parametrov pre algoritmy strojového učenia, využitia klasickej predikcie aj nami navrhnutého postupu, až po porovnanie rôznych algoritmov sa teda snažíme vytvoriť model, ktorý by bol schopný s čo najvyššou účinnosťou predikovať správny výsledok zápasu.

Obsah

Úvod **1**

1 Formulácia úlohy **2**

2 Strojové učenie a algoritmy strojového učenia **3**

2.1 Lineárna regresia 4

2.2 Logistická regresia 4

2.3 Metóda podporných vektorov 6

2.4 Gradient boosting 9

2.5 Náhodný les 10

3 Určovanie stávkových kurzov **12**

3.1 Preberanie kurzov 13

3.2 Funkcia pre výpočet kurzu 13

3.3 Využitie strojového učenia pre určenie kurzu 13

3.3.1 Výber vhodnej ligy 14

3.3.2 Získavanie dát 14

3.4 Existujúce riešenia 14

4 Spracovanie a analýza dát **17**

4.1 Zoznámenie sa s dátami 17

4.2 Význam premenných 19

5 Prvé trénovanie modelov **27**

5.1 Úspešnosť na trénovacích dátach 28

5.2 Úspešnosť predikcie na testovacích dátach 29

5.3 Rozšírenie o dlhodobé štatistiky 34

6 Úprava parametrov jednotlivých algoritmov **42**

6.1 Úprava parametrov pre lineárnu regresiu 42

6.2 Úprava parametrov pre logistickú regresiu 43
6.3 Úprava parametrov pre SVM 44
6.4 Úprava parametrov pre XGB 46
6.5 Úprava parametrov pre náhodný strom 48

7 Predikcia konkrétneho výsledku 50
7.1 Úspešnosť predikcie v jednom kroku 51
7.2 Úspešnosť predikcie pri oddelenej predikcii výhry a prehry 52

8 Portál pre stávkovanie 59
8.1 Úvodná stránka 59
8.2 Registrácia 60
8.3 Odohrané a nadchádzajúce zápasy 61
8.4 Vytváranie stávok 62
8.5 Prehľad stávok 63

9 Vyhodnotenie 65

10 Záver 68

Zoznam použitej literatúry 70

Zoznam príloh 72

Zoznam obrázkov

2 – 1 Príklad lineárnej regresie. 4

2 – 2 Grafický priebeh funkcie logistickej regresie a rozdelenie množín pozitívnych a negatívnych bodov. 6

2 – 3 Porovnanie lineárneho, polynomiálneho a radiálneho jadra [5]. . . . 8

4 – 1 Atribúty v pôvodnom dátovom rámci. 17

4 – 2 Porovnanie celkového počtu jednotlivých výsledkov a celkovej a správnej predikcie týchto výsledkov v minulosti. 22

4 – 3 Porovnanie počtu predikcií každej možnosti (os y) výsledku pri skutočnom výsledku (os x). 22

4 – 4 Počty výhier a prehier tímu s viac bodmi vo forme a týždni, tímu s lepším gólovým skóre, a tímu s lepším ligovým umiestnením. . . . 24

4 – 5 Počty výhier, prehier a remíz domáceho tímu ak bol práve v sérii výhier alebo prehier. 25

4 – 6 Počty výhier, prehier a remíz hosťovského tímu ak bol práve v sérii výhier alebo prehier. 26

4 – 7 Počty výhier, prehier a remíz oboch tímov podľa ich predošlého zápasu. 26

5 – 1 Úspešnosť predikcie na trénovacích dátach pri ich postupnom znižovaní. 28

5 – 2 10 najdôležitejších premenných pre algoritmus gradient boosting, pre počet 16 až 19 sezón v trénovacích dátach. 33

5 – 3 Porovnanie úspešnosti predikcie na trénovacích dátach vzhľadom na počet sezón, bez informácií o dátume a s nimi, v rozšírenom dátovom rámci s agregovanými hodnotami. 38

7 – 1 Porovnanie úspešnosti predikcie na testovacích dátach s rôznou dĺžkou vzhľadom na počet sezón pre štyri sledované algoritmy. . . . 51

8 – 1 Úspešnosť predikcie pri oddelenej predikcii výhry a prehry algoritmom náhodného stromu. 60
8 – 2 Registračný formulár na webovom portáli pre stávkovanie. 61
8 – 3 Prvých 5 odohraných zápasov v tabuľke. 62
8 – 4 Prvých 5 nadchádzajúcich zápasov v tabuľke. 62
8 – 5 Formulár pre vytváranie stávok. 63
8 – 6 Prehľad stávok používateľa Marek. 64

Zoznam tabuliek

4 – 1 Počet výhier, prehier a remíz domáceho tímu v jednotlivých sezónach, ich priemer a celkový počet. 20
5 – 1 Porovnanie úspešnosti predikcie na testovacích dátach vzhľadom na počet sezón v trénovacích dátach, bez informácií o dátume, pri vzorke 70 zápasov. 30
5 – 2 Porovnanie úspešnosti predikcie na testovacích dátach vzhľadom na počet sezón v trénovacích dátach, s informáciou o dátume, pri vzorke 70 zápasov. 30
5 – 3 Porovnanie úspešnosti predikcie na testovacích dátach vzhľadom na počet sezón v trénovacích dátach, bez informácií o dátume, pri vzorke 380 zápasov. 31
5 – 4 Porovnanie úspešnosti predikcie na testovacích dátach vzhľadom na počet sezón v trénovacích dátach, s informáciou o dátume, pri vzorke 380 zápasov. 31
5 – 5 Početnosť výhier, prehier a remíz tímu s lepšou pozíciou v predchádzajúcom roku za prvých 10 sezón. 32
5 – 6 Porovnanie úspešnosti predikcie na testovacích dátach vzhľadom na počet sezón v trénovacích dátach , bez informácií o dátume, pri vzorke 70 zápasov, s novou dátovou sadou. 35
5 – 7 Porovnanie úspešnosti predikcie na testovacích dátach vzhľadom na počet sezón v trénovacích dátach, s informáciou o dátume, pri vzorke 70 zápasov, s novou dátovou sadou. 36
5 – 8 Porovnanie úspešnosti predikcie na testovacích dátach vzhľadom na počet sezón v trénovacích dátach, bez informácií o dátume, pri vzorke 380 zápasov, s novou dátovou sadou. 36

5 – 9 Porovnanie úspešnosti predikcie na testovacích dátach vzhľadom na počet sezón v trénovacích dátach, s informáciou o dátume, pri vzorke 380 zápasov, s novou dátovou sadou. 37
5 – 10 Porovnanie úspešnosti predikcie na testovacích dátach vzhľadom na počet sezón v trénovacích dátach , bez informácií o dátume, pri vzorke 70 zápasov, s upravenou novou dátovou sadou. 39
5 – 11 Porovnanie úspešnosti predikcie na testovacích dátach vzhľadom na počet sezón v trénovacích dátach, s informáciou o dátume, pri vzorke 70 zápasov, s upravenou novou dátovou sadou. 40
5 – 12 Porovnanie úspešnosti predikcie na testovacích dátach vzhľadom na počet sezón v trénovacích dátach, bez informácií o dátume, pri vzorke 380 zápasov, s upravenou novou dátovou sadou. 40
5 – 13 Porovnanie úspešnosti predikcie na testovacích dátach vzhľadom na počet sezón v trénovacích dátach, s informáciou o dátume, pri vzorke 380 zápasov, s upravenou novou dátovou sadou. 41
7 – 1 Úspešnosť predikcie algoritmu SVM pri oddelenej predikcii výhry a prehry, vzorka 70 zápasov. 53
7 – 2 Úspešnosť predikcie algoritmu SVM pri oddelenej predikcii výhry a prehry, vzorka 380 zápasov. 53
7 – 3 Úspešnosť predikcie algoritmu lineárnej regresie pri oddelenej predikcii výhry a prehry, vzorka 70 zápasov. 54
7 – 4 Úspešnosť predikcie algoritmu lineárnej regresie pri oddelenej predikcii výhry a prehry, vzorka 380 zápasov. 54
7 – 5 Úspešnosť predikcie algoritmu logistickej regresie pri oddelenej predikcii výhry a prehry, vzorka 70 zápasov. 55
7 – 6 Úspešnosť predikcie algoritmu logistickej regresie pri oddelenej predikcii výhry a prehry, vzorka 380 zápasov. 55
7 – 7 Úspešnosť predikcie algoritmu XGB pri oddelenej predikcii výhry a prehry, vzorka 70 zápasov. 56

7 – 8 Úspešnosť predikcie algoritmu XGB pri oddelenej predikcii výhry a prehry, vzorka 380 zápasov. 57
7 – 9 Úspešnosť predikcie algoritmu náhodného stromu pri oddelenej predikcii výhry a prehry, vzorka 70 zápasov. 57
7 – 10 Úspešnosť predikcie algoritmu náhodného stromu pri oddelenej predikcii výhry a prehry, vzorka 380 zápasov. 58

Zoznam symbolov a skratiek

AT z angl. Away Team - hosťujúci tím. 22

CSV z angl. Comma Separated Values - formát tabuľkových dát, ktoré sú oddelené čiarkou. 14

D z angl. draw - remíza. 19

GC z angl. goals conceded - inkasované góly. 22

HT z angl. Home Team - domáci tím. 22

L z angl. loss - prehra. 19

M z angl. match - zápas. 24

P z angl. points - bodové ohodnotenie. 23

rbf z angl. radial basis function - radiálna bázová funkcia. 7, 8

SVM z angl. Support vector machine - algoritmus strojového učenia. 6–8, 27, 29, 37, 38, 50–52, 66, 68

W z angl. win - výhra. 19

XGB z angl. Extreme Gradient Boosting - implementácia algoritmu gradient boosting v knižnici scikit learn. 29, 37, 41, 44, 48, 50, 52, 55, 66, 68, 69

Slovník termínov

Strojové učenie - z angl. Machine learning, oblasť umelej inteligencie, zaoberajúca sa metódami a algoritmami, ktoré umožňujú programu učiť sa a následne adekvátne reagovať na rôzne vstupné hodnoty bez toho, aby bol na ne explicitne naprogramovaný, iba na základe informácií, ktoré sa naučil [2].

Parser Program alebo časť programu určená na interpretáciu vstupu pre počítač rozpoznávaním kľúčových slov a analýzou štruktúry výrazu.

Mriežkové vyhľadávanie Z angl. grid search. Jedná sa o algoritmus hľadania najvhodnejšej hodnoty parametrov v algoritmoch strojového učenia ich postupným skúšaním.

Pretrénovanie Z angl. overfitting. Jedná sa jav, kedy sa algoritmus strojového učenia naučí na dátach niečo, čo nemusí nutne platiť. Stupeň funkcie je v tomto prípade vyšší ako by mal byť. Výsledkom je vysoká úspešnosť pri trénovaní, no slabšie výsledky pri testovaní. Opakom je podtrénovanie, z angl. underfitting, kde je stupeň funkcie nižší ako by mal byť.

Úvod

V dnešnej dobe veľkých dát a stále nových technológií musia všetky odvetvia držať krok s dobou. Vďaka obrovskému množstvu dát, ktoré sa v dnešnej dobe dajú získať o konkrétnej doméne, či už historických dát, alebo aktuálneho správania sa prostredia či ľudí v rámci domény, vieme s vysokou presnosťou určiť možný vývoj v danej doméne. Aplikáciou neurónových sietí a strojového učenia vieme predpovedať či už správanie zákazníka, či, ako v našom prípade, najpravdepodobnejší výsledok futbalového zápasu. Výsledok zápasu môže ovplyvňovať veľké množstvo faktorov. V tejto práci budeme vychádzať z niekoľkých predpokladov, ktoré si postupne dokážeme, prípadne vyvrátime, a na základe našich zistení sa budeme posúvať ďalej. Zaujímavou hypotézou napríklad je, že domáci tím má väčšiu šancu na výhru, ako hosťujúci. Hypotézou číslo dva je, že stávkové kancelárie, ktoré majú dlhoročnú prax, by mali mať už teraz veľkú úspešnosť pri odhadovaní výsledku. Ďalej bude zaujímavé sa pozrieť, ako ovplyvňuje výsledok súčasná forma oboch tímov, ale taktiež aj ich dlhodobý trend vo výsledkoch.

Cieľom diplomovej práce je vytvoriť model moderného stávkového systému, v ktorom sa na určovanie kurzov bude využívať strojové učenie, ktoré natrénujeme pomocou výsledkov zápasov z minulosti. Preto budeme preverovať správnosť našich hypotéz a pokúšať sa čo najlepšie nastaviť dáta tak, aby sme vedeli s vysokou presnosťou určiť, ako zápas skončí.

1 Formulácia úlohy

Na začiatku práce je nutné zadefinovať úlohy, ktoré je potrebné splniť na to, aby sme sa dopracovali k funkčnému prototypu stávkového portálu. Hlavným cieľom tejto práce je analýza jednotlivých algoritmov, a až na jej základe je možné vytvorenie portálu. Preto delíme prácu z praktického hľadiska na dve časti:

1. Analýza algoritmov strojového učenia
    - Vytvorenie dostatočne komplexnej dátovej sady, pomocou ktorej bude možné vytvoriť modely a natrénovať algoritmy strojového učenia.
    - Vyhľadanie najvhodnejších parametrov pre algoritmy strojového učenia tak, aby bola úspešnosť predikcie čo najvyššia pri jednotlivých dátových sadách.
    - Určenie najvhodnejšieho objemu dát pri ktorom budú mať algoritmy najvyššiu úspešnosť.
    - Vyhodnotenie najvhodnejšieho (prípadne viac) algoritmu, ktorý sa na riešenie predikcie najlepšie hodí.
2. Vytvorenie modelu webového portálu pre zobrazenie predikovaných hodnôt v užívateľsky blízkej forme.

2 Strojové učenie a algoritmy strojového učenia

V dnešnej dobe sa v informatike často stretávame s pojmom strojové učenie s predpokladom, že intuitívne vieme, čo by to vlastne malo byť. Tento prístup často vedie k tomu, že si význam strojového učenia interpretujeme chybne. Čo to teda je? Strojové učenie je podľa [3] oblasť vedeckého skúmania, ktoré sa zameriava na indukčné a iné algoritmy, ktoré majú schopnosť sa učiť. Z toho prakticky vyplýva, že strojové učenie je spôsob skladania funkcie zo známych dát tak, aby sme ďalej mohli predpovedať dáta, ktoré známe nie sú. Podstatou teda je, že algoritmy strojového učenia vytvárajú model z existujúcich vstupov tak, aby boli schopné vytvárať dátami riadené predikcie alebo rozhodnutia [14], miesto toho, aby sa riadili preddefinovanou postupnosťou inštrukcií alebo príkazov. Výsledkom strojového učenia je teda model, ktorý je schopný z doposiaľ nevidených dát, na základe dát ktoré sú známe, určiť predikciu či rozhodnutie. Aplikovanie strojového učenia je známe v mnohých sférach informatiky. Ako príklad využitia strojového učenia v ktorom je výsledkom rozhodnutie môžeme spomenúť optické rozpoznávanie znakov, v ktorom na základe rôznych obrázkov znakov má systém rozhodnúť, o aký znak sa jedná. Strojové učenie, ktorého výsledkom je predikcia bude napríklad náš systém, v ktorom sa pokúsime odhadnúť správny výsledok zápasu ešte pred tým, ako sa uskutoční. Takéto odhadovanie náhodnej hodnoty na základe znalosti známych veličín sa nazýva regresná analýza.

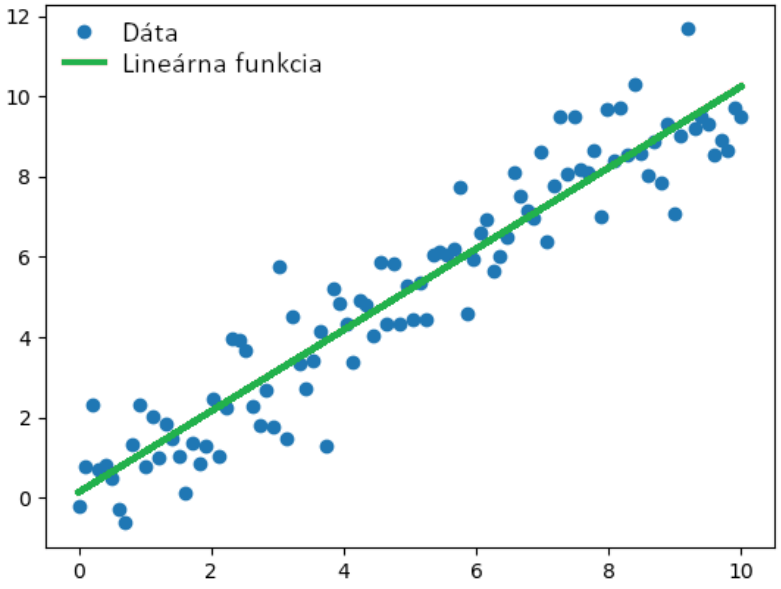
Aký je potom rozdiel medzi samotným strojovým učením a algoritmami strojového učenia? Strojové učenie je všeobecný pojem, označujúci akýkoľvek spôsob vytvorenia výsledného modelu. Keďže sa časom ukázalo, že pre strojové učenie môžeme uplatniť viacero prístupov v závislosti na charaktere dát, začali sa rozlišovať rôzne algoritmy strojového učenia. V nasledujúcich podkapitolách si rozoberieme princíp tých, ktorými sa budeme v tejto práci zaoberať.

2.1 Lineárna regresia

Lineárna regresia je v štatistike spôsob modelovania vzťahu medzi skalárnou premenou y, ktorú máme na výstupe, s jednou alebo viacerými vstupnými premennými označovanými x_n. Ak je n=1, nazývame regresiu jednoduchou, inak hovoríme o viacnásobnej lineárnej regresii [15]. Ako už napovedá názov, v lineárnej regresii sa buduje lineárna funkcia, respektíve lineárny model na predikciu alebo rozhodovanie. Funkcia, ktorú sa snažíme vytvoriť môže mať tvar:

$$y = \beta_1 * x_1 + \beta_2 * x_2 + ... + \beta_n * x_n + \alpha, \tag{2.1}$$

kde n je počet premenných x a β_n je koeficient, ktorým pripisujeme váhu jednotlivých premenných [16]. α je konštanta, ktorú pripočítame k výsledku pre dosiahnutie čo najpresnejšieho modelu. Grafický príklad lineárnej regresie môžete vidieť na 2 – 1.



Obrázok 2 – 1: Príklad lineárnej regresie.

2.2 Logistická regresia

Lineárna regresia sa zo svojej podstaty využíva najmä v prípade, že sú dáta distribuované v grafe tak, že sa dá vytvoriť spojitá funkcia. Lineárnu regresiu doká-

žeme adaptovať aj na polynomiálne alebo monotónne funkcie [16]. Problém nastáva, ak máme dáta distribuované vo viacerých triedach, ktoré sú evidentne odlišné. Je zrejmé, že ak by sa hodnoty týchto tried nahádzali blízko seba, lineárne nevieme množiny rozdeliť tak, aby sme si boli istý správnosťou rozdelenia. Prakticky by sa nám vždy vo výsledku objavili aj chybne určené hodnoty. Z tohto dôvodu sa využíva pri problémoch tohto typu logistická regresia. Napriek názvu regresia, sa jedná o klasifikačnú metódu, v ktorej určíme v rozsahu (0,1), nakoľko je pravdepodobné že vzorka spadá do danej triedy [16]. Názov logistická regresia vychádza z využitia logistickej funkcie, tiež zvanej sigmoid, ktorá má nasledovný (základný) tvar:

$$\sigma(x) = 1/(1 + e^{-x}). \tag{2.2}$$

Grafické znázornenie logistickej funkcie aj s rozdelením množín bodov je na obrázku 2 – 2. Je zrejmé že funkcia má hodnotu 0.5 pri x=0 a má dve asymptoty, 0 a 1. Vďaka tomu sa dá pravdepodobnosť, že sa vzorka nachádza v triede, vyjadriť takto:

$$P(y|x) = \sigma(x; w). \tag{2.3}$$

Ako je zmienené v [16], aplikáciou logaritmu môžeme následne nájsť optimálne parametre, čo vyjadríme vzťahom:

$$L(w; y) = \sum_i (\log P(y_i|x_i, w)). \tag{2.4}$$

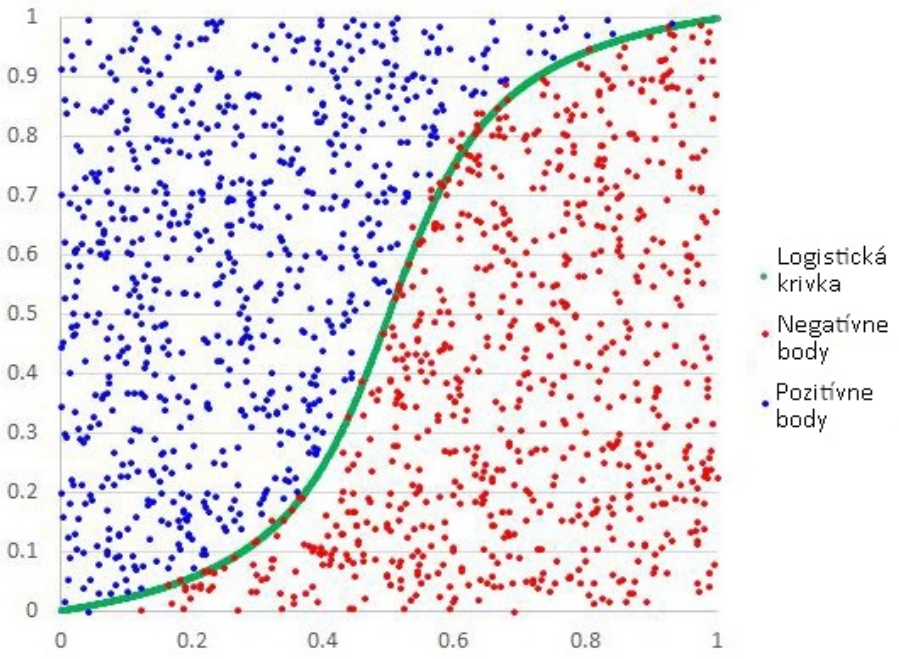
Z tohto vieme odvodiť nasledovný optimalizačný vzťah:

$$J(w; y) = -\sum_i (y_i \log \sigma(z_i) + (1 - y_i) \log(1 - \sigma(z_i)). \tag{2.5}$$

Ak y=0, prvý výraz sa rovná 0 a ten druhý je rovný $\log(1 - x)$, čo je logaritmickou pravdepodobnosťou triedy 0. V opačnom prípade, kedy y=1, je druhý výraz 0 a prvý výraz vyjadruje logaritmickú pravdepodobnosť x. Pre využitie v praxi tento postup zjednodušíme takto:

$$H(X) = \sum_{x \in X} (p(x) \log_2 q(x)). \tag{2.6}$$

V konečnom dôsledku teda vieme každý jeden bod umiestniť do jednej z tried. Pri dobrých trénovacích dátach môžeme určiť parametre tak, že určenie správneho výsledku je vysoko pravdepodobnejšie, než pri lineárnej regresii.



Obrázok 2 – 2: Grafický priebeh funkcie logistickej regresie a rozdelenie množín pozitívnych a negatívnych bodov.

2.3 Metóda podporných vektorov

Metóda podporných vektorov, ďalej iba SVM (z angl. Support Vector Machine), je algoritmus strojového učenia s učiteľom, tzn. s dátami, na ktorých sa učí, primárne využívaný pre klasifikáciu dát do niektorej skupiny, alebo na regresnú analýzu. Základným princípom metódy SVM je snaha nájsť lineárny klasifikátor, ktorý predelí množinu trénovacích dát na dve triedy, v ktorých sa vyskytuje čo najmenej, alebo najlepšie nula, prvkov inej triedy. V prípade našej úlohy si zjednodušene môžeme predstaviť SVM ako algoritmus, ktorý dáta delí do triedy kde jeden tím zápas vyhrá a do triedy, kde tento tím nevyhrá. Formálne vieme tento algoritmus zadefinovať tak, že pre každý súbor dát zadefinovaný ako $X = x_1, X_2, ..., x_i$, kde $x_i \in R^m$, a sú-

bor výsledkov (zjednodušene len binárny problém) $X = y_1, y_2, ..., y_i$, kde $y_i \in \{0, 1\}$ sa snažíme nájsť čo najlepšie rozdeľujúcu funkciu, pre ktorú platí:

$$w^T x + b = 0, \text{kde } w = \begin{bmatrix} w_1 \\ ... \\ w_m \end{bmatrix}, x = \begin{bmatrix} x_1 \\ ... \\ x_m \end{bmatrix} [16]. \tag{2.7}$$

Väčšina dátových sád však nemá výsledky rozmiestnené tak, aby sa dali predeliť lineárnym prístupom. U mnohých sa výsledkové triedy rôzne prekrývajú, mnohokrát je jedna z množín dokonca rozdelená na rôzne časti. Príklad takejto množiny môže byť napr. vývoj teploty počas dňa, kde môžeme ráno aj večer očakávať nižšie teploty ako práve uprostred dňa, a tým pádom je trieda chladnejších hodín predelená množinou tých teplejších. V tomto prípade by pravdepodobne bolo možné uplatniť lineárne SVM, existujú však aj iné typy SVM ktoré sa dajú využiť, vzhľadom na charakter dát. Vzhľadom na charakter funkcie, ktorou v SVM budeme deliť množiny, hovoríme o rôznych typoch jadra, z angličtiny kernel. V tejto práci otestujeme tri najpoužívanejšie typy jadra, a to lineárne, polynomiálne a rbf (z angl. radial basis function).

Čo sa týka lineárneho jadra, pracuje na vyššie popísanom princípe a snaží sa predeliť klasifikačné triedy na základe lineárnej funkcie, čím sa veľmi neodlišuje od lineárnej regresie.

Polynomiálne jadro je založené na polynomiálnej funkcii, ktorá má tvar:

$$K(x, y) = (x^T * y + r)^c [16]. \tag{2.8}$$

X a y sú v tejto funkcii vektory vypočítané z trénovacích alebo testovacích súborov, konštanta c, ktorá sa vo frameworku scikit-learn v jazyku python nastavuje parametrom degree predstavuje exponent a konštanta r nastaviteľná parametrom coef0 predstavuje odsadenie polynomiálnej funkcie. Ak je konštanta r=0, hovoríme o homogénnej funkcii [16][13]. Táto metóda dokáže jednoducho riešiť problémy s veľkým množstvom premenných a riešiť rôzne nelineárne problémy, čo však môže a

často aj má za následok vyššie nároky na zdroje[16]. Vďaka tomu dokážeme týmto algoritmom riešiť mnohé komplexné problémy.

Jadro typu rbf je založené na radiálnej bázovej funkcii, ktorá má nasledujúci tvar:

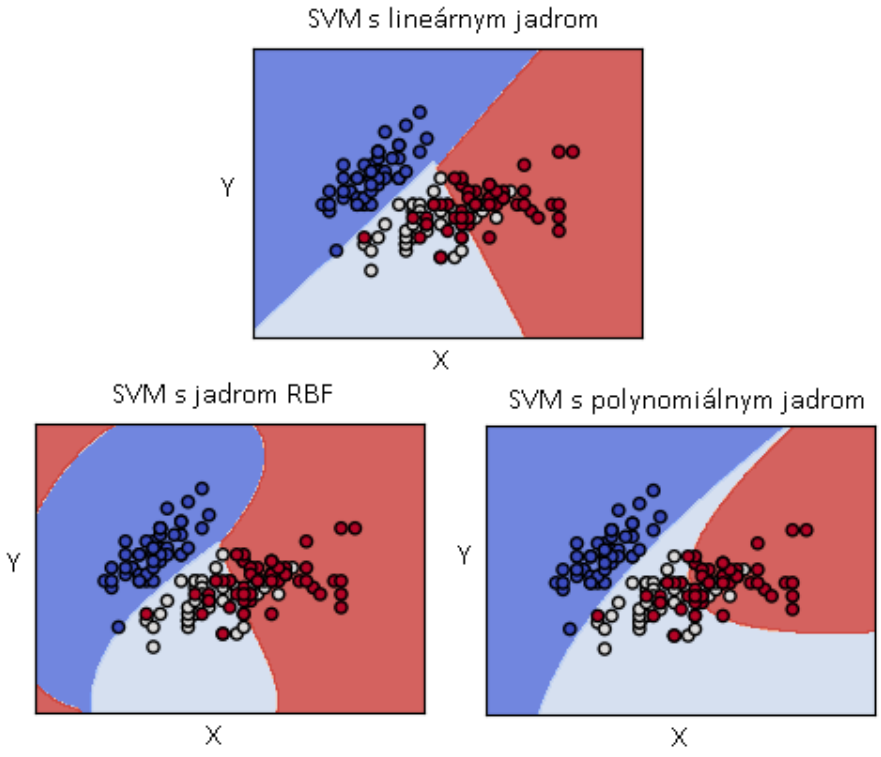
$$K(x, y) = e^{-\gamma|x-y|^2}[16]. \tag{2.9}$$

V knižnici scikit-learn je práve toto jadro pre SVM prednastavené. Parameter γ určuje amplitúdu funkcie, ktorá nie je ovplyvnená smerom, iba vzdialenosťou medzi x a y. Keďže γ v skutočnosti predstavuje funkciu v tvare:

$$1/2 * \sigma, \tag{2.10}$$

toto číslo môže nadobudnúť rôzne veľké hodnoty [4].

Pre porovnanie, na obrázku 2 – 3 môžeme vidieť, ako sa množiny dát oddelia pri použití spomínaných jadier. Je evidentné, že výber správneho jadra môže veľmi ovplyvniť schopnosť modelu určovať správne výsledky.



Obrázok 2 – 3: Porovnanie lineárneho, polynomiálneho a radiálneho jadra [5].

2.4 Gradient boosting

Jedná sa o techniku strojového učenia pre regresné a klasifikačné problémy založenú na spájaní slabých predikčných modelov, zvyčajne rozhodovacích stromov. Z praktického hľadiska ide teda o spájanie slabých metód strojového učenia tak, aby sme v závere získali lepšie výsledky. Môžeme povedať, že táto metóda nám dovoľuje vytvoriť rozhodovací strom krok za krokom, s cieľom minimalizovať cieľovú stratovú funkciu. Funkcia tohto algoritmu môže byť všeobecne zapísaná ako:

$$y = \sum_i \alpha_i f_i(x)[16]. \tag{2.11}$$

V tomto zápise $f_i(x)$ reprezentuje tzn. weak learner, slabú funkciu pre učenie. Algoritmus je navrhnutý tak, aby v každom kroku vykonávania pridal nový rozhodovací strom, čím sa zníži celková stratová funkcia využitím metódy najväčšieho klesania (z angl. gradient descent) [16]. V pythone je tento algoritmus v súčasnosti najlepšie optimalizovaný pomocou knižnice XGBoost. Tento algoritmus má najviac možností úpravy parametrov, čo môže značne ovplyvniť jeho výkon. Preto sa v tejto podkapitole pozrieme skôr na parametre, ako na formálny popis. Existuje niekoľko desiatok parametrov pre ladenie XGBoostu, preto vyberieme len tie dôležitejšie.

Parametrom booster si vieme vybrať konkrétny typ boostra, buď lineárny, stromový, alebo tzn. dart, ktorý funguje obdobne ako stromový, ale vynecháva v niektorých prípadoch časť výsledkov, čo môže viesť k problémom pri zlej implementácii. Každý druh boostra má aj svoje vlastné parametre na ladenie. Z parametrov pre ladenie ďalej spomenieme:

- learning_rate je parameter, ktorým zamedzujeme pretrénovaniu modelu po každom kroku.
- gamma - premenná, ktorá určuje minimálnu redukciu straty potrebnú na vytvorenie ďalších rozdelení na listoch stromu. Čím je toto číslo väčšie, tým je algoritmus konzervatívnejší.

- max_depth - je maximálna hĺbka stromu, čím je toto číslo väčšie, tým viac hrozí pretrénovanie. Pri nízkych hodnotách však nebude strom schopný vytvárať správne rozhodnutia.
- min_child_weight - najnižšia váha potrebná v podstrome, ak dospejeme do bodu, kde už nebude možné vytvoriť list s váhou väčšou ako toto číslo, strom sa prestane deliť.
- colsample_bytree je pomer čiastkových vzoriek pri vytváraní jednotlivých stromov. Tento proces sa deje vždy pri vytváraní nového stromu. Podobne existuje rozdeľovanie podľa uzlov a stupňov.
- lambda a alpha - L2 a L1 regularizačné parametre pre váhu, zvyšovaním ich hodnôt získavame konzervatívnejší model
- scale_pos_weight - ovládanie rovnováhy medzi kaladnými a zápornými váhami, užitoné v nevyrovnaných triedach.
- n_estimators určuje počet stromov v rozhodovacom procese.
- subsample - rozdeľovanie dát do čiastkových vzoriek, na prevenciu overfittingu. Ak je nastavená hodnota tohto parametra na 0.5, algoritmus automaticky rozmiestni vzorky náhodným spôsobom do jednotlivých stromov. Tento proces sa opakuje pri každom kroku algoritmu.

2.5 Náhodný les

Náhodný les je súbor rozhodovacích stromov vytvorených pomocou náhodných vzoriek pôvodného dátového rámca, ktoré majú rozdielne vlastnosti [16]. Miesto toho, aby sa v modeli hľadala najvhodnejšia možnosť, sa v tomto algoritme využíva náhodná časť dát a snažíme sa určiť rozdelenie, ktoré najlepšie vystihuje dané dáta. Ako výsledok máme množinu stromov, čiže les, kde každý strom má svoje vlastné pravidlá a tým pádom iný výstup. Výsledok je nakoniec určený pomocou voľby.

Výsledok z množiny výsledkov za ktorý "hlasovalo"najväčšie množstvo stromov je potom považovaný za najpravdepodobnejší. V module scikit-learn sa ale stretávame s iným prístupom. Výsledky sa sčítajú a vytvorí sa priemerná hodnota zo všetkých hodnôt. V prípade binárneho modelu sú oba prístupy rovnaké, no v prípade väčšej množiny výsledkov môže vytváranie priemernej hodnoty priniesť lepšie výsledky.

3 Určovanie stávkových kurzov

V stávkovom systéme je prioritou prinášať zisk bez obmedzovania používateľov. To prakticky znamená, že je potrebné na jednotlivé zápasy a podujatia určiť stávkový kurz tak, aby bola vyššia pravdepodobnosť prehry než výhry. Pravdepodobnosť určitého výsledku sa určí niektorým so spôsobov, ktoré spomenieme v tejto kapitole. Následne, sa určí kurz, a to jednoduchým vzorcom:

$$k = 1/P_a, \tag{3.1}$$

kde P_a predstavuje pravdepodobnosť výsledku [1]. Ako príklad môžeme uviesť futbalový zápas dvoch tímov - A a B. Zápas sa môže pre tím A skončiť výhrou (a), prehrou (b), alebo remízou (x). Povedzme že pravdepodobnosť výhry tímu A je 40%, potom kurz na výsledok a bude $1/P_a = 1/0.4 = 2,5$. Výsledný súčet pravdepodobností všetkých troch možností $P_a + P_b + P_x$ je 100%. Pri takomto kurze by však bola štatisticky ziskovosť nulová, pretože výhry by vyvážili prehry. Kvôli tomu je nutné kurzy umelo upraviť tak, aby celkové príjmy stávkového systému boli vyššie, než výdavky. V praxi sa kurz umelo zníži tak, aby súčet pravdepodobností jednotlivých možností výsledku zápasu bol vyšší, ako 100%. Percentuálny rozdiel medzi reálnym súčtom a 100% následne predstavuje výnos systému. Pokiaľ nie je tento rozdiel vysoký, systém môže prinášať zisky bez toho, aby sa výrazne znížila šanca užívateľov na možnú výhru. Na určenie dobrého kurzu môžeme využiť niektorú z nasledujúcich možností.

- Preberanie stávkového kurzu od iných spoločností.
- Vytvorenie funkcie pre výpočet kurzu.
- Využitie strojového učenia pre určenie kurzu.

V nasledujúcich podkapitolách si rozoberieme jednotlivé spôsoby a ich výhody a nevýhody.

3.1 Preberanie kurzov

Preberanie kurzov je najjednoduchšou alternatívou získania aktuálneho kurzu na určitý zápas, alebo podujatie. Jedná sa prakticky o využitie kurzu, ktorý stanovila iná stávková spoločnosť alebo bookmaker. Z hľadiska náročnosti na programovanie a matematické postupy sa jedná o najmenej náročnú možnosť, nie je pri nej potrebné spraviť skoro nič. Problémom tohto prístupu však môže byť dostupnosť dát. Otázkou je, čo ak sa niečo stane so zdrojom, z ktorého informácie čerpáme. Ďalšia otázka spočíva v cene kurzov, využitie niekým vypočítaných kurzov na komerčné účely môže byť spoplatňované, čiže v konečnom dôsledku môže stáť spoločnosť viac peňazí, než vývoj vlastného systému.

3.2 Funkcia pre výpočet kurzu

Vytvorenie kvalitnej funkcie na výpočet kurzu je veľmi dôležité. Zle definovaná funkcia môže predpovedať kurzy nesprávne, čo môže viesť k veľkým finančným stratám spoločnosti a v konečnom dôsledku až ku krachu. V prípade že vytvárame funkciu sami, je potrebné vziať v úvahu väčšie množstvo faktorov. Správne zohľadnenie všetkých faktorov je pre človeka náročnou úlohou a trvá aj roky, kým sa vytvorí funkcia dostatočne presná pre správne určenie stávkového kurzu s vysokou presnosťou. Preto sa tomuto prístupu v našej práci vyhneme a použijeme na určenie stávkových kurzov strojové učenie, ktoré má lepšiu schopnosť zhodnotiť podstatné faktory v dátach.

3.3 Využitie strojového učenia pre určenie kurzu

Pre určovanie pravdepodobnosti výhry, prehry a remízy sa dá efektívne využiť strojové učenie. Na efektívne určenie kurzu nie je nutné poznať všetky okolnosti jednotlivých zápasov, ako napríklad červené karty či fauly. Mnohé z týchto udalostí sú síce z hľadiska zápasu zaujímavé, no neovplyvnia v zásade jeho výsledok. Hráč totiž môže zlyhať, no hodnotenie zlyhaní hráčov je často pomerne subjektívnym názorom rozhodcu a preto môže dostať hráč kartu často aj neprávom. Výsledok zápasu

závisí hlavne od schopností tímu spolupracovať ako celok. Preto sa nezameriame na jednotlivých hráčov, no na tím samotný. Prvým dôležitým krokom je vytvorenie kvalitného dátového rámca, ktorý sa využije pre natrénovanie modelu strojového učenia.

3.3.1 Výber vhodnej ligy

Na natrénovanie neurónovej siete a následné testovanie správnosti v praxi musíme vybrať vhodnú ligu. Z dôvodu obdobia, kedy sa bude na práci pracovať, nie je vhodné vybrať si ligu, ktorá nemá zápasy v zimných mesiacoch. Z tohto dôvodu sme vylúčili slovenské národné ligy, v ktorých sa neodohrávajú zápasy v januári ani februári. Vyhovujúcou ligou z hľadiska obdobia hier aj z hľadiska množstva historických dát je anglická Premier League [10]. Liga má 38 herných týždňov a každý z nich 10 zápasov. Na oficiálnej stránke majú pomerne kvalitne spracované dlhodobé dáta a podrobnejšie informácie sa dajú získať aj z iných zdrojov[10].

3.3.2 Získavanie dát

Podrobnejšie štatistiky zápasov sú zvyčajne spoplatňované pre ich komerčnú využiteľnosť. Základné štatistiky, ako výsledok zápasu, dátum podujatia, či kurzy na jednotlivé zápasy sa dajú získať z rôznych portálov. My sme si vybrali portál www.betexplorer.com. Dáta sú tu síce dostupné, no neponúkajú možnosť stiahnuť ich. Preto sme vytvorili parser, ktorým sme dáta stiahli a skonvertovali na formát CSV, ktorý je vhodný pre použitie v strojovom učení. Dáta obsahujú dátum zápasu, výsledný stav a kurzy, ktoré boli nastavené na výhru, prehru a remízu.

3.4 Existujúce riešenia

Na záver kapitoly by sme sa mali ešte pozrieť na niektoré existujúce riešenia. Tieto riešenia nám totiž nastavujú v závere zrkadlo, ako úspešní skutočne sme. V tejto chvíli si musíme uvedomiť, že problém predikcie futbalových výsledkov nie je binárny,

ale ternárny, čo značne znižuje celkovú úspešnosť.

V nasledujúcej kapitole si pri analýze dát opíšeme podrobnejšie aj kurzy určené portálom www.oddsportal.com. Zatiaľ uvedieme len fakt, že celková úspešnosť určenia správneho výsledku je iba 53%, čo by bolo v prípade binárneho problému veľmi nízke. Lepšie výsledky sú dosiahnuté určením najmenej pravdepodobného výsledku, kde sa dostávajú predikcie portálu tesne nad 78%. Čo sa týka remíz, za dobu 21 sezón, boli predikované iba 9 krát. Práve toto je pravdepodobne hlavný dôvod nízkej úspešnosti.

Zaujímavou aplikáciou, ktorá sa zaoberá podobnou problematikou ako naša práca a ponúka aj percentuálne predikcie pravdepodobností je kickoff.ai. Tento portál uvádza, že využíva strojové učenie a pomocou bayesovej funkcie určuje pravdepodobnosť jednotlivých výsledkov. Z 1670 zápasov, ktoré do doby merania (22.3.2019) predikovali, sme vybrali vzorku 100 posledných zápasov. Na tejto vzorke bolo správne predikovaných 56 zápasov, chybne bolo predikovaných 44. Čo sa týka najmenšej pravdepodobnosti, opäť môžeme pozorovať vyššiu úspešnosť. Zo 100 zápasov sa výsledok označený ako najmenej pravdepodobný vyskytol len 21-krát. To znamená, že v 79% prípadov sa podarilo určiť správne, ako zápas nedopadne. 19 zápasov vo vzorke skončilo remízou, no ani jeden krát nebola remíza predikovaná ako najpravdepodobnejší výsledok. Rovnako ako v predchádzajúcom prípade, dôvodom pravdepodobne je, že v dátach sa vždy ukazuje jeden z tímov ako štatisticky lepší.

V podobnej diplomovej práci nazvanej Machine learning for Soccer analytics môžeme sledovať podobné výsledky [17]. Práca sa zaoberá štatistikami tímu na úrovni hráčov a na určenie výsledku používa vyše 300 premenných. Napriek tomu sa v rámci predikcie budúcich zápasov v práci podarilo dosiahnuť iba 53.4%. Dôvodom podľa autora je zložitosť predikcie remízy.

Michael Purucker sa vo svojej práci Neural Network quarterbacking [19] venuje tematike ako jeden z prvých. Jeho práca však bola značne obmedzená počtom premenných a veľkosťou dátového rámca, ktorý využíval. Snažil sa pomocou strojového učenia predikovať výsledky NFL (americká národná rugbyová liga) s dátovým

rámcom, ktorý obsahoval informácie iba z predchádzajúcich zápasov danej sezóny. Ďalším problémom práce bol aj relatívne malý testovací súbor. V ňom sa snažil predikovať naraz iba 14 zápasov. V jednom prípade systém určil správne všetkých 14 zápasov. V mnohých prípadoch však určil iba 8 či 9 zápasov zo 14. Celkovú úspešnosť uvádza 61%.

Na túto prácu v roku 2003 nadviazal Khan [20], ktorý využil viacero premenných a dvihol úspešnosť na 75%. Aj v tejto práci však použil na testovanie iba jeden hrací týždeň, testoval model na týždni 14 a následne týždni 15. V oboch prácach je ťažko odhadnúť, ako úspešný by systém bol z dlhodobého hľadiska.

McCabe a Trevathan sa venovali problematike v ďalšej práci Artificial Intelligence in Sports Prediction [21], kde používali dáta od roku 2002. V tejto práci skúmali štyri rôzne ligy vrátane Premier league a NFL. V práci sa uvádza priemerná úspešnosť pri Premier league 54,6%, čo je približne toľko, koľko sme namerali na nami získaných dátach. Najlepšia úspešnosť je uvádzaná 58,9% za jednu sezónu. Pri americkom futbale je úspešnosť predikcie vyššia, pohybuje sa medzi 63 až 67% v priemere.

Tax a Joustra vo svojej práci [22] využili dáta z trinástich rokov v nemeckom futbale. Ich práca porovnávala viaceré algoritmy. Najlepším výsledkom ktorý dosiahli, bolo 56,1%. Práve v tejto práci sa využili k určeniu výsledku nie len informácie o tímoch, ale aj o stávkovaní samotnom. V našej práci ako informáciu o stávkovaní využívame kurzy iných kancelárií.

Ako môžeme sledovať na predchádzajúcich aplikáciách, určovanie remíz sa javí ako najkritickejší problém pri predikcii výsledku. V tejto práci sa pokúsime zaoberať aj týmto problémom a zvýšiť kvalitu predikcie remíz. Pri futbalových výsledkoch sa pohybujú úspešnosti medzi 53 až 59%, pri rugby, ktoré je podobným športom, sú výsledky lepšie.

4 Spracovanie a analýza dát

Dáta, ktoré v tomto bode máme, sú zatiaľ nepostačujúce na určenie výhry alebo prehry. Aby sme určili, ako si ktorý tím viedol doteraz využijeme dáta z predchádzajúcich zápasov. V prvom rade vytvoríme funkciu na pridanie štyroch stĺpcov, obsahujúcich súčet gólov domáceho a hosťovského tímu po týždeň zápasu. Tak ako nás zaujímajú góly skórované, musíme brať v úvahu aj tie inkasované. Ako ďalšie dáta využijeme body získané v súťaži oboma tímami. Výpočet tohto čísla je jednoduchý, v prípade výhry tím získava 3 body, v prípade remízy jeden bod [10]. Nasledujúcou úpravou dátového rámca je pridanie výsledkov posledných piatich zápasov oboch tímov. K dátovému rámcu sme pridali taktiež výsledné umiestnenie oboch tímov a týždeň v ktorom sa zápas hral. Následne použitím výsledkov posledných piatich zápasov vytvoríme 8 stĺpcov, v ktorých budeme uchovávať informáciu, či tímy v posledných troch a posledných piatich zápasoch mali iba výhry, iba prehry, alebo ani jedno. Nakoniec sa z týchto dát vytvoria stĺpce s rozdielmi skórovaných a inkasovaných gólov, bodmi v súťaži a miestami medzi tohtoročnými a minuloročnými súťažami. S takto pripravenými dátami môžeme pokročiť k trénovaniu modelu.

4.1 Zoznámenie sa s dátami

Pripravené dáta v tejto chvíli majú hlavičku ako na obr. 4 – 1.

```
Stlpce atributov v datovom ramci (42 atributov celkovo):
['result', 'goals1', 'goals2', 'mw', 'HTGS', 'ATGS', 'HTGC', 'ATGC', 'HTP', 'AT
P', 'HM1', 'HM2', 'HM3', 'HM4', 'HM5', 'AM1', 'AM2', 'AM3', 'AM4', 'AM5', 'Home
TeamLP', 'AwayTeamLP', 'HTFormPts', 'ATFormPts', 'HTWinStreak3', 'HTWinStreak
5', 'HTLossStreak3', 'HTLossStreak5', 'ATWinStreak3', 'ATWinStreak5', 'ATLossSt
reak3', 'ATLossStreak5', 'HTGD', 'ATGD', 'DiffPts', 'DiffFormPts', 'DiffLP', 't
eam1_code', 'team2_code', 'w_rate', 'd_rate', 'l_rate']
```

Obrázok 4 – 1: Atribúty v pôvodnom dátovom rámci.

Význam týchto premenných je nasledovný:

- result - výsledok zápasu

- goals1, goals2 - počet gólov oboch tímov
- mw - hrací týždeň
- HTGS, ATGS - skórované góly domáceho tímu a hostí k danému týždňu (H-domáci, a angl. home team, A - hostia, z angl. away team)
- HTGC, ATGC - inkasované góly domáceho tímu a hostí k danému týždňu
- HTP, ATP - body tímov k danému týždňu
- HM1-HM5 a AM1-AM5 - výsledky posledných piatich zápasov oboch tímov
- HomeTeamLP, AwayTeamLP - umiestnenie v lige v poslednom roku pre oba tímy
- HTFormPts, ATFormPts - body vrámci aktuálnej formy pre oba tímy
- HTWinStreak3, HTWinStreak5, ATWinStreak3, ATWinStreak5 - informácia, či sa tímy nachádzajú v sérii troch, prípadne piatich výhier
- HTLossStreak3, HTLossStreak5, ATLossStreak3, ATLossStreak5 - informácia, či sa tímy nachádzajú v sérii troch, prípadne piatich prehier
- HTGD, ATGD - rozdiely gólov medzi tímami
- DiffPts - rozdiel bodov k danému týždňu medzi tímami
- DiffFormPts - rozdiel bodov formy k danému týždňu medzi tímami
- DiffLP - rozdiel umiestnenia v poslednom roku medzi tímami
- team1_code, team2_code - kódové označenia tímov
- w_rate, d_rate, l_rate - kurzy určené na jednotlivé možné výsledky

4.2 Význam premenných

V tejto časti je nutné vysvetliť prečo sme tieto dáta použili a aký majú význam.

Pre začiatok, pôvodný stĺpec result predstavuje výsledný stav zápasu. Obsahuje tri hodnoty: W, D a L, ktoré znamenajú výhru (z angl. win), remízu (z angl. draw) a prehru (z angl. loss). Práve toto je hodnota, ktorú sa budeme snažiť v našom modeli predpovedať. Keď sa pozrieme na celkový počet výhier, prehier a remíz domácich tímov od roku 1998 v tabuľke 4 – 1, je zrejmé, že domáce mužstvo má väčšiu tendenciu vyhrať alebo remízovať, ako prehrať. Takmer polovicu zápasov teda vyhrá domáci tím, zatiaľ čo hostia vyhrávajú len necelých 28% zápasov. Z tohto dôvodu sme rozdelili dáta nie podľa tímov, ale podľa toho, kto je hosť a kto je domáci. V tejto tabuľke môžeme vidieť, že podobné tendencie sa opakujú v každej sezóne. V niektorých sezónach je však zastúpenie remíz vyššie ako prehier.

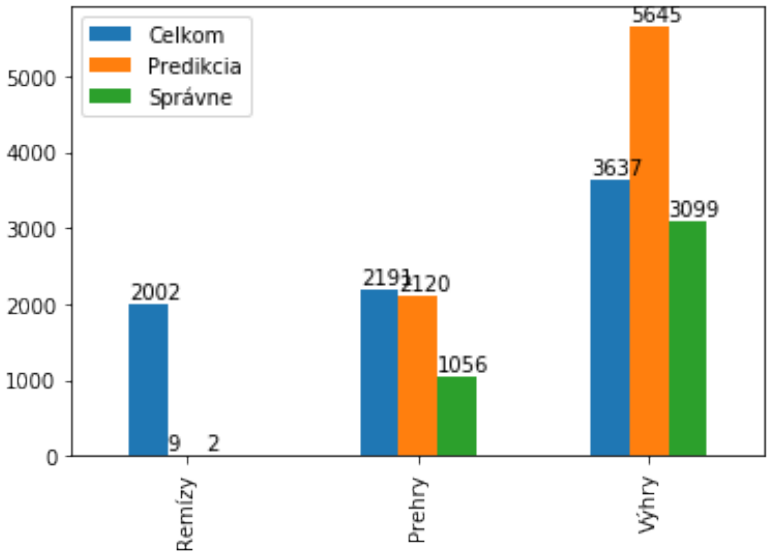
Stĺpce goals1 a goals2 pri trénovaní nevyužijeme, počet gólov je pred začiatkom zápasu neznámy. Potrebné však sú pre výpočet iných hodnôt.

Tabuľka 4 – 1: Počet výhier, prehier a remíz domáceho tímu v jednotlivých sezónach, ich priemer a celkový počet.

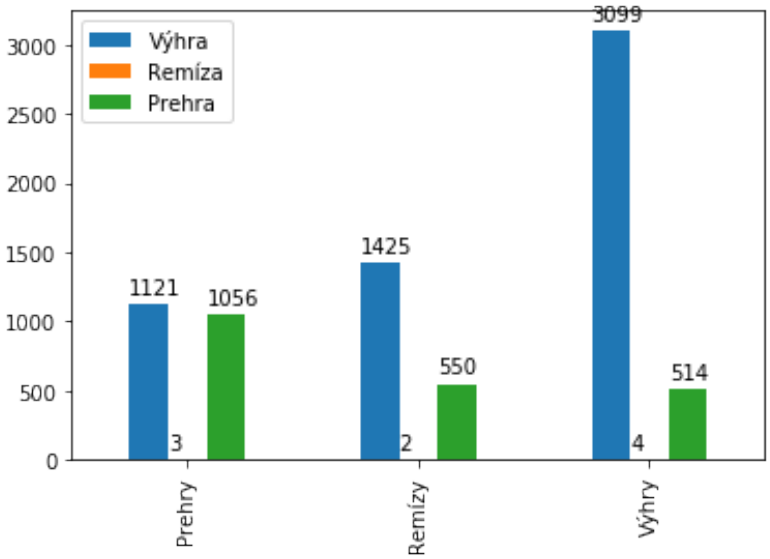
Rok	**Výhry**	**Prehry**	**Remízy**
1998	169	96	115
1999	187	101	92
2000	184	95	101
2001	165	114	101
2002	187	103	90
2003	167	105	108
2004	173	97	110
2005	192	111	77
2006	182	100	98
2007	176	104	100
2008	173	110	97
2009	193	91	96
2010	179	90	111
2011	171	116	93
2012	166	106	108
2013	179	123	78
2014	172	115	93
2015	157	116	107
2016	187	109	84
2017	173	108	99
2018	105	81	44
Priemer	176.6	105.5	97.9
Spolu	3637	2191	2002

Pomerne dôležitými premennými sú premenné w_rate, d_rate a l_rate so stáv-

kovými kurzami. Dáta sa dajú voľne získať na portáli www.betexplorer.com [11]. Na grafe 4 – 2 môžeme vidieť, ako efektívne predpovedali jednotlivé výsledky stávkové spoločnosti v minulosti. Je zjavné, že používaná metóda predikcie je dosť nepresná. Najlepšie výsledky sú pri výhrach, kde z 3637 výhier správne odhadli 3099, čo je zaokrúhlene 85%, no z celkového počtu predikovaných výhier 5645 je to iba necelých 55%. Pri prehrách je to horšie. Zo všetkých prehier bolo správne odhadnutých iba cca 48%, pričom z celkového počtu tipov to predstavuje iba necelých 50%, čo sú veľmi zlé výsledky. Zarážajúce je, že systém takmer vôbec nepredikoval možnosť remízy, pričom pravdepodobnosť remíz sa blíži k pravdepodobnosti prehry domáceho tímu. Percentuálne sú v prípade remíz správne len 2 zápasy z 2002 reálne remízovaných. Zo 7830 celkových zápasov bolo teda správne predpovedaných 4157, čo dáva výslednú úspešnosť iba 53%. Je nutné podotknúť, že v dátovom rámci sme vyberali najnižší kurz na výhru, prehru alebo remízu, aby sme určovali, aký výsledok zápasu bol predpovedaný ako najpravdepodobnejší. Naopak čo sa týka odhadovania najmenej pravdepodobného výsledku, úspešnosť presahuje 78%. V tejto práci sa budeme snažiť vytvoriť algoritmus, ktorý bude schopný odhadovať výsledok zápasu lepšie, ako analyzované. Zaujímavou môže byť v tomto prípade aj analýza toho, aký výsledok systém očakával a aký reálne bol. Toto porovnanie máme graficky znázornené na grafe 4 – 3. Je vidno, že systém má pomerne dobré výsledky pri predikcii výhry, no na remízu a prehru sa v jeho prípade nedá spoľahnúť.



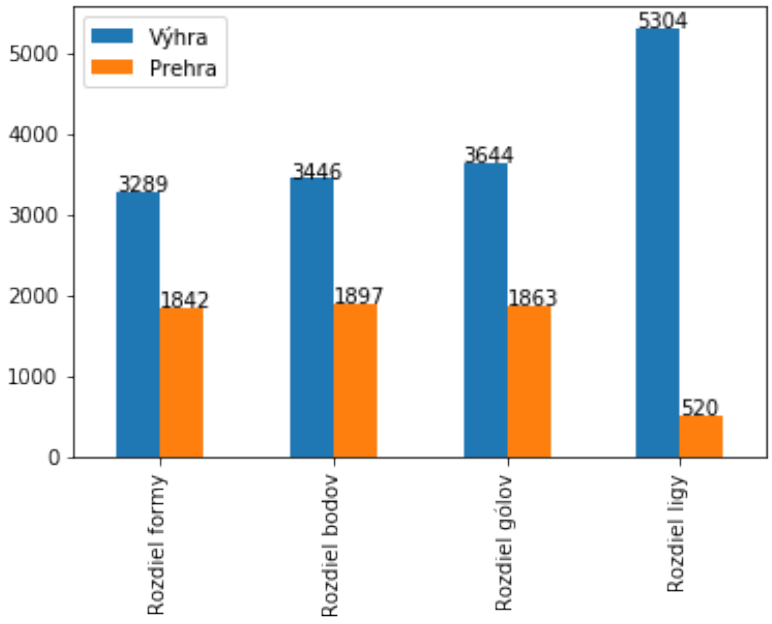
Obrázok 4 – 2: Porovnanie celkového počtu jednotlivých výsledkov a celkovej a správnej predikcie týchto výsledkov v minulosti.



Obrázok 4 – 3: Porovnanie počtu predikcií každej možnosti (os y) výsledku pri skutočnom výsledku (os x).

Ďalšími dôležitými premennými sú HTGC, HTGS, ATGS a ATGC. Pri mnohých premenných budeme používať miesto číselného označenia označenie HT ako Home

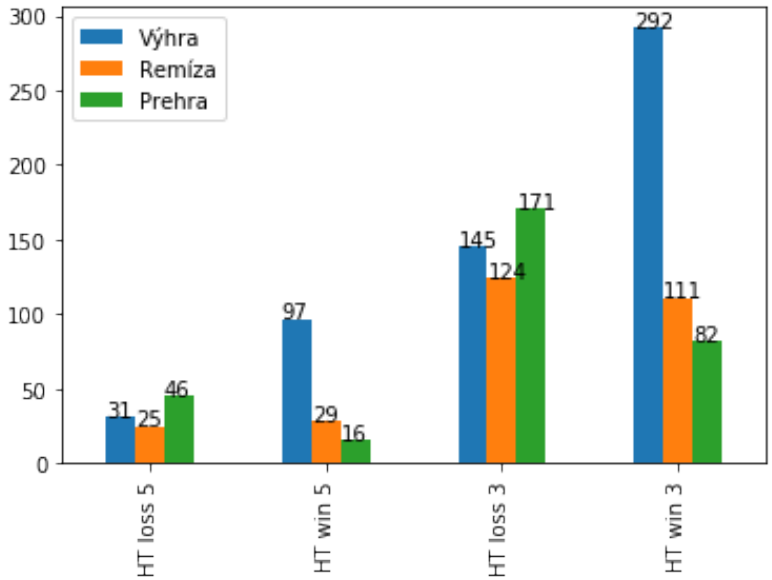
Team - domáci, a AT - Away Team, čiže hostia. Jedná sa o skórované a inkasované góly oboch tímov v súťaži do uskutočnenia zápasu. V prvých týždňoch súťaže tieto premenné síce veľmi nezavážia, no v závere súťaže by jasne mali odzrkadľovať, ktorý tím je silnejší a má väčšiu šancu vyhrať. Premenné ATP a HTP podobne odzrkadľujú, ako si tím vedie v súťaži z bodového hľadiska vzhľadom na aktuálny súťažný týždeň. Vypočítajú sa ako súčet výsledkov doterajších zápasov, kde za výhru sa prideľujú 3 body, za remízu jeden a za prehru 0 bodov. To celé sa predelí aktuálnym týždňom. Premenné HomeTeamLP a AwayTeamLP obsahujú umiestnenia z posledného roka. Do tretice máme premenné HTFormPts a ATFormPts, ktoré definujú body získané v aktuálnej forme tímu, za posledných 5 zápasov. Všetky tieto premenné sme od seba aj odčítali, aby sme získali rozdiely medzi inkasovanými a strelenými gólmi, ligovým umiestnením, bodmi formy aj bodmi vzhľadom na hrací týždeň. Na ďalšom grafe $4-4$ môžeme vidieť, že vo väčšine prípadov má naozaj vyššiu pravdepodobnosť výhry tím, ktorý má čísla lepšie. Remízy v tomto grafe nie sú, pretože pri remíze sú hodnotené rovnako oba tímy. Umiestnenie v lige sa javí ako najsilnejší ukazovateľ pravdepodobnosti výhry.



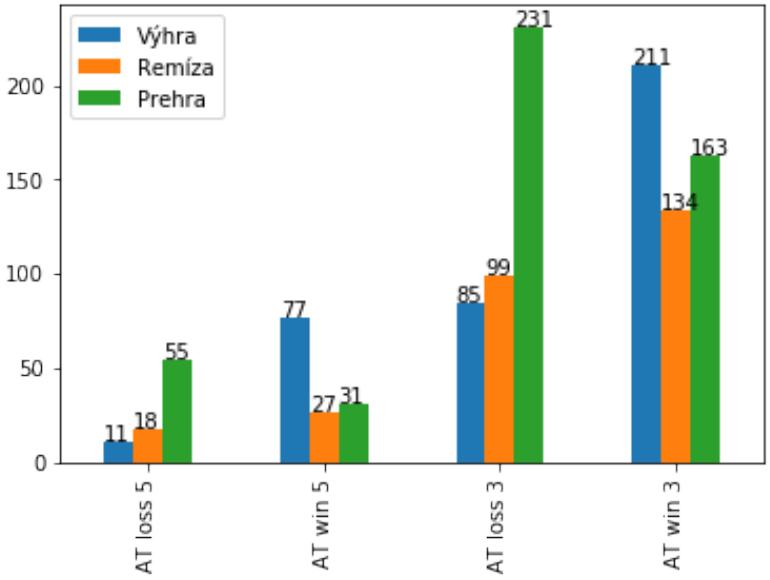
Obrázok 4 – 4: Počty výhier a prehier tímu s viac bodmi vo forme a týždni, tímu s lepším gólovým skóre, a tímu s lepším ligovým umiestnením.

Čo sa týka súčasnej formy tímu, máme tu 5 premenných reprezentujúcich posledných 5 zápasov oboch tímov - HM1 až HM5 a AM1 až AM5. Body súčasnej formy sa vypočítajú práve z týchto premenných. Z výsledkov posledných zápasov si vytvoríme aj premenné určujúce, či sú tímy v sérii víťazstiev alebo prehier počas posledných troch a posledných piatich zápasov. Porovnanie toho, ako si viedli domáci tím a hosťujúci tím, keď boli v sérii výhier alebo prehier, máme na grafoch 4 – 5 a 4 – 6. Z grafov sa dá vidieť veľmi zaujímavé správanie tímov, ktoré nie je pre oba tímy rovnaké. Ak sa jedná o tím domáci, takmer vôbec nezáleží, či sú v sérii prehier. Pravdepodobnosť prehry je iba mierne vyššia, ako výhry. Ak je tento tím v sérii výhier, naopak, je veľmi vysoká pravdepodobnosť, že vyhrá. Pri piatich výhrach v rade skončí víťazne viac než 63% zápasov, pri troch sa to pohybuje okolo 60%. U hostí je to mierne inak, pri piatich výhrach je to približne 57%, pri troch však iba cca. 41,5%, čo je však stále viac, ako je priemer. Naopak, pri prehrách je to jednoznačné, pri piatich v rade prehráva hosťujúci tím vyše 65% zápasov, pri troch je to takmer

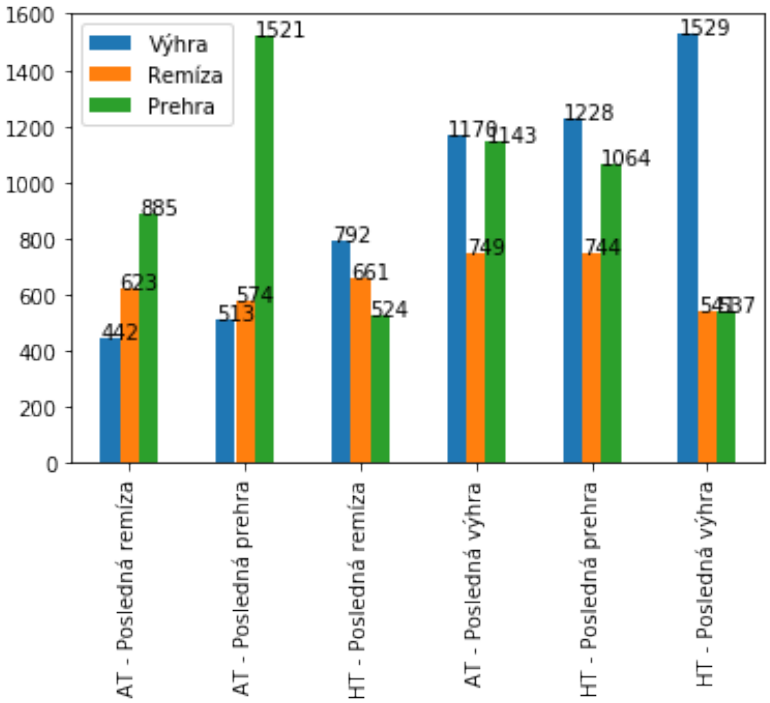
56%. Je zjavné, že aktuálna forma má veľký vplyv na výsledok zápasu. Na grafe 4 – 7 môžeme ešte vidieť porovnanie výsledku zápasu podľa predchádzajúceho zápasu. Je pomerne dobre viditeľné, že v prípade hostí zápas často skončí ich prehrou, ak ich predchádzajúci zápas skončil prehrou, alebo remízou. Ak však ich predchádzajúci zápas skončil výhrou, aj tak je pravdepodobnosť ďalšej výhry len 38%. U domáceho tímu je to mierne inak, ak bol predchádzajúci výsledok výhra, alebo remíza, je viac pravdepodobné, že tím vyhrá. Pri predchádzajúcej výhre je to až takmer 59%, čo je približne taká pravdepodobnosť, ako že hostia, ktorí predchádzajúci zápas prehrali, prehrajú aj tento. V prípade prehry posledného zápasu domácim tímom sa pravdepodobnosť prehry v ďalšom zápase zvyšuje iba mierne nad 40%, čo je podobné ako výhra v prípade výhry predchádzajúceho zápasu u hostí.



Obrázok 4 – 5: Počty výhier, prehier a remíz domáceho tímu ak bol práve v sérii výhier alebo prehier.



Obrázok 4 – 6: Počty výhier, prehier a remíz hosťovského tímu ak bol práve v sérii výhier alebo prehier.



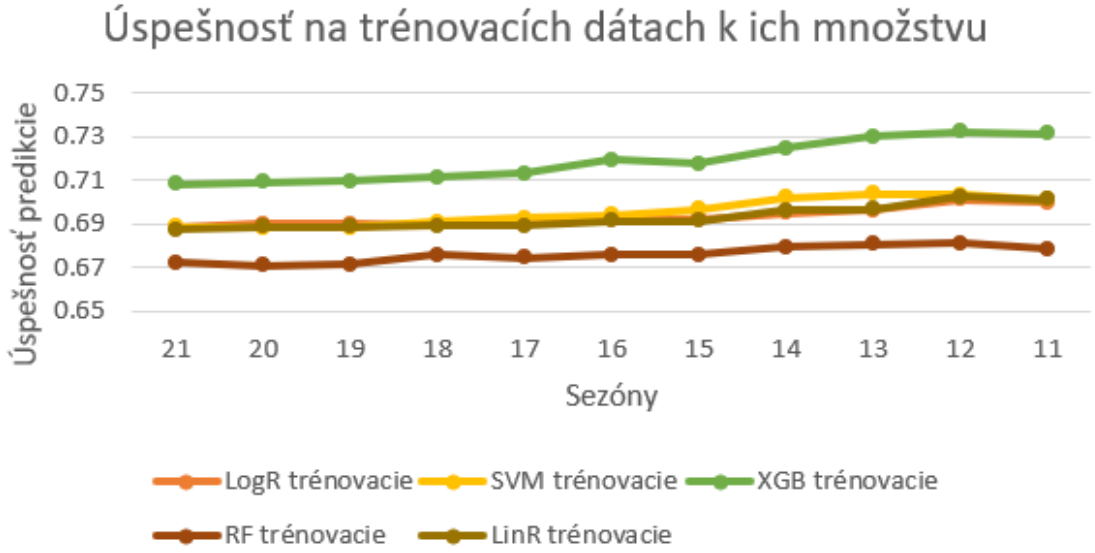
Obrázok 4 – 7: Počty výhier, prehier a remíz oboch tímov podľa ich predošlého zápasu.

5 Prvé trénovanie modelov

Čo sa týka prvého trénovania modelov, je nutné podotknúť, že ešte pred trénovaním každého modelu je potrebné nastaviť správne parametre. Pre zvýšenie prehľadnosti práce sme sa rozhodli popísať ako sme parametre nastavovali a ako sa zvýšila úspešnosť vďaka nim až v kapitole 6. Vyskúšali sme úspešnosť učenia pri lineárnej a logistickej regresii, gradient boostingu, algoritme SVM a náhodnom lese. Pre zjednodušenie problému sme v úvode zjednodušili výsledky len na dve možnosti, výhra domácich alebo niečo iné. Binárny problém je totiž značne jednoduchší ako ternárny, a preto sme očakávali pomerne dobré výsledky. Ďalšia vec, ktorú si musíme uvedomiť, je, aké obdobie chceme predpovedať. Z toho odvodíme aj množstvo testovacích dát. Pri stávkovaní ide o krátke časové obdobie. Nemá zmysel predpovedať výsledok zápasu, ktorý sa odohrá o 2 mesiace, pretože za tak dlhý čas sa môže stať veľké množstvo vecí. Malý počet zápasov v testovacom súbore však môže mať za následok skreslené výsledky. V tejto chvíli si musíme uvedomiť aj fakt, aké množstvo dát máme k dispozícii. Za bežných okolností sa odporúča rozdeliť dáta na trénovacie a testovacie dáta v rozsahu 90% trénovacie a 10% testovacie dáta [12]. Naše dáta však obsahujú informácie ani nie o 8000 zápasoch, takže odobrať 700-800 zápasov z trénovacích dát môže mať značný vplyv na kvalitu trénovania modelu. Takéto množstvo dát predstavuje až dve celé série, čo taktiež presahuje nutné množstvo testovacích dát. To predstavuje taký podiel, aby sa vo vzorke nachádzali dáta reprezentujúce celú dátovú sadu [18]. Najmä z dôvodu, že niektoré tímy sa vyskytli v dátovej sade len niekoľko krát, nie je možné uvažovať nad generalizáciou dát podľa tímov. Čo sa však pravidelne opakuje, sú týždne v sezónach. Tých je 38, takže na to, aby sme v testovacích dátach mali reprezentatívnu vzorku dát, musíme si zvoliť viac ako 38 zápasov pre testovanie. Vybrali sme si množstvo 70 vzorových zápasov a 380, čo sa rovná jednej celej sérii.

5.1 Úspešnosť na trénovacích dátach

Pri trénovaní modelu sa model trénuje najprv na trénovacích dátach a určuje sa jeho úspešnosť, až následne sa odskúša na testovacích dátach. Vzhľadom na charakter našich dát je zrejmé, že na testovacom súbore nevieme dosiahnuť vysokú úspešnosť. Jednak je to preto, že rôzne sezóny sa od seba odlišovali ako participantmi, tak aj ich výkonmi. Analýzou vývoja tímov môžeme dospieť k záveru, že ich forma sa môže v čase výrazne meniť. Príkladom môže byť napr. tím Manchester City, ktorý bol v minulosti priemerný až slabý, no dnes patrí medzi najlepšie tímy. Na grafe 5 – 1 je vidieť, že ani postupným zmenšovaním dátového rámca sa nám nepodarilo úspešnosť predikcie na trénovacích dátach markantne zmeniť. Síce sa pri zmenšujúcom sa počte sezón mierne zvyšuje úspešnosť, je to však len minimálny rozdiel. Spôsobené to okrem iného môže byť aj zápasmi, v ktorých výsledok ovplyvnili nečakané udalosti ako napr. zranenie dôležitého hráča alebo nepriaznivé počasie. Úspešnosť je približne rovnaká pri jednotlivých algoritmoch nezávisle od veľkosti testovacej vzorky. Najlepšie výsledky má v tomto prípade algoritmus XGBoost.



Obrázok 5 – 1: Úspešnosť predikcie na trénovacích dátach pri ich postupnom znižovaní.

5.2 Úspešnosť predikcie na testovacích dátach

Zaujímavejšie výsledky sme dosiahli na testovacích dátach. Prvotné výsledky predstavovali pri všetkých algoritmoch hodnoty veľmi nízke, niekde až pod 50%. Vzhľadom na fakt, že sme sa snažili určiť iba to, či tím vyhrá, alebo nie, je to výsledok veľmi slabý. Preto sme sa rozhodli bližšie preskúmať, nakoľko je množstvo sezón na ktorých sa algoritmy učia dôležité. Napriek predpokladu, že na kvalitné natrénovanie je nutné veľké množstvo dát sa ukázalo, že pri nižších počtoch sezón sme pri všetkých algoritmoch dostali lepšie výsledky.

V tejto chvíli sme sa rozhodli taktiež skúmať aj to, ako a či vôbec vplýva na predikciu informácia o dátume konania. Z dátumu sme extrahovali deň v týždni a mesiac kedy sa zápas konal.

V tabuľkách 5 – 1, 5 – 2, 5 – 3 a 5 – 4 máme grafické porovnanie úspešnosti predikcie na testovacích dátach o veľkosti 70 zápasov a 380 zápasov. Tabuľky 5 – 2 a 5 – 4 obsahujú hodnoty namerané pri využití dátumových premenných. Počet skúmaných sezón je 21 až 12, pri nižšom počte nevieme vytvoriť relevantný testovací súbor. Pri rôzne veľkom testovacom súbore zaznamenávame rôzne výsledky. Výsledky sú však pri testovacom súbore 380 zápasov v oboch prípadoch veľmi podobné, pohybujú sa v rozsahu 67-69% u všetkých algoritmov. V prípade testovacieho súboru s veľkosťou 70 zápasov majú mierne horšie výsledky algoritmy XGB a SVM. Pri oboch testovacích súboroch je využitie dátumu otázne, keďže v niektorých prípadoch boli lepšie výsledky zaznamenané s ním, no v iných zasa bez neho.

Tabuľka 5 – 1: Porovnanie úspešnosti predikcie na testovacích dátach vzhľadom na počet sezón v trénovacích dátach, bez informácií o dátume, pri vzorke 70 zápasov.

Počet sezón	**Lineárna r.**	**Logistická r.**	**SVM**	**XGB**	**Náhodný les**
21	0.6429	0.6429	0.6	0.6143	0.6857
20	0.6571	0.6571	0.6	0.6143	0.6429
19	0.6714	0.6714	0.6	0.6429	0.6571
18	0.6571	0.6857	0.6	0.6143	0.6571
17	0.6857	0.7	0.6143	0.6571	0.6714
16	0.6571	0.6714	0.6286	0.6286	0.6571
15	0.6571	0.6714	0.6429	0.6429	0.6429
14	0.6857	0.6857	0.6143	0.6429	0.6571
13	0.6857	0.6857	0.6286	0.6286	0.6714
12	0.6571	0.6714	0.6143	0.6571	0.6143

Tabuľka 5 – 2: Porovnanie úspešnosti predikcie na testovacích dátach vzhľadom na počet sezón v trénovacích dátach, s informáciou o dátume, pri vzorke 70 zápasov.

Počet sezón	**Lineárna r.**	**Logistická r.**	**SVM**	**XGB**	**Náhodný les**
21	0.6571	0.6571	0.6	0.6286	0.6571
20	0.6714	0.6714	0.6	0.6429	0.6714
19	0.6571	0.6571	0.6143	0.6429	0.6571
18	0.6714	0.6714	0.6143	0.6143	0.6429
17	0.6571	0.6571	0.6286	0.6286	0.6429
16	0.6571	0.6571	0.6286	0.6286	0.6429
15	0.6714	0.6714	0.6429	0.6571	0.6429
14	0.6714	0.6714	0.6143	0.6286	0.6714
13	0.6714	0.6714	0.6429	0.6571	0.6571
12	0.6429	0.6429	0.6143	0.6429	0.6429

Tabuľka 5 – 3: Porovnanie úspešnosti predikcie na testovacích dátach vzhľadom na počet sezón v trénovacích dátach, bez informácií o dátume, pri vzorke 380 zápasov.

Počet sezón	**Lineárna r.**	**Logistická r.**	**SVM**	**XGB**	**Náhodný les**
21	0.6711	0.6921	0.6921	0.6868	0.6895
20	0.6789	0.6921	0.6868	0.6895	0.6895
19	0.6711	0.6947	0.6868	0.6763	0.7
18	0.6711	0.6947	0.6868	0.6868	0.6974
17	0.6711	0.6974	0.6974	0.6842	0.6947
16	0.6711	0.6921	0.6868	0.6947	0.6974
15	0.6605	0.6947	0.6868	0.6711	0.6868
14	0.6605	0.6842	0.6974	0.6684	0.6921
13	0.6526	0.6789	0.6974	0.6868	0.6974
12	0.6737	0.6789	0.6947	0.6895	0.6947

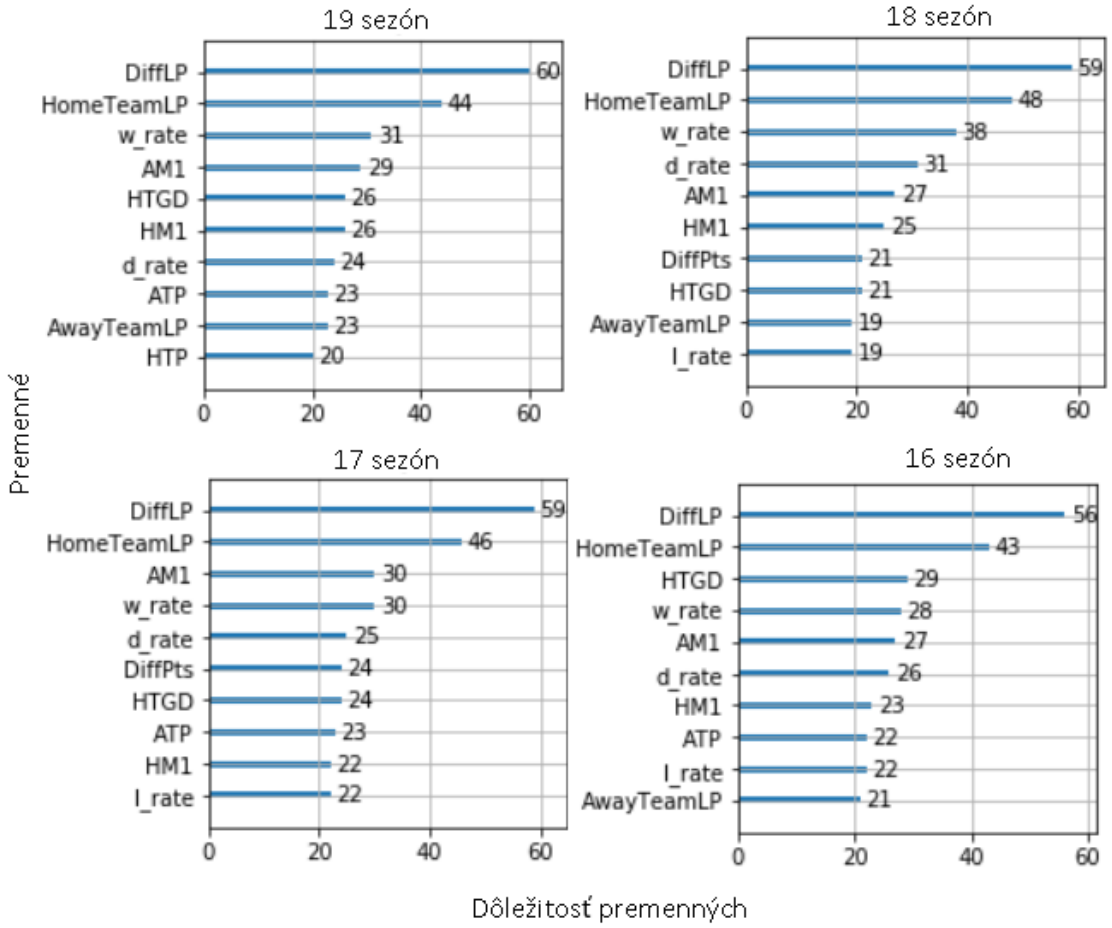
Tabuľka 5 – 4: Porovnanie úspešnosti predikcie na testovacích dátach vzhľadom na počet sezón v trénovacích dátach, s informáciou o dátume, pri vzorke 380 zápasov.

Počet sezón	**Lineárna r.**	**Logistická r.**	**SVM**	**XGB**	**Náhodný les**
21	0.6842	0.6974	0.6868	0.6816	0.6974
20	0.6921	0.6881	0.6816	0.6789	0.6947
19	0.6868	0.6974	0.6895	0.6763	0.7
18	0.6763	0.6947	0.6816	0.6868	0.6921
17	0.6868	0.6974	0.6816	0.6974	0.7
16	0.6947	0.7026	0.6789	0.6789	0.7
15	0.6632	0.6921	0.6895	0.6816	0.6895
14	0.6711	0.6868	0.6895	0.6737	0.6974
13	0.6605	0.6789	0.6921	0.6868	0.6895
12	0.6684	0.6763	0.6921	0.6921	0.6921

Keď sa pozrieme na dôležitosť jednotlivých premenných pomocou funkcie plot_importances knižnice XGBoost na grafoch 5 – 2, môžeme vidieť, že opakovane je považované ako jedno z najdôležitejších faktorov ligové umiestnenie, konkrétne rozdiel umiestnení jednotlivých tímov. Vybrali sme si počet sezón 19-16, kde sú badateľné určité rozdiely v kvalite predikcie. V tabuľke 4 – 1, kde máme počty víťazstiev, prehier a remíz tímov s lepším umiestnením v lige, môžeme pozorovať iný jav. Ako sme mohli vidieť v predchádzajúcej kapitole, rozdiel ligy je veľmi dôležitým ukazovateľom výsledku. Predpokladáme, že sezóny, v ktorých je vyšší pomer prehier a remíz k výhram majú vplyv na zhoršenú kvalitu predikcie, a ich odobratie môže mať pozitívny vplyv na predikciu. Problémom však je, že odoberať tieto sezóny sa nedá selektívne, pretože by sa narušila časová postupnosť v dátach.

Tabuľka 5 – 5: Početnosť výhier, prehier a remíz tímu s lepšou pozíciou v predchádzajúcom roku za prvých 10 sezón.

Rok	**Výhry**	**Prehry**	**Remízy**
1998	189	76	115
1999	204	84	92
2000	185	94	101
2001	201	78	101
2002	205	85	90
2003	191	79	110
2004	194	76	110
2005	229	74	77
2006	178	103	99
2007	217	104	59



Obrázok 5 – 2: 10 najdôležitejších premenných pre algoritmus gradient boosting, pre počet 16 až 19 sezón v trénovacích dátach.

Celkovo však pri týchto dátach môžeme pozorovať, že kvalita predikcie je pomerne nízka, dosahuje pri väčšom počte testovacích dát úspešnosť najviac okolo 70%. Preto sme sa rozhodli, že dátovú sadu rozšírime o množstvo charakteristík opisujúcich tím z dlhodobého hľadiska. Charakteristiky sa týkajú počtu gólov, odohraných zápasov, výhier, prehier a podobne v rámci celého pôsobenia v lige. V nasledujúcej kapitole si ich priblížime.

5.3 Rozšírenie o dlhodobé štatistiky

Postupne sme pridali do dátového rámca nové atribúty. Najprv sme získali celkové hodnoty na stránke www.premierleague.com a následne sme z nich vytvorili priemerné a agregované hodnoty. Jedná sa o hodnoty celkového počtu odohraných sezón, hier, výhier, prehier, remíz, odohraných zápasov, inkasovaných a strelených gólov oboch tímov.

Z týchto hodnôt sme následne vypočítali priemerný počet skórovaných a inkasovaných gólov za hru, rozdiel medzi skórovanými a inkasovanými gólmi, priemerné umiestnenie počas celého účinkovania v súťaži. Ďalej sme pridali hodnotu WLRatio a WDLRatio, ktoré určujú pomer výhier tímu k prehrám a pomer výhier a remíz tímu k jeho prehrám.

Význam jednotlivých pridaných premenných je nasledovný:

- SeasonsPlayedHT, SeasonsPlayedAT - počet sezón ktoré boli tímy v súťaži
- LosesTotalHT, LosesTotalAT - celkový počet prehier oboch tímov
- WinsTotalHT, WinsTotalAT - celkový počet výhier oboch tímov
- DrawssTotalHT, DrawssTotalAT - celkový počet remíz oboch tímov
- WLRatioHT, WLRatioAT - pomer výhier k prehrám
- WDLRatioHT, WDLRatioAT - pomer výhier a remíz k prehrám
- GoalsScorededTotalHT, GoalsScorededTotalAT - celkový počet skórovaných gólov tímov
- GoalsConcededTotalHT, GoalsConcededTotalAT - celkový počet inkasovaných gólov tímov
- GPGRatioHT, GPGRatioAT - priemerný počet gólov na zápas oboch tímov
- GCPGRatioHT, GCPGRatioAT - priemerný počet inkasovaných gólov na zápas oboch tímov

- SCDiffHT, SCDiffAT - rozdiel medzi skórovanými a inkasovanými gólmi oboch tímov
- AvgStandingHT, AvgStandingAT - priemerné postavenie v ligách počas účinkovania tímov v súťaži
- GamesPlayedHT, GamesPlayedAT - celkový počet hier ktoré tímy odohrali

V tabuľkách 5 – 6,5 – 7,5 – 8 a 5 – 9 môžeme vidieť, ako sa nám oproti pôvodnému dátovému rámcu zmenila úspešnosť predikcie. Opäť sme vykonávali testy bez dátumových premenných a s nimi. Pre zjednodušenie sme robili testy len na dátach s počtom sezón 21 až 14. Ostatné sme vylúčili pre slabé dosiahnuté výsledky. Na reprezentáciu zmeny sme znova sledovali zmeny na testovacích dátach o veľkosti 70 a 380 zápasov.

Tabuľka 5 – 6: Porovnanie úspešnosti predikcie na testovacích dátach vzhľadom na počet sezón v trénovacích dátach , bez informácií o dátume, pri vzorke 70 zápasov, s novou dátovou sadou.

Počet sezón	**Lineárna r.**	**Logistická r.**	**SVM**	**XGB**	**Náhodný les**
21	0.6286	0.6286	0.6429	0.6714	0.6571
20	0.6429	0.6571	0.6429	0.6429	0.6286
19	0.6571	0.6571	0.6571	0.6286	0.6429
18	0.6571	0.6571	0.6286	0.6429	0.6571
17	0.6714	0.6714	0.6286	0.6429	0.6429
16	0.6714	0.6571	0.6429	0.6286	0.6571
15	0.6714	0.6571	0.6571	0.6286	0.6429
14	0.6429	0.6714	0.6571	0.6429	0.6429

Tabuľka 5 – 7: Porovnanie úspešnosti predikcie na testovacích dátach vzhľadom na počet sezón v trénovacích dátach, s informáciou o dátume, pri vzorke 70 zápasov, s novou dátovou sadou.

Počet sezón	Lineárna r.	Logistická r.	SVM	XGB	Náhodný les
21	0.6286	0.6286	0.6429	0.6714	0.6286
20	0.6429	0.6571	0.6429	0.6429	0.6571
19	0.6429	0.6571	0.6571	0.6429	0.6571
18	0.6714	0.6714	0.6286	0.6143	0.6571
17	0.6714	0.6714	0.6286	0.6286	0.6571
16	0.6857	0.6714	0.6429	0.6286	0.6429
15	0.6571	0.6571	0.6571	0.6429	0.6286
14	0.6571	0.6571	0.6571	0.6286	0.6143

Tabuľka 5 – 8: Porovnanie úspešnosti predikcie na testovacích dátach vzhľadom na počet sezón v trénovacích dátach, bez informácií o dátume, pri vzorke 380 zápasov, s novou dátovou sadou.

Počet sezón	Lineárna r.	Logistická r.	SVM	XGB	Náhodný les
21	0.6658	0.6842	0.6026	0.6816	0.6868
20	0.6579	0.6921	0.6026	0.6816	0.6842
19	0.6605	0.6921	0.6237	0.6868	0.6895
18	0.6658	0.6921	0.6316	0.6684	0.6763
17	0.6763	0.6921	0.6263	0.6737	0.6816
16	0.6789	0.6974	0.6211	0.6737	0.6842
15	0.6605	0.6974	0.6211	0.6711	0.6789
14	0.6526	0.6789	0.6158	0.6737	0.6789

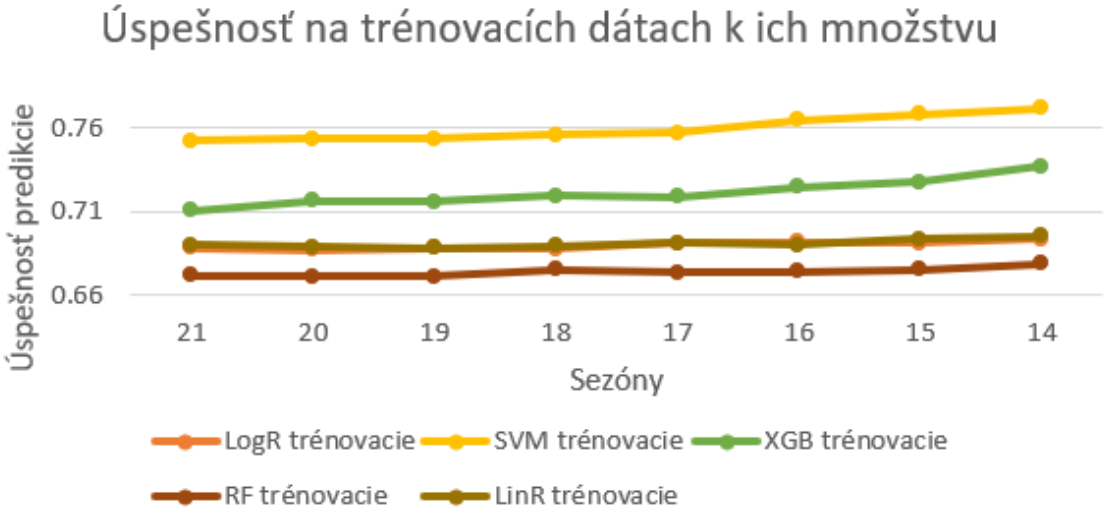
Tabuľka 5 – 9: Porovnanie úspešnosti predikcie na testovacích dátach vzhľadom na počet sezón v trénovacích dátach, s informáciou o dátume, pri vzorke 380 zápasov, s novou dátovou sadou.

Počet sezón	Lineárna r.	Logistická r.	SVM	XGB	Náhodný les
21	0.6658	0.6868	0.6026	0.6605	0.6816
20	0.6579	0.6921	0.6053	0.6921	0.6868
19	0.6658	0.6895	0.6237	0.6868	0.7026
18	0.6737	0.6868	0.6342	0.6632	0.6737
17	0.6711	0.6974	0.6263	0.6737	0.6895
16	0.6868	0.7079	0.6184	0.6684	0.6816
15	0.6632	0.6947	0.6211	0.6763	0.6895
14	0.6737	0.6868	0.6158	0.6737	0.6895

Čo sa týka úspešnosti na trénovacích dátach, ostáva aj v tomto prípade v podobných hodnotách pre všetky algoritmy. Pri testovacích dátach však pozorujeme mierne zmeny vo viacerých aspektoch. V tomto prípade sa javí použitie dátumu kontraproduktívne, vo väčšine prípadov sme s dátumom zaznamenali o niečo nižšiu úspešnosť ako bez neho. Najslabšie výsledky má algoritmus SVM a nasleduje ho algoritmus lineárnej regresie. Pri 70 testovacích zápasoch mal najhoršie výsledky algoritmus XGB. Určenie najvhodnejšieho počtu sezón je taktiež zložité, každý algoritmus sa správa odlišne. Algoritmy náhodný les a XGB sa javia ako úspešnejšie pri 19 a 20 sezónach, zatiaľ čo regresné algoritmy dosahujú najlepšie výsledky s počtami sezón 16-19. V priemere najlepšie výsledky má algoritmus náhodného lesa, logistická regresia má porovnateľné výsledky. Celkovo sa však úspešnosť algoritmov na dátových sadách mierne znížila v porovnaní s predchádzajúcou dátovou sadou.

Ako príčina sa javí spôsob zadefinovania premenných, kde sme ku každému zápasu pridali informácie o súčasnom stave tímu. Vzhľadom na fakt, že v dobe konania zápasov tieto informácie neboli známe, rozhodli sme sa dátový rámec upraviť.

Všetky premenné ktoré sme dopĺňali v predchádzajúcej kapitole sme predefinovali tak, aby obsahovali iba hodnoty platné v dobe konania zápasu. Preto sa v novom dátovom rámci už nemôže stať, že by sa v danom zápase javil niektorý tím silnejší, rsp. slabší, ako reálne v dobe jeho konania bol. Správnosť tohoto kroku môžeme sledovať na grafe 5 – 3, kde je vyobrazená úspešnosť na trénovacej vzorke na nových opravených dátach.



Obrázok 5 – 3: Porovnanie úspešnosti predikcie na trénovacích dátach vzhľadom na počet sezón, bez informácií o dátume a s nimi, v rozšírenom dátovom rámci s agregovanými hodnotami.

Je zrejmé, že úspešnosť predikcie je mierne vyššia, a to najmä u algoritmu SVM. Úspešnosť sa tu dvihla z cca. 69% až na 76%. Ostatné algoritmy sa zachovali približne na pôvodných hodnotách.

Napriek zvýšenej úspešnosti na trénovacej vzorke je však výsledok na testovacích dátach u algoritmu SVM nízky. Vidieť to môžeme na ďalšom porovnaní v tabuľkách 5 – 11, 5 – 12, 5 – 10 a 5 – 13. Zdá sa preto, že tento algoritmus neposkytuje vhodnú metódu pre predikciu výsledkov zápasu. Regresné algoritmy a algoritmus náhodného lesa dosahujú v najlepších prípadoch viac 70%, čo sú doposiaľ najlepšie dosiahnuté výsledky. Ukazuje sa, že použitie dátumu v dátovom rámci nemá pozitívny vplyv na

úspešnosť predikcie v takmer žiadnom prípade. Tento jav pozorujeme ako pri testovacom súbore s počtom 380 zápasov, tak aj pri 70 zápasoch. Preto sme sa rozhodli ďalej nezaoberať vplyvom času konania zápasu na výsledok. Vo väčšine prípadov sa javí ako najlepší počet sezón 18 19 pre regresné algoritmy, kde zaznamenávame pomerne vysokú úspešnosť oproti iným počtom sezón. V prípade náhodného lesa sme najvyššiu úspešnosť dosiahli pri pošte 15 sezón.

Tabuľka 5 – 10: Porovnanie úspešnosti predikcie na testovacích dátach vzhľadom na počet sezón v trénovacích dátach , bez informácií o dátume, pri vzorke 70 zápasov, s upravenou novou dátovou sadou.

Počet sezón	**Lineárna r.**	**Logistická r.**	**SVM**	**XGB**	**Náhodný strom**
21	0.6571	0.6429	0.5857	0.6429	0.6571
20	0.6714	0.6714	0.6	0.6571	0.6714
19	0.6857	0.6714	0.6	0.6429	0.6429
18	0.6857	0.6857	0.6	0.6286	0.6429
17	0.6571	0.7	0.5857	0.6429	0.6714
16	0.6857	0.7	0.6143	0.6143	0.6143
15	0.6857	0.7	0.6143	0.6571	0.6714
14	0.6714	0.6714	0.6143	0.6286	0.6714

Tabuľka 5 – 11: Porovnanie úspešnosti predikcie na testovacích dátach vzhľadom na počet sezón v trénovacích dátach, s informáciou o dátume, pri vzorke 70 zápasov, s upravenou novou dátovou sadou.

Počet sezón	Lineárna r.	Logistická r.	SVM	XGB	Náhodný strom
21	0.6429	0.6429	0.5857	0.6429	0.6429
20	0.6714	0.6571	0.5857	0.6571	0.6571
19	0.6857	0.6714	0.6	0.6571	0.6571
18	0.6857	0.6857	0.6	0.6286	0.6714
17	0.6857	0.6857	0.5857	0.6286	0.6429
16	0.6857	0.6857	0.6143	0.6143	0.6429
15	0.6714	0.6857	0.6143	0.6571	0.6286
14	0.6714	0.6857	0.6143	0.6429	0.6571

Tabuľka 5 – 12: Porovnanie úspešnosti predikcie na testovacích dátach vzhľadom na počet sezón v trénovacích dátach, bez informácií o dátume, pri vzorke 380 zápasov, s upravenou novou dátovou sadou.

Počet sezón	Lineárna r.	Logistická r.	SVM	XGB	Náhodný strom
21	0.6868	0.6895	0.6316	0.6842	0.7026
20	0.6868	0.7	0.6342	0.6974	0.6947
19	0.7053	0.7079	0.6289	0.6895	0.6868
18	0.7105	0.7132	0.6316	0.6842	0.6974
17	0.7053	0.7	0.6316	0.6868	0.7026
16	0.7	0.6947	0.6342	0.6763	0.7026
15	0.6921	0.6974	0.6395	0.6789	0.7105
14	0.6895	0.6974	0.6395	0.6763	0.7026

Tabuľka 5 – 13: Porovnanie úspešnosti predikcie na testovacích dátach vzhľadom na počet sezón v trénovacích dátach, s informáciou o dátume, pri vzorke 380 zápasov, s upravenou novou dátovou sadou.

Počet sezón	**Lineárna r.**	**Logistická r.**	**SVM**	**XGB**	**Náhodný strom**
21	0.6868	0.6868	0.6316	0.6868	0.6974
20	0.6895	0.7	0.6342	0.6974	0.6974
19	0.7079	0.7053	0.6316	0.6921	0.7053
18	0.7079	0.7053	0.6316	0.6789	0.7
17	0.6947	0.6974	0.6316	0.6737	0.6921
16	0.7	0.7026	0.6342	0.6789	0.7026
15	0.6921	0.6947	0.6395	0.6842	0.6947
14	0.6921	0.7079	0.6395	0.6816	0.6921

Oproti pôvodnému dátovému súboru sme zaznamenali zvýšenie úspešnosti u algoritmov XGB a regresných algoritmov okolo troch percent. Toto číslo síce nie je vysoké, ale môže byť kritické.

6 Úprava parametrov jednotlivých algoritmov

Dôležitým mechanizmom zlepšovania výsledkov jednotlivých algoritmov je hľadanie najvhodnejšej kombinácie parametrov. O význame jednotlivých parametrov sme niečo načrtli pri opise algoritmov strojového učenia. Viac sa o nich dozvieme v tejto kapitole. Každý parameter má v danom algoritme svoju špecifickú úlohu. Ak je jeho hodnota nesprávna, môže to spôsobiť nesprávne natrénovanie a následný zlý výstup modelu. Vo väčšine prípadov možno na zisťovanie parametrov použiť mechanizmus zvaný grid search, v preklade by sme mohli povedať že sa jedná o mriežkové vyhľadávanie. Mechanizmus tohoto hľadania parametrov je založený na postupnom skúšaní všetkých kombinácií parametrov, ktoré zadefinujeme. Vzhľadom na časovú zložitosť tohto postupu je nutné zadefinovať si určitú množinu hodnôt ešte pred začiatkom vyhľadávania. Ako testovací súbor sme pri porovnávaní použili 70 zápasov.

6.1 Úprava parametrov pre lineárnu regresiu

Pri lineárnej regresii máme vzhľadom na jednoduchosť algoritmu minimum parametrov. Existujú aj parametre na zrýchlenie výpočtov, no tie nemajú vplyv na učenie. Z praktického hľadiska učenie ovplyvňujú iba nasledujúce parametre:

- fit_intercept - môže mať hodnoty True alebo False, určuje či sa má pre model vypočítať prekrytie.
- normalize - rovnako binárny parameter, ak je fit_intercept True, tak sa ignoruje, inak určuje či sa majú dáta normalizovať podľa priemeru [7].

Keďže lineárna regresia nevracia na výstupe diskrétne hodnoty, ale spojitú hodnotu pravdepodobnosti, nemôžeme použiť v našom prípade grid search. Ten totiž porovnáva rovnosť predikovaných hodnôt s ich reálnou hodnotou. Je teda potrebné, aby sme porovnávali hodnoty pravdepodobnosti a priradili ich konkrétnym výsledkom. Preto v prípade tohoto algoritmu mapujeme hodnoty od 0 po 0.5 ako prehru a tie ostatné ako výhru.

Keďže máme iba dva parametre s dvoma možnými hodnotami, testujeme štyri rôzne prípady. V každom z nich nám však vyšiel výsledok totožný, a to 69% na trénovacom súbore a 68,57% na testovacom. Môžeme teda tvrdiť, že s aktuálnym dátovým rámcom vieme dosiahnuť maximálne 68,57% úspešnosť pri vzorke 70 zápasov.

6.2 Úprava parametrov pre logistickú regresiu

Logistická regresia na rozdiel od lineárnej podporuje väčšie množstvo parametrov. Pri tomto algoritme môžeme využiť grid search. Je nutné podotknúť, že výstupom grid search-u je vždy iba úspešnosť na trénovacích dátach. Tá ale nemusí znamenať, že na testovacích dátach budeme podobne úspešní, ako sme už mohli sledovať pri meraniach v predchádzajúcej kapitole.

Prvým parametrom ktorý je nutné určiť je parameter solver, ktorý určuje spôsob riešenia regresie. Existuje niekoľko rôznych solverov. Účelom práce však nie je popis všetkých parametrov, o solveroch sa dočítate v dokumentácii algoritmu [8]. Za pomoci grid searchu sme získali výsledok najvhodnejšieho solveru. Je ním liblinear s výsledkom 68,81% na trénovacích dátach. Táto hodnota je pomerne nízka, pokúsime sa ju zvýšiť pomocou ďalších parametrov.

Ďalšími parametrami, ktoré môžu pomerne významne ovplyvniť schopnosť učenia sú parametre C a penalty. Parameter C podobne ako u SVM určuje silu regularizácie. Čím sú hodnoty menšie, tým je regularizácia striktnejšia. Parameter penalty určuje typ regularizácie, pri type l1 sa jedná o súčet absolútnych hodnôt parametrov priradených jednotlivým premenným v regresnej funkcii, zatiaľ čo pri type l2 sa jedná o súčet ich druhých mocnín. Testovaním rôznych hodnôt sme sa však dostali opäť len na výsledok 68,81%. Prednastavené hodnoty sa ukázali pri tomto algoritme ako najvhodnejšie. Rovnako ako pri algoritme lineárnej regresie sa nám ani tu úpravou parametrov nepodarilo zvýšiť úspešnosť.

6.3 Úprava parametrov pre SVM

Pri algoritme SVM ako prvé musíme zistiť najvhodnejší typ jadra. Typ jadra určuje spôsob tvorenia funkcie na predikciu, ako sme už spomínali pri opise tohoto algoritmu.

Ako prvé sme vyskúšali polynomiálne jadro. Keďže však trvalo vyše 6,5 hodín kým sa natrénoval jediný model, a úspešnosť tohto modelu bola pri testovacích dátach iba 50%, vylúčili sme tento typ jadra hneď na začiatku.

Pomerne nečakané výsledky prinieslo aj lineárne jadro. Doba trénovania modelov presiahla u tohto jadra hodinu. Výsledky u trénovacích dát však nedosiahli ani očakávaných 60%. Na trénovacích dátach sme dosiahli úspešnosť 58,05% a na testovacích iba 51,14%.

Z praktického hľadiska nám vyšli najlepšie výsledky pri použití jadra rbf. Trénovacie dáta dosiahli v tomto prípade úspešnosť vyše 71% a tým sme sa dostali na prijateľné čísla. Na testovacom súbore sme dosiahli 53%, čo sa vzhľadom na zvýšenú úspešnosť u trénovacích dát javí ako výhodnejšie než lineárne jadro. Musíme zdôrazniť, že trénovanie modelu v tomto prípade trvalo iba niekoľko sekúnd.

Po určení najvhodnejšieho typu jadra sme sa zamerali na úpravu číselných parametrov. Parameter C funguje podobne ako learning_rate pri algoritme XGB. Určením správnej hodnoty zamedzíme overfittingu či underfittingu predikčnej funkcie. Parameter gamma určuje, ako veľmi je dôležitá jednotlivá vzorka v trénovacích dátach. Pokiaľ je hodnota nízka dôležitosť jednotlivých vzoriek je označená ako ďalekosiahla, ak je vysoká má menší dopad. Základná hodnota auto bude v budúcnosti nahradená hodnotou scale, ktorú sme využívali doposiaľ. Hodnota auto totiž viedla k pretrénovaniu a výsledky na testovacích dátach boli veľmi nízke. Pre grid search sme použili hodnoty v rozsahu 0,001 až 10 pre gamma a 0.1 až 100 pre C. Vo výstupe 1 môžeme pozorovať, že akékoľvek hodnoty C vyššie ako 1 viedli k overfittingu (na trénovacích dátach bola úspešnosť 100%, no na testovacích 53,6%). Nižšie hodnoty naopak viedli k výrazne nízkej úspešnosti.

Listing 1: Výstup grid searchu pre SVM - úspešnosť pre testovacie a trénovacie dáta plus parametre.

```
Best: 0.552617 using {'C': 1, 'gamma': 'scale', 'kernel': 'rbf'}
0.535570 (0.535570) with: {'C': 0.1, 'gamma': 0.001, 'kernel': 'rbf'}
...
0.535973 (0.554094) with: {'C': 0.1, 'gamma': 'scale', 'kernel': 'rbf'}
0.533691 (0.999463) with: {'C': 1, 'gamma': 0.001, 'kernel': 'rbf'}
...
0.535570 (1.000000) with: {'C': 1, 'gamma': 10, 'kernel': 'rbf'}
0.552617 (0.789664) with: {'C': 1, 'gamma': 'scale', 'kernel': 'rbf'}
0.533154 (1.000000) with: {'C': 10, 'gamma': 0.001, 'kernel': 'rbf'}
...
0.532483 (1.000000) with: {'C': 100, 'gamma': 'scale', 'kernel': 'rbf'}
```

Gamma bola jednoznačne najvýhodnejšie nastavená v pôvodnej hodnote scale. Parameter C sme sa v ďalsom testovaní rozhodli upraviť na hodnoty od 0,5 do 8. Na ďalšom výpise 2 môžeme vidieť, že hodnoty C nad 1 vedú k overfittingu.

Listing 2: Výstup grid searchu pre SVM s hodnotou gamma=scale.

```
Best: 0.552617 using {'C': 1, 'gamma': 'scale', 'kernel': 'rbf'}
0.550336 (0.745101) with: {'C': 0.5, 'gamma': 'scale', 'kernel': 'rbf'}
0.549530 (0.772215) with: {'C': 0.8, 'gamma': 'scale', 'kernel': 'rbf'}
0.552617 (0.789664) with: {'C': 1, 'gamma': 'scale', 'kernel': 'rbf'}
0.548054 (0.845906) with: {'C': 2, 'gamma': 'scale', 'kernel': 'rbf'}
0.547114 (0.883893) with: {'C': 3, 'gamma': 'scale', 'kernel': 'rbf'}
0.542148 (0.924027) with: {'C': 5, 'gamma': 'scale', 'kernel': 'rbf'}
0.538926 (0.952483) with: {'C': 8, 'gamma': 'scale', 'kernel': 'rbf'}
```

Posledné testovanie 3 sme vyskúšali s hodnotami od 0,6 do 0,9. Ako najlepšia hodnota sa javí 0,75. Grid search využíva mechanizmus zvaný k-folding, čo je roz-

delenie trénovacieho súboru na k častí a následné vytváranie priemeru z modelov vytvorených na základe týchto čiastkových dát. Ako testovaciu sadu potom vyberie 10% danej k-tiny a vo výsledku určí priemer týchto čísel. Tento priemer sa v našom prípade nerovná skutočnej úspešnosti, keďže rozdeľovať náš dátový rámec na menšie časti je pre jeho malú veľkosť nevhodné. Po testovaní na celom dátovom rámci sme dosiahli s upravenými parametrami úspešnosť 76,79% na trénovacích dátach a 55,26 na testovacích.

Listing 3: Výstup grid searchu pre SVM s hodnotou gamma=scale, C=0.6-0.9.

```
Best: 0.551409 using {'C': 0.75, 'gamma': 'scale', 'kernel': 'rbf'}
0.535973 (0.554094) with: {'C': 0.1, 'gamma': 'scale', 'kernel': 'rbf'}
0.549799 (0.755839) with: {'C': 0.6, 'gamma': 'scale', 'kernel': 'rbf'}
0.549262 (0.760134) with: {'C': 0.65, 'gamma': 'scale', 'kernel': 'rbf'}
0.550604 (0.764564) with: {'C': 0.7, 'gamma': 'scale', 'kernel': 'rbf'}
0.551409 (0.767919) with: {'C': 0.75, 'gamma': 'scale', 'kernel': 'rbf'}
0.549530 (0.772215) with: {'C': 0.8, 'gamma': 'scale', 'kernel': 'rbf'}
0.550067 (0.776510) with: {'C': 0.85, 'gamma': 'scale', 'kernel': 'rbf'}
0.550604 (0.781074) with: {'C': 0.9, 'gamma': 'scale', 'kernel': 'rbf'}
```

6.4 Úprava parametrov pre XGB

Parametre využité pri tomto algoritme sme opísali už pri formálnej definícii algoritmu. Tu sa pokúsime zvýšiť úspešnosť pomocou ich úpravy. Od začiatku sa javí gradient boosting ako najvýhodnejší algoritmus na daný problém a v priemere sme práve pri tomto algoritme zaznamenávali najlepšie výsledky.

Ako prvé sme sa rozhodli odskúšať parametre max_depth a n_estimators. Pre parameter max_depth sme vybrali na začiatku hodnoty od 2 po 10 a 15. Pre parameter n_estimators sme hľadali parametre v rozsahu 0-100 s odstupom 5. Najlepšie z testu vyšla kombinácia kde bol parameter max_depth rovný dvom a n_estimators 75. Hodnoty n_estimators pod 50 mali horšie výsledky, a preto sme v ďalších tes-

toch zúžili rozsah iba na 50-100 s odstupom 2 a max_depth na 2-5. Ukázalo sa, že hodnota max_depth na hodnote 2 bola opäť najvhodnejšia. Tentokrát však v kombinácii s n_estimators=72. Úspešnosť na trénovacích dátach tu však bola iba 62%, čo je takmer o 3 percentá menej ako hodnota bez nastavovania parametrov. Preto je nutné nájsť správnu kombináciu aj ďalších parametrov.

V ďalšom kroku sme zúžili hodnoty n_estimators na 55-85 s krokom 1 a hodnotu max_depth sme ponechali na hodnote 2. Portfólio testovaných parametrov sme rozšírili o learning_rate, s hodnotami rovnými mocninám desiatky od 0,01 až po 100. Podľa výstupu grid searchu sme určili ako najlepšiu hodnotu learning_rate 0,01. Keď sme ju však použili, pozorovali sme značnú chybu pri určovaní dôležitosti parametrov, keďže v tomto prípade algoritmus priradil premennej DiffLP až 111 bodov, čo je približne 2-krát viac ako v iných prípadoch. Dôležitosť vyššiu ako 10 mali okrem tejto premennej iba 2 ďalšie, a zvyšné premenné boli označené ako nepodstatné. Preto sme sa rozhodli odskúšať ďalšie parametre min_child_weight a subsample súbežne s learning_rate. Testovali sme hodnoty od 1 po 10 pre min_child_weight a 0,1 až 1 pre subsample.

Výsledkom testovania bola kombinácia parametrov n_estimators=78, max_depth=2, learning_rate=0.1, min_child_weight=8 a subsample=0,7. Výsledok na 70 zápasoch tentokrát dosiahol 65,71%, čo je rovnaký výsledok ako bez úpravy parametrov.

Ďalšie parametre ktoré sme pridali do testovania boli scale_pos_weight v rozsahu 0,01 až 10 a colsample_bytree v rozsahu 0.1 až 1. Výsledná kombinácia parametrov bola pomerne odlišná od predchádzajúceho testu. Konkrétne n_estimators=71, max_depth=2, learning_rate=0.1, min_child_weight=7 a subsample=0,1. Hodnoty nových parametrov boli najlepšie keď colsample_bytree bolo rovné 0.6 a scale_pos_weight sa rovnalo 1. Pri týchto parametroch sme však mali rovnakú úspešnosť ako pri predchádzajúcom teste. Po rozšírení testov aj o parametre reg_lambda a reg_alpha sme taktiež nezaznamenali vyššiu úspešnosť ako 65,71%.

Po vykonaní všetkých testov môžeme tvrdiť, že algoritmus je schopný predpove-

dať výsledky s najlepšou pravdepodobnosťou 65,71%, pri vzorke 380 zápasov maximálne 68,95%.

6.5 Úprava parametrov pre náhodný strom

Algoritmus náhodného stromu nám s prednastavenými hodnotami dával pomerne zlé výsledky. Spôsobené to bolo pravdepodobne pretrénovaním modelu, keďže sme mali veľmi vysokú úspešnosť na trénovacích dátach. Preto nie je prekvapením, že práve pri tomto algoritme sme zaznamenali najväčšie zlepšenie úpravou parametrov.

Existuje určitá paralela medzi parametrami XGB a parametrami algoritmu náhodného stromu, keďže oba algoritmy sú založené na budovaní rozhodovacích stromov [9]. Opäť sme začali zo začiatku upravovať parametre max_depth a n_estimators. V tomto prípade sme skúšali hodnoty n_estimators rovné 4, 8, 16, 32, 64, 100, 200, 400 a 800 a max_depth v rozsahu 1 až 20. Najlepšie výsledky sme dosiahli obdobne ako pri boostingu pri hodnotách max_depth=7. Hodnota n_estimators však bola najvýhodnejšie nastavená pri hodnote 100. Pri týchto hodnotách sme sa síce na testovacom súbore dostali z hodnoty 98,33% na 65,1, no na testovacom sme zvýšili úspešnosť z necelých 52% na viac než 62%.

V nasledujúcom kroku sme začali hľadať vhodnú hodnotu parametru min_samples_split, ktorý reprezentuje minimálny počet vzoriek potrebných pre rozdelenie uzla. Z rozpätia 1 až 100 sa ako najlepšia hodnota javí 92 v kombinácii s hodnotami max_depth=9 a max_depth=0,1. Pri týchto hodnotách sme dosiahli na testovacích dátach 67,14%.

Predposledným parametrom, ktorý sme sa rozhodli meniť je parameter min_samples_leaf, a to v rozsahu 0,1 až 0,5. Výsledok grid searchu, hodnota 0,1, však nepriniesol želaný úspech. Úspešnosť predikcie sa nám v tomto prípade znížila na 65,71%.

Posledný parameter ktorý sme upravovali, bol parameter max_features, ktorý určuje najvyšší počet premenných pre rozdelenie uzla. Testovali sme hodnoty od

1 po počet premenných v dátovom rámci. Po spustení grid searchu so všetkými parametrami nám nakoniec vyšla najvýhodnejšia kombinácia hodnôt max_depth=5, min_samples_split=0.2, n_estimators=64, min_samples_leaf=0.1 a max_features=44. Pri týchto hodnotách sme dosiahli na testovacích dátach s 380 zápasmi úspešnosť predikcie 69,74%, čo je zatiaľ najvyšším dosiahnutým výsledkom.

7 Predikcia konkrétneho výsledku

Hlavným cieľom tejto práce je pomocou strojového učenia predpovedať čo najpresnejšie konkrétny výsledok zápasu a následne určiť percentuálnu pravdepodobnosť jednotlivých výsledkov. Predpovedať konkrétny výsledok môžeme dvoma spôsobmi. Prvý z nich je natrénovať jeden model tak, že bude predpovedať všetky tri možnosti výsledku. Druhou je metóda, kde natrénujeme dva modely. Jeden model bude predpovedať, či prvý tím zápas vyhrá alebo nie. Tento označíme WX. Ten druhý bude následne predpovedať pre zápasy kde bolo predpovedané že nevyhrá, či prehrá alebo nie. Tento označíme XL. Pravdepodobnosť výhry z modelu 1 budeme považovať za konečnú, pravdepodobnosť prehry alebo remízy dostaneme ako obrátenú hodnotu pravdepodobnosti výhry krát výstupy druhého modelu.

Prvým problémom pri druhom prístupe je, že na trénovanie budeme musieť vyžiť dva klasifikátory. Vzhľadom na podstatu problému totiž nemusí platiť, že najvhodnejšie parametre pre predikciu v modeli WX budú rovnaké ako v modeli XL. Rovnaký problém sa objavuje aj pri prvom spôsobe. Preto bolo nutné parametre pomocou algoritmu grid search vyhľadať znova.

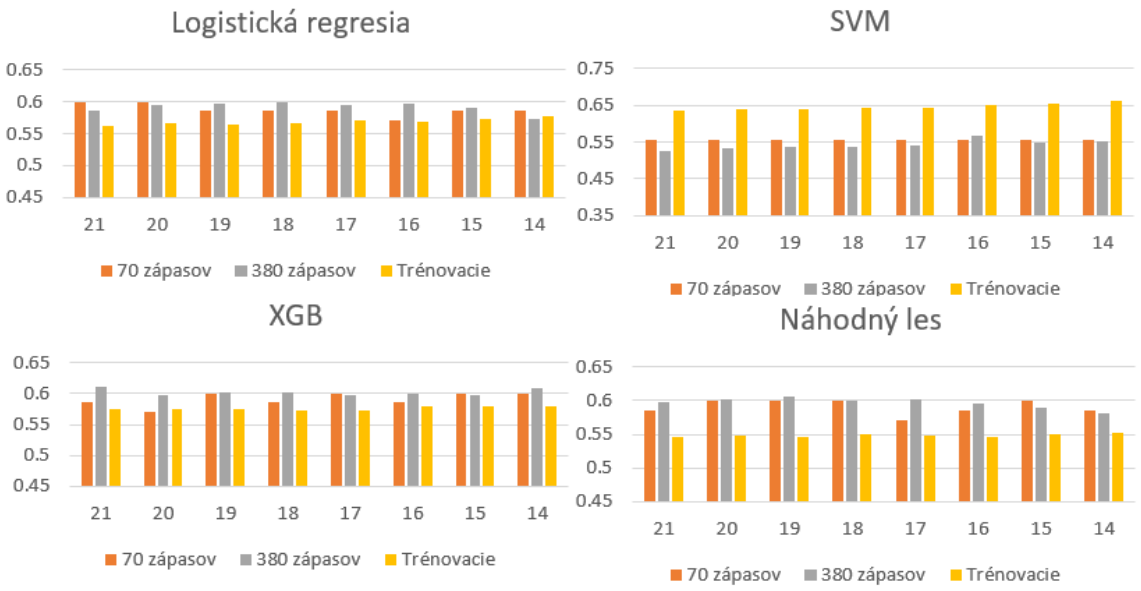
Pri prvom prístupe bolo nečakané, že mimo algoritmu SVM, mali všetky algoritmy parametre zhodné s modelom WX. Pri algoritme SVM bola odchýlka tiež pomerne malá, parameter C sa zmenil z 0,8 na 0,75.

Väčšie rozdiely sme zaznamenali pri predikcii v modeli XL. Pri lineárnej regresii sme zmenu nezaznamenali, pri algoritme SVM bola použitá opäť hodnota C=0,75. Pri algoritme XGB boli však takmer všetky parametre rozdielne. Výsledná kombinácia parametrov bola nasledovná, n_estimators=68, max_depth=2, learning_rate=0.1, min_child_weight=3, subsample=0,5, scale_pos_weight=1, colsample_bytree=1, reg_alpha=0,3 a reg_lambda=0,6. Obdobne to bolo pri algoritme náhodného stromu. Tu bola kombinácia parametrov max_depth=10, min_samples_split=0,2, n_estimators=34, min_samples_leaf=0,1, max_features=63.

Po určení parametrov sme mohli vyhodnotiť úspešnosť oboch prístupov.

7.1 Úspešnosť predikcie v jednom kroku

Ako prvú sme vyhodnocovali jednoduchšiu alternatívu. Pri predikcii výhry, prehry a remízy sme robili testy s testovacím súborom o veľkosti 70 a 380 zápasov. Opäť sme testovali dátové sady s rôznou dĺžkou, konkrétne 14 až 21 sezón. Na grafoch 7 – 1 môžeme vidieť porovnanie všetkých štyroch algoritmov testovaných s rôznymi počtami sezón a s rôzne dlhými testovacími sadami. Algoritmus lineárnej regresie je z tohto testovania vynechaný, keďže je určený, ako bolo spomínané pri jeho popise, na určenie pravdepodobnosti jednej z dvoch situácií.



Obrázok 7 – 1: Porovnanie úspešnosti predikcie na testovacích dátach s rôznou dĺžkou vzhľadom na počet sezón pre štyri sledované algoritmy.

Môžeme vidieť, že v tomto prípade sú výkyvy vzhľadom na počty sezón minimálne, no predsa pozorovateľné. Pri najúspešnejších algoritmoch môžeme pozorovať najlepšie výsledky pri počte 19 sezón. Tento trend sme pozorovali už pri predchádzajúcich testoch. Pomerne dobré výsledky ale môžeme sledovať aj pri 21 či 16 sezónach.

Celkovo najslabšie výsledky má opätovne algoritmus SVM, kde sme napriek vy-

sokej úspešnosti na trénovacích dátach zaznamenali najlepší výsledok pri testovacích dátach 380 zápasov 56,84%. Toto číslo je síce porovnateľné s inými prácami ktoré sme uviedli, no omnoho nižšie než výsledky iných algoritmov v našej práci.

Ostatné algoritmy dosahovali výsledky, ktoré by bolo možné využiť aj v praxi. Pri 380 zápasoch mal algoritmus XGB najlepšiu úslešnosť pri 21 sezónach, konkrétne 61,05%. Pri 70 zápasoch sme zaznamenali viacnásobne úspešnosť až 60%. Algoritmus náhodného lesa mal pri 380 zápasoch najlepšiu úspešnosť pri 19 sezónach, tuto 60,53%. Pri 70 zápasoch to bolo rovnako viacnásobne 60%. Logistická regresia mala úspešnosť až 60% pri 18 sezónach a rovnako 60% pri 21 a 20 sezónach pri použití 70 testovacích zápasov. Úspešnosť predikcie najnepravdepodobnejšieho výsledku bola sa pri týchto troch algoritmoch pohybovala okolo 84%. Toto číslo je o viac než 5% vyššie ako pri skúmaných dátach. Zaujímavosťou tiež je, že zo 108 remíz bolo pri logistickej regresii správne predikovaných 71, čo sa rovná úspešnosti 65,74%.

7.2 Úspešnosť predikcie pri oddelenej predikcii výhry a prehry

Druhá alternatíva je mierne zložitejšia. Musíme pri nej totiž trénovať dva modely, a v druhom testujeme len tie prípady, kde prvý model neurčil výhru. Vyhodnocovali sme preto dve úspešnosti. V prvej ide o úspešnosť určenia výhry alebo iného výsledku. Ak je predikovaný iný výsledok, testujeme dáta v druhom modeli. Ten má za úlohu určiť pravdepodobnosť prehry alebo niečoho iného. Výsledná úspešnosť sa teda rovná súčinu úspešností týchto dvoch modelov.

V tabuľkách 7 – 1 a 7 – 2 môžeme vidieť výsledky algoritmu SVM. Ten iba v dvoch prípadoch presiahol 50%, a preto považujeme akékoľvek úvahy o tomto algoritme ako bezpredmetné. WX a XL označujú v grafe predikcie pre jednotlivé modely pri danom počte zápasov, celková úspešnosť je v stĺpcoch nazvaných spolu.

Tabuľka 7 – 1: Úspešnosť predikcie algoritmu SVM pri oddelenej predikcii výhry a prehry, vzorka 70 zápasov.

Počet sezón	Model WX	Model XL	Spolu
21	0.5429	0.7556	0.41021524
20	0.5857	0.7708	0.45145756
19	0.7143	0.7174	0.51243882
18	0.6429	0.6512	0.41865648
17	0.6571	0.7027	0.46174417
16	0.7	0.7568	0.52976
15	0.6714	0.6667	0.44762238
14	0.6571	0.7442	0.48901382

Tabuľka 7 – 2: Úspešnosť predikcie algoritmu SVM pri oddelenej predikcii výhry a prehry, vzorka 380 zápasov.

Počet sezón	Model WX	Model XL	Spolu
21	0.6105	0.749	0.4572645
20	0.5605	0.7415	0.41561075
19	0.6421	0.7366	0.47297086
18	0.6184	0.7021	0.43417864
17	0.6	0.7467	0.44802
16	0.6079	0.6978	0.42419262
15	0.6026	0.7227	0.43549902
14	0.5763	0.7093	0.40876959

V ďalších tabuľkách 7 – 3, 7 – 4, 7 – 5, 7 – 6 môžeme vidieť výsledky oboch porovnávaných regresných algoritmov. Môžeme si všimnúť, že výsledky sa s nezhodujú s predchádzajúcou metódou ani čo sa týka úspešnosti, ani čo sa týka najlepších počtov

sezón. Tento krát máme najlepšie výsledky pri 17-19 sezónach, hodnoty sa pohybujú medzi 51 až 54%. Pri 70 zápasoch je to u lineárnej regresie 58% pri 18 sezónach a u logistickej 59,4% pri 18 sezónach. V priemere sme však taktiež dosahovali hodnoty oproti predošlej metóde nižšie, mnoho krát pod 50%.

Tabuľka 7 – 3: Úspešnosť predikcie algoritmu lineárnej regresie pri oddelenej predikcii výhry a prehry, vzorka 70 zápasov.

Počet sezón	**Model WX**	**Model XL**	**Spolu**
21	0.6286	0.5676	0.35679336
20	0.6857	0.7333	0.50282381
19	0.8	0.65	0.52
18	0.7571	0.7674	0.58099854
17	0.7143	0.7297	0.52122471
16	0.6429	0.7576	0.48706104
15	0.6714	0.6842	0.45937188
14	0.6429	0.6842	0.43987218

Tabuľka 7 – 4: Úspešnosť predikcie algoritmu lineárnej regresie pri oddelenej predikcii výhry a prehry, vzorka 380 zápasov.

Počet sezón	**Model WX**	**Model XL**	**Spolu**
21	0.6553	0.7342	0.48112126
20	0.6263	0.7273	0.45550799
19	0.6816	0.7681	0.52353696
18	0.6947	0.7658	0.53200126
17	0.6553	0.8143	0.53361079
16	0.6447	0.7349	0.47379003
15	0.6947	0.6947	0.48260809
14	0.6737	0.6737	0.45387169

Tabuľka 7 – 5: Úspešnosť predikcie algoritmu logistickej regresie pri oddelenej predikcii výhry a prehry, vzorka 70 zápasov.

Počet sezón	Model WX	Model XL	Spolu
21	0.6143	0.5526	0.33946218
20	0.6857	0.8	0.54856
19	0.8	0.6842	0.54736
18	0.74729	0.8	0.59432
17	0.7143	0.7027	0.50193861
16	0.6143	0.7576	0.46539368
15	0.6714	0.7105	0.4770297
14	0.6429	0.6579	0.42296391

Tabuľka 7 – 6: Úspešnosť predikcie algoritmu logistickej regresie pri oddelenej predikcii výhry a prehry, vzorka 380 zápasov.

Počet sezón	Model WX	Model XL	Spolu
21	0.6526	0.7466	0.48723116
20	0.6263	0.7415	0.46440145
19	0.6816	0.7826	0.53342016
18	0.6974	0.788	0.5495512
17	0.65	0.7857	0.510705
16	0.6447	0.723	0.4661181
15	0.6921	0.7798	0.53969958
14	0.6737	0.7717	0.51989429

Rovnako slabšie výsledky môžeme odčítať aj v tabuľkách 7 – 7, 7 – 8, 7 – 9, 7 – 10, kde máme výsledky algoritmov XGB a náhodného lesa.

Najlepší výsledok algoritmu XGB je 54,8% pri 15 sezónach. Toto číslo je síce

porovnateľné s inými riešeniami, no menšie ako výsledky našej priamej metódy. Podobne je to aj pri testovacom súbore 70 zápasov, kde je najlepší výsledok 59,54%. Algoritmus náhodného stromu má takmer 61% úspešnosť na teste so 70 zápasmi, no iba 52,6% s 380 zápasmi v testovacom súbore. Priemerné výsledky sú rovnako ako u ostatných algoritmov značne slabšie ako pri priamej metóde.

Tabuľka 7 – 7: Úspešnosť predikcie algoritmu XGB pri oddelenej predikcii výhry a prehry, vzorka 70 zápasov.

Počet sezón	**Model WX**	**Model XL**	**Spolu**
21	0.6571	0.5676	0.37296996
20	0.7	0.7727	0.54089
19	0.8143	0.7317	0.59582331
18	0.7429	0.7368	0.54736872
17	0.7	0.675	0.4725
16	0.6571	0.75	0.492825
15	0.6429	0.6842	0.43987218
14	0.6857	0.7692	0.52744044

Tabuľka 7 – 8: Úspešnosť predikcie algoritmu XGB pri oddelenej predikcii výhry a prehry, vzorka 380 zápasov.

Počet sezón	**Model WX**	**Model XL**	**Spolu**
21	0.6737	0.7467	0.50305179
20	0.6711	0.7157	0.48030627
19	0.6921	0.7681	0.53160201
18	0.6711	0.7653	0.51359283
17	0.6763	0.7944	0.53725272
16	0.6737	0.7019	0.47287003
15	0.7211	0.76	0.548036
14	0.6816	0.7808	0.53219328

Tabuľka 7 – 9: Úspešnosť predikcie algoritmu náhodného stromu pri oddelenej predikcii výhry a prehry, vzorka 70 zápasov.

Počet sezón	**Model WX**	**Model XL**	**Spolu**
21	0.7286	0.625	0.455375
20	0.6429	0.7826	0.50313354
19	0.8	0.7619	0.60952
18	0.7571	0.7179	0.54352209
17	0.7	0.7381	0.51667
16	0.6714	0.8485	0.5696829
15	0.6842	0.675	0.461835
14	0.6714	0.725	0.486765

Tabuľka 7 – 10: Úspešnosť predikcie algoritmu náhodného stromu pri oddelenej predikcii výhry a prehry, vzorka 380 zápasov.

Počet sezón	**Model WX**	**Model XL**	**Spolu**
21	0.6289	0.7391	0.46481999
20	0.6474	0.7324	0.47415576
19	0.6921	0.7606	0.52641126
18	0.6684	0.7617	0.50912028
17	0.6526	0.7753	0.50596078
16	0.7105	0.7322	0.5202281
15	0.6789	0.7424	0.50401536
14	0.6763	0.7557	0.51107991

Z tohto dôvodu môžeme vyvodiť, že nami skúmané riešenie je slabo podpriemerné až priemerné. Lepšie výsledky sme v každom zo sledovaných algoritmov dosiahli priamou predikciou jednej z možností výhry, prehry, alebo remízy.

8 Portál pre stávkovanie

V prvej časti zadania práce sme dostali za úlohu vytvoriť aj webový portál pre stávkovanie, kde využijeme nami určené kurzy pomocou strojového učenia. V tejto kapitole si opíšeme vytvorený portál a jeho funkčné časti. Webový portál sme pracovne nazvali DP-betting.

Keďže hlavným cieľom práce bolo skúmanie algoritmov strojového učenia a ich možností predikcie výsledkov zápasov, stránka ktorú sme vytvorili je iba modelom toho, ako by mohla reálna stránka využívať takto určené pravdepodobnosti výsledkov a na ich základe určené kurzy. Stránku sme písali v jazyku PHP a HTML, pričom dáta sú ukladané v databáze mysql. Na vzhľad stránky sme využili kaskádové štýly pomocou jazyka CSS a frameworku Bootstrap.

Všetky zdrojové súbory a databáza stránky sa nachádzajú v prílohe.

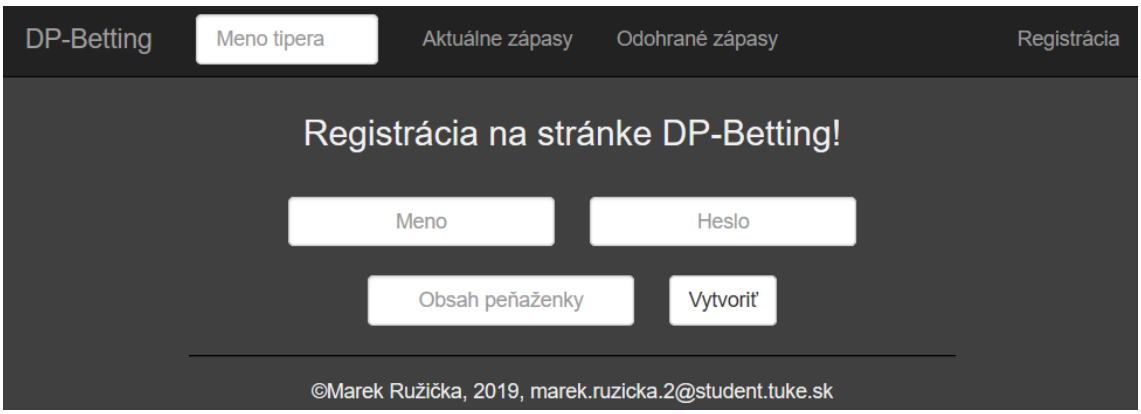
8.1 Úvodná stránka

Na úvodnej stránke sme umiestnili iba informáciu o účele stránky. Dôležitejšou časťou je vrchný panel, odkiaľ sa dá dostať na iné časti stránky. Úvodnú stránku môžete vidieť na obrázku 8 – 1. V ľavom rohu sa nachádza logo, ktoré naviguje užívateľa na hlavnú stránku. Hneď vedľa sa nachádza vyhľadávanie, ktoré nás presmeruje na stránku s výpisom stávok užívateľa, ktorého meno sme do tohto poľa zadali. Ďalšie dva odkazy nás presmerujú na stránky so zoznamom nadchádzajúcich zápasov (Aktuálne zápasy) a už odohraných zápasov (Odohrané zápasy). Posledným odkazom celkom vpravo je Registrácia, kde si môže užívateľ vytvoriť nový účet. V danom modeli sme ale neriešili prihlasovanie, autentifikácia prebieha iba počas vytvárania stávok.

Obrázok 8 – 1: Úspešnosť predikcie pri oddelenej predikcii výhry a prehry algoritmom náhodného stromu.

8.2 Registrácia

Pre registráciu sme si pripravili jednoduchý formulár, kde užívateľ zadá iba základné informácie potrebné pre vytvorenie profilu. Tými sú meno, heslo a obsah peňaženky. Keďže je náš web prezentačným modelom, a nepoužívame reálne peniaze, v databáze na reprezentáciu disponibilnej čiastky na tipovanie používame práve túto hodnotu. Užívateľ potom môže vytvoriť virtuálnu stávku maximálne tak vysokú, akú hodnotu zadal pri vytvorení účtu. V databáze sa nemôže nachádzať viac užívateľov s rovnakým menom. Registračný formulár môžete vidieť na obrázku 8 – 2.

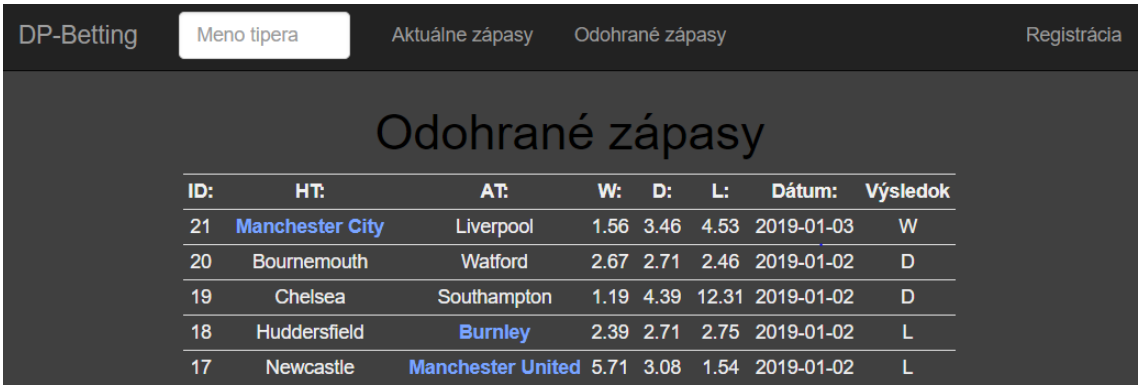


Obrázok 8 – 2: Registračný formulár na webovom portáli pre stávkovanie.

8.3 Odohrané a nadchádzajúce zápasy

V zozname odohraných zápasov sa nachádzajú všetky zápasy, ktoré sa uskutočnili v minulosti. Vzhľadom na dátovú sadu, ktorú sme získali v januári, sme pre potreby prezentácie označili ako aktuálny dátum 5.1.2019. Zápasy odohrané pred týmto dátumom sú preto zobrazené na tejto stránke.

Z dôvodu šetrenia miesta sme na obrázku 8 – 3 zobrazili iba prvých päť zápasov. V prvom stĺpci tabuľky sa nachádza identifikátor zápasu, pod ktorým je uložený v databáze. Ďalšie dva stĺpce obsahujú názvy tímov, pričom ak existuje víťazný tím, je označený modrou farbou. V nasledujúcich troch stĺpcoch sú nami určené hodnoty kurzov. V databáze máme pri zápasoch uložené pravdepodobnosti, ktoré sme získali pomocou strojového učenia. Po pripočítaní 5% ku každej pravdepodobnosti sme získali prevrátenú hodnotu, ktorá určuje stávkový kurz, ako sme písali v kapitole 3. Ďalší stĺpec obsahuje informáciu o dátume konania zápasu a v poslednom stĺpci sme vypísali finálny výsledok zápasu v tvare W, D alebo L.



Obrázok 8 – 3: Prvých 5 odohraných zápasov v tabuľke.

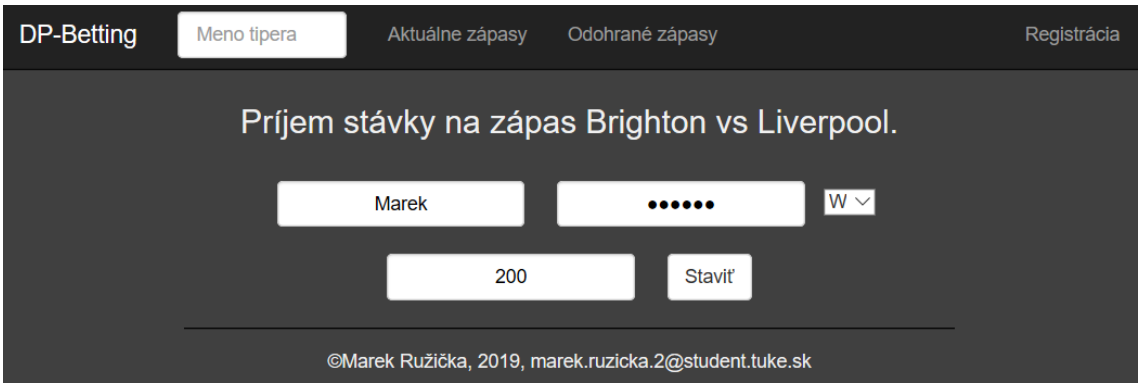
Ako môžete na obrázku 8 – 4 vidieť, tabuľka nadchádzajúcich zápasov sa podstatne nelíši od tej s odohranými. Rozdielom je jedine to, že nepoznáme výsledok, a tak nevieme vytvoriť posledný stĺpec. Miesto výsledkov sme však v poslednom stĺpci umiestnili pre každý zápas tlačidlo, ktoré užívateľa presmeruje na stránku, kde si na zápas môže staviť.

Obrázok 8 – 4: Prvých 5 nadchádzajúcich zápasov v tabuľke.

8.4 Vytváranie stávok

Odklikom na tlačidlo Staviť stránka užívateľa presmeruje na formulár na obrázku 8 – 5. V tomto formulári môže užívateľ vytvoriť stávku použitím svojho mena a hesla, ktoré zadal pri registrácii. Pokiaľ sa meno a heslo nezhodujú, stávka sa nevytvorí.

Rovnako ak stavená suma prevyšuje čiastku ktorou užívateľ virtuálne disponuje, stávka nebude prijatá. V menu si užívateľ vyberie na aký výsledok chce staviť z možností W/D/L, a ak je všetko v poriadku, systém stávku zapíše do databázy. Stávky sa následne dajú zobraziť pomocou už spomínaného vyhľadávača.



Obrázok 8 – 5: Formulár pre vytváranie stávok.

8.5 Prehľad stávok

Poslednou časťou stránky je prehľad existujúcich stávok. Tento je opäť vo forme tabuľky, viď. obr. 8 – 6. V tabuľke sa nachádzajú informácie o tímoch, ktoré hrali, o stávke ktorú užívateľ na zápas podal, stavenej sume a kurze ktorý sme určili na tento zápas. V databáze sú však uložené pri jednotlivých stávkach iba identifikátory zápasu a užívateľa s informáciou o stavenej sume, tipe a konečnom výsledku. Výsledok v predposlednom stĺpci je potom porovnávaný s tipovaným výsledkom. Pokiaľ užívateľ vyhral, v poslednom stĺpci je zobrazená suma, ktorú by dostal vyplatenú, ako súčin kurzu a stavenej sumy.

Obrázok 8 – 6: Prehľad stávok používateľa Marek.

9 Vyhodnotenie

Počas vypracovania tejto práce sme vykonali veľké množstvo meraní, na základe ktorých môžeme hodnotiť celkový prínos. Vykonávali sme merania pre päť rôznych algoritmov strojového učenia, vďaka ktorých môžeme určiť najvhodnejší algoritmus. Ďalej sme sa venovali dôležitosti jednotlivých premenných, kde sa ukázalo, že dátová sada s väčším množstvom informácií o tíme prináša lepšie výsledky predikcie. Taktiež sme sa zaoberali najvhodnejším počtom sezón použitých pre trénovanie modelov, kde sa zasa ukázalo, že v danom probléme nemusí nutne viac dát znamenať lepšie výsledky. V neposlednom rade sme porovnávali dve metódy predikcie, s účelom lepšej predikcie remíz, ktoré sú klasickými metódami často vynechané. Ako posledný krok v práci sme vytvorili webové rozhranie, v ktorom sme využili dáta predikované jedným z modelov na vytvorenie kurzov.

V začiatkoch práce sme vykonávali testy na najjednoduchšej dátovej sade obsahujúcej iba malé množstvo informácií o aktuálnej forme tímu založenej na posledných piatich zápasoch. Výsledky pri tejto dátovej sade však boli pomerne slabé, prakticky podpriemerné. Preto sme dátovú sadu rozšírili o množstvo dlhodobejších štatistík tímov, čo umožnilo dvihnúť úspešnosť v mnohých prípadoch približne o 3%. Počas všetkých meraní sa však ako najdôležitejšie javili informácie o postavení v tabuľke, prebraté kurzy, počty bodov a gólov oboch tímov.

Počas všetkých meraní zameraných na vytvorenie najvhodnejšej dátovej sady sme sa taktiež pokúšali zistiť, či a ako vplývajú informácie o dátume na úspešnosť predikcie. Vo väčšine prípadov sa však ukázalo, že model trénovaný pomocou dátovej sady s informáciou o dni a mesiaci konania má horšie výsledky ako ten, kde sa táto informácia nenachádzala. V tomto prípade musíme ale poznamenať, že v niektorých prípadoch boli výsledky s dátumom lepšie, a preto by možno bolo vhodné sa dôkladnejšie na tento aspekt pozrieť.

Vyhodnocovať to, ktorý algoritmus má najlepšie výsledky bolo možné až po tom, ako sme vyhodnotili najlepšie parametre pre jednotlivé algoritmy pomocou algoritmu

grid search. Tento postup sme museli opakovať celkovo dva-krát, pre klasickú predikciu a pre predikciu v oddelených krokoch osobitne, keďže sa prakticky jednalo o rôzne modely. Vďaka úprave parametrov sa nám podarilo zvýšiť úspešnosť najmä pri algoritme SVM, a to až o 6%.

Po určení korektných parametrov sme prešli merania celkovej úspešnosti predikcie, ktorú môžeme označiť ako finálnu úspešnosť v našej práci. Celý čas sme sa paralelne s meraniami úspešnosti v rámci jednotlivých aspektov pokúšali určiť aj to, aká veľká dátová sada je najviac vyhovujúca. Merania sme preto uskutočňovali na dátových sadách, ktoré obsahovali dáta v rozsahu 21 až 14 sezón.

Napriek faktu, že niektoré práce spomínané v kapitole 3 brali ako testovací súbor iba jeden alebo dva herné týždne, my sme sa snažili vytvoriť komplexnejší testovací súbor. Preto sme používali na testovanie v jednom prípade 70 a v druhom prípade 380 zápasov. Z hľadiska úspešnosti na rôzne dlhom dátovom súbore sme však dostávali opakovane konštantnejšie výsledky pri dlhšom testovacom súbore, čo je spôsobené jeho väčšou škálou možností ktoré testujeme. Pri menšej dátovej sade je väčšia šanca, že budú výsledky skreslené.

Metóda, kde sme oddelene predpovedali výhru a prehru, sa nakoniec ukázala ako mylná. V tejto metóde sme dosiahli najlepší výsledok 54,8%, ktorý dosiahol algoritmus XGB pri 15 sezónach. Tento výsledok je slabší ako v klasickej metóde, kde predpovedáme jednu z troch možností. Napriek tomu stojí za zmienku, že tento výsledok je lepší, ako priemer dát, z ktorých sme čerpali, ale aj viacerých prác skúmajúcich možnosti predikcie spomínaných v závere kapitoly 3. Algoritmus SVM nepresiahol takmer v žiadnom prípade 50%, logistická regresia dosiahla mierne vyššie hodnoty ako lineárna, blížiace sa k hodnote 55%, pri počte 18 sezón. Algoritmus náhodného lesa mal najlepší výsledok 52,6%.

Pri tradičnej predikčnej metóde sme opäť vylúčili SVM ako algoritmus s najhoršími výsledkami. Lineárna regresia pre svoju podstatu v tomto teste nebola zahrnutá. Logistická regresia ako aj algoritmus náhodného stromu sa pohybovali vo väčšine prípadov v hodnotách tesne pod 60%. Algoritmus XGB sa v tomto prípade ukázal

ako najviac vhodný, s výsledkom až 61,05% pri 21 sezónach. V mnohých prípadoch však algoritmus XGB ale aj algoritmus náhodného lesa presiahli hranicu 60%. Tieto výsledky sú vyššie než ktorýkoľvek z výsledkov skúmaných prác v kapitole 3, ktoré sa zaoberali futbalom. Dobré výsledky sme mali aj pri predikcii remíz, kde sa nám zo 108 remíz podarilo určiť až 71. Pravdepodobnosť, že systém priradí najnižšiu pravdepodobnosť výsledku, ktorý sa nestane, sa blížila v našom riešení až k 85%.

Môžeme preto konštatovať, že sme dokázali vytvoriť model, ktorý má lepšiu úspešnosť než doteraz publikované práce. Ďalším rozšírením tejto práce by mohli byť informácie o aktuálnej zostave tímu, či štatistiky ohľadom stávkovania užívateľov na jednotlivé možnosti. Dokázali sme však, že aj na základe voľne dostupných a ľahko získateľných informácií je možné vytvoriť kvalitný predikčný model.

Na záver sme naše výsledky využili vo webovom portáli, kde sme ich pretvorili na stávkové kurzy. Vďaka tomu môžeme viditeľne prezentovať využiteľnosť výstupu našej práce v praxi.

10 Záver

Hlavným cieľom tejto diplomovej práce bolo porovnať využitie algoritmov lineárnej regresie, SVM a algoritmu XGB implementovaného v knižnici Tensorflow v jazyku python a nájsť najvhodnejší algoritmus pre určenie výsledku zápasu. Ďalej bolo úlohou predikované hodnoty pravdepodobností výsledkov premeniť na stávkové kurzy, a tieto využiť v stávkovom systéme, ktorý navrhneme.

K skúmaným algoritmom sme mimo tie v zadaní v kapitole dva pridali ďalšie dva, ktoré sa javili ako vhodné algoritmy na riešenie daného problému. Tými bola logistická regresia a náhodný strom. V tejto kapitole sme sa iba formálne zoznámili s fungovaním algoritmov, aby sme v ďalších častiach práce mohli s týmito algoritmami pracovať. Pridanie týchto algoritmov sa ukázalo ako vhodná voľba, keďže algoritmy lineárna regresia a SVM sa ukázali ako nedostačujúce.

V kapitole číslo 3, nazvanej Určovanie stávkových kurzov, sme si prešli metódy tvorby kurzov. Od najjednoduchšej, ktorou je preberanie kurzov, cez funkcie pre ich výpočet až po využitie strojového učenia. V tejto kapitole sme si taktiež prešli rôzne existujúce riešenia, aby sme mohli našu prácu porovnať.

V štvrtej kapitole sme rozanalyzovali získané dáta a prešli sme jednotlivé premenné a zdôvodnili ich dôležitosť. Na túto kapitolu totiž priamo nadviazala kapitola 5, Prvé trénovanie modelov. V tejto kapitole sme následne skúmali úspešnosť predikcie na pripravenom modeli a model rozšírili o množstvo dlhodobých štatistík.

V šiestej kapitole sme sa po vytvorení kvalitného dátového rámca zaoberali úpravou parametrov jednotlivých algoritmov, vďaka čomu sme pri viacerých dvihli úspešnosť predikcie. S úpravou parametrov sme sa čiastočne zaoberali aj v ďalšej kapitole venovanej predikcii konkrétneho výsledku, keďže tu sme porovnávali dva rozdielne modely. Táto kapitola je najdôležitejšou, pretože v tejto kapitole dostávame obraz o celkovej úspešnosti pozorovaných modelov a algoritmov.

V poslednej kapitole sme predikované dáta využili v modeli stávkového portálu, kde sme prezentovali praktické využitie výstupu našej práce.

V práci sa nám podarilo vytvoriť kvalitný dátový model, ktorý sme použili pri trénovaní jednotlivých algoritmov. Po kompletizácii všetkých testov môžeme konštatovať, že najlepšie výsledky dosiahli algoritmy XGB, náhodný les a logistická regresia. Všetky tri algoritmy dosiahli pomerne dobré výsledky. V závere môžeme tvrdiť, že naše riešenie predikcie výsledku zápasu má úspešnosť dokonca lepšiu než iné, už existujúce riešenia.

Podarilo sa nám taktiež preukázať využiteľnosť dát v praxi na modeli stávkového portálu. Je viditeľné, že z výstupu vieme vytvoriť stávkové kurzy tak, aby stávková spoločnosť na systéme zarobila, no neukrátila užívateľa o zážitok z hry.

V budúcnosti by bolo možné predikciu ešte viac vylepšiť rôznymi ďalšími údajmi. V prípade nasadenia do prevádzky by sa mohli zbierať informácie o množstve stávok na daný výsledok a analyzovať to, či majú ľudia tendenciu tipovať správne. Ďalšími indikátormi by mohli byť hráči, ktorých nasadí tím, zvyčajné taktiky tímov a podobne. Z hľadiska predikcie by mohlo byť zaujímavým indikátorom aj množstvo zmienok o tímoch na rôznych sociálnych sieťach a podobne.

Literatúra

[1] Catalin Barboianu, 2006. *Probability Guide to Gambling: The Mathematics of dice, slots, roulette, baccarat, blackjack, poker, lottery and sport bets* In: Infarom, 2006, ISBN: 973-87520-3-5

[2] Pedro Domingos, 2012. *A Few Useful Things to Know about Machine Learning.* In: Communications of the ACM 55(10)

[3] Ron Kohavi; Foster Provost (1998). *"Glossary of terms". Machine Learning 30* Dostupné online: https://link.springer.com/content/pdf/10.1023%2FA%3A1017181826899.pdf

[4] Jean-Philippe Vert; Koji Tsuda; Bernhard Schölkopf (2004). *A primer on kernel methods* Dostupné online: http://members.cbio.mines-paristech.fr/ jvert/publi/04kmcbbook/kernelprimer.pdf

[5] (2019). *1.4. Support Vector Machines* Dostupné online: https://scikit-learn.org/stable/modules/svm.html

[6] (2019). *GPU support* Dostupné online: https://www.tensorflow.org/install/gpu https://www.tensorflow.org/install/gpu

[7] (2019). *Linear regression* Dostupné online: https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LinearRegression.html

[8] (2019). *Logistic regression* Dostupné online: https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html

[9] (2019). *Random forest* Dostupné online: https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html

[10] (2019). *Premier league* Dostupné online: https://www.premierleague.com

[11] (2019). *Betexplorer* Dostupné online: https://www.betexplorer.com/soccer/england/premier-league/

[12] (2019). *Training and Test Sets: Splitting Data* Dostupné online: https://developers.google.com/machine-learning/crash-course/training-and-test-sets/splitting-data

[13] Amnon Shashua (2009). *Introduction to Machine Learning: Class Notes 67577* arXiv:0904.3664v1

[14] C. M. Bishop (2006). *Pattern Recognition and Machine Learning.* In: Springer, 2006, ISBN: 0-387-31073-8

[15] David A. Freedman (2009), *Statistical Models: Theory and Practice, Cambridge University Press. p. 26.* ISBN: 0-521-11243-5

[16] Giuseppe Bonaccorso (2017), *Machine Learning Algorithms, Packt Publishing* ISBN: 978-1-78588-962-2

[17] Kumar, Gunjan (2013), *Machine Learning for Soccer Analytics* DOI: 10.13140/RG.2.1.4628.3761

[18] Z. Reitermanová (2010), *Data Splitting, WDSÍ10 Proceedings of Contributed Papers, Part I, 31–36, 2010* ISBN: 978-80-7378-139-2

[19] Purucker, Michael C. (1996), *Neural network quarterbacking* DOI: 10.1109/45.535226

[20] Khan, Joshua (2003). *Neural Network Prediction of NFL Football Games* Dostupné online: http://homepages.cae.wisc.edu/ ece539/project/f03/kahn.pdf

[21] McCabe, A., Trevathan, J. (2008) *Artificial Intelligence in Sports Prediction* DOI: 10.1109/itng.2008.203

[22] Tax, N., Joustra, Y (2015) *Predicting The Dutch Football Competition Using Public Data: A Machine Learning Approach* DOI: 10.13140/RG.2.1.1383.4729

Zoznam príloh

Príloha A Používateľské príručka

Príloha B Systémová príručka

Príloha C CD médium - práca v elektronickej podobe, zdrojové kódy a dátové sady