

Slovenská technická univerzita v Bratislave  
Fakulta informatiky a informačných technológií  
FIIT-XXXX-XXXX

**Peter Markuš**  
**Interaktívne tutoriály v strojovom učení**  
Bakalárska práca

Vedúci práce: Mgr. Peter Laurinec

December 2017



Slovenská technická univerzita v Bratislave  
Fakulta informatiky a informačných technológií  
FIIT-XXXX-XXXX

**Peter Markuš**  
**Interaktívne tutoriály v strojovom učení**  
Bakalárska práca

Študijný program: Informatika

Študijný odbor: 9.2.1 Informatika

Miesto vypracovania: Ústav informatiky a softvérového inžinierstva, FIIT STU,  
Bratislava

Vedúci práce: Mgr. Peter Laurinec

December 2017



Čestne vyhlasujem, že som túto prácu vypracoval samostatne, na základe konzultácií a s použitím uvedenej literatúry.

V Bratislave, 11.12.2017

Peter Markuš



# Annotation

Slovak University of Technology Bratislava

Faculty of Informatics and Information Technologies

Degree Course: Informatics

Author: Peter Markuš

Bachelor Thesis: Interactive tutorials in machine learning

Supervisor: Mgr. Peter Laurinec

December 2017

The aim of the bachelor thesis is to provide basic overview of machine learning, its used approaches and basic algorithms. The greatest emphasis will be placed upon classification methods, in which will be used several algorithms focusing on the text data in particular. The main task will be to explain the processes of these algorithms to users in the simplest way possible through interactive web application. The application will provide an interactive programming environment through which the user's problem-solving tasks can be solved and provide helpful visualization. The greatest attention will be paid to the Naive Bayesian classifier, along with providing extensive interactive visualizations to allow deep tracking of its computational processes. By using interaction elements, we will be able to experiment with different algorithm configurations and observe relationships between data samples and its attributes via graphs.





# Anotácia

Slovenská technická univerzita v Bratislave

Fakulta informatiky a informačných technológií

Študijný program: Informatika

Autor: Peter Markuš

Bakalárska práca: Interaktívne tutoriály v strojovom učení

Vedúci bakalárskeho projektu: Mgr. Peter Laurinec

December 2017

Cieľom bakalárskej práce je podať základný prehľad o strojovom učení, jeho používaných prístupov a základných algoritmov. Najväčší dôraz bude kladený na metódy klasifikácie, pri ktorých sa použije niekoľko algoritmov zamerajúc sa najmä na textové dáta. Hlavnou úlohou práce bude vysvetliť fungovanie týchto algoritmov používateľom čo najjednoduchším možným spôsobom prostredníctvom interaktívnej webovej aplikácie. Aplikácia bude poskytovať interaktívne programové prostredie, cez ktoré bude možné riešiť problémové úlohy daných algoritmov zo strany používateľa a poskytnúť k nim aj prípadnú nápomocnú vizualizáciu. Najväčšia pozornosť bude venovaná Naivnému Bayesovskému klasifikátoru, pri ktorom budú poskytované rozsiahle interaktívne vizualizácie, umožňujúce sledovať hlbšie jeho výpočtové procesy. Pomocou interakčných prvkov bude možné experimentovať s rôznymi konfiguráciami algoritmu a pozorovať vzťahy medzi vzorkami dát a ich atribútov cez grafy.







# Obsah

<b>1 Úvod</b> .....	1
<b>2 Analýza problému</b> .....	5
2.1 Obraz o stave riešenia daného problému .....	5
2.2 Zdôvodnenie voľby spôsobu riešenia a jeho stručný opis .....	7
<b>3 Opis riešenia návrhu</b> .....	9
3.1 Návrh vizualizácii Naivného Bayesa .....	9
<b>4 Literatúra</b> .....	15



# Kapitola 1

## Úvod

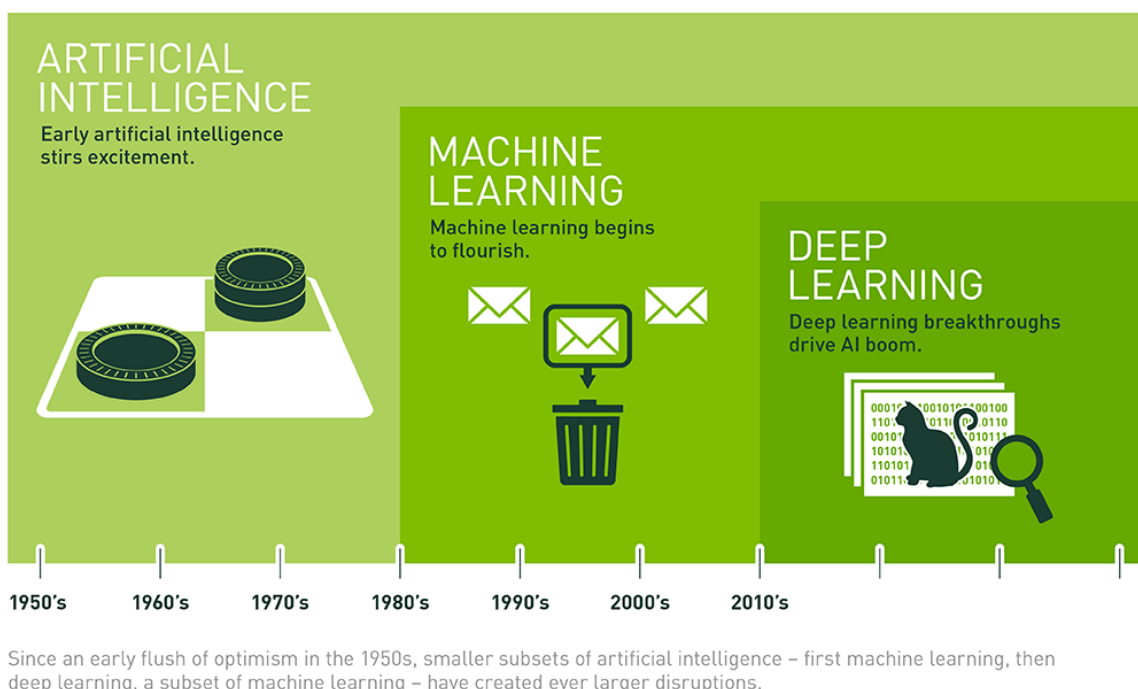
V súčasnosti sa so strojovým učením stretávame čoraz viac, hoci si to ľudia nemusia vôbec uvedomovať. Vzorovým príkladom môže byť vyhľadávanie na internete, pri ktorom sa softvér naučil vhodne vyhodnocovať výber stránok podľa predstáv používateľa. Taktiež pri prezeraní emailov je to naučený spamovací filter, rozlišujúci relevantné emaily od spamov. Schopnosť systém učiť sa očividne prináša množstvo nových možností riešenia komplexných problémov, ktorými sa zaoberá oblasť umelej inteligencie.

Pôvodne sa pri zostrojovaní inteligentných systémov vychádzalo z algoritmov ako hľadať riešenia v problémoch. Napríklad tak, ako pri hľadaní najkratšej cesty z bodu X do bodu Y. Ak by sme však takýmto spôsobom chceli riešiť mechanizmus pre odporúčacie systémy, ktoré sú implementované napríklad u Amazonu, Youtubeu či Netflixu, boli by potrebné modifikácie algoritmov pre každého individuálneho používateľa zvlášť, a to by zrovna nebolo reálne riešenie. Jediným spôsobom ako predísť problému takéhoto typu je zostrojiť systém, ktorý je schopný sa prispôbovať novým podmienkam a učiť sa samostatne. Je ťažkopádne považovať systém za inteligentný bez možnosti učenia sa, preto je strojové učenie súčasťou umelej inteligencie.

Jednou z úloh umelej inteligencie je napodobňovať aj ľudskú inteligenciu. Aby sme sa k tomu približovali v čo najbližšej možnej podobe, bolo by vhodné sa inšpirovať práve aj ľudským „hardvérom“, mozgom. Mnoho vedcov verí, že prostredníctvom neurónových sietí, napodobňujúce spôsob fungovania mozgu sa dosiahnu najlepšie možné výsledky. V súčasnosti patrí medzi najpresnejšie metódy riešenia problémov v strojovom učení. Neurónové siete boli vynájdené už dávnejšie, no jej potenciál sa ukázal až neskôr pri výkonnejšej výpočtovej sile a internetu, cez ktorý bolo možné čerpať veľké množstvá dát. Boli to práve dáta, ktoré prinášali možnosť zvyšovať zložitosť procesov prostredníctvom hlbokých neurónových sietí. Mnohé známe

spoločnosti ako Microsoft a Google investujú do výskumu v tejto oblasti, o ktorej sa hovorí Deep learning. [1]

Umelá inteligencia čoraz viac napreduje a je obohacovaná stále novými prístupmi k jej realizácii. Vhodným zhrnutím historických pokrokov vyjadruje nasledujúci obrázok od NVIDIA.



Obrázok 1.1: História umelej inteligencie od NVIDIA [2]

Čo je to strojové učenie? Podobne ako je to pri umelej inteligencii, ťažko je podať výstižnú definíciu strojového učenia. Opäť sa môžu vyskytnúť mylné predstavy o učení tak ako je to s inteligenciou. Pre jednoduchosť pod pojmom učenie v kontexte strojov si môžeme predstaviť adaptívny prístup riešenia úloh v tom zmysle, že ich systém vyrieši po každom pokuse efektívnejšie. Takémuto učeniu zodpovedá aj jedna zo známych definícií strojového učenia od Arthura Samuela. Jeho myšlienka spočíva v tom, že strojové učenie sa zaoberá metódami učenia sa programov tak, aby vedeli adekvátne odpovedať na nové vstupy dát bez toho, aby na ne boli explicitne naprogramované.



V dnešnom svete sú dáta jedným z najdôležitejších zdrojov informácií. Je ich však také veľké množstvo, že ich nieje možné spracovávať manuálne alebo bežnými štatistickými metódami. Bývajú vytvorené v biznisových oblastiach ako je marketing, obchod, financie alebo aj pri výskume v medicíne a biológii. Všetky tieto dáta je možné analyzovať a objavovať v nich skryté vlastnosti a vzťahy interpretujúce nové znalosti. V medicíne alebo biológii prostredníctvom elektronických dát napríklad môžeme objavovať skryté stránky rakovín či génov pre ich lepšie porozumenie. Pre získavanie nových znalostí z dostupných dát vznikla nová interdisciplinárna oblasť zvaná objavovanie znalostí alebo Data mining. Používa najmä prístupy strojového učenia, no dopomôcť môžu aj databázové systémy, štatistika a prípadne na to stačia jednoduché vizualizačné techniky. [1]



# Kapitola 2

## Analýza problému

### 2.1 Obraz o stave riešenia daného problému

Jeden z najpodstatnejších vplyvných udalostí vzdelávania nastal práve aj pri vzniku internetu. Poznatky sa stali ľahko šíriteľné a dostupné všade tam, kde by inak prístupné neboli. Bývajú sprostredkované na webových stránok najmä prostredníctvom videí a dokumentov, ktoré je možné rozšíriť aj novými inovatívnejšími metódami akými je napríklad interaktivita s vizualizáciou. Mnoho z týchto metód nieje možné použiť pri tradičnom vzdelávaní. Potenciál E-learningu v tomto smere stále rastie a stáva lacnejšou, dostupnejšou a často krát aj efektívnejšou metódou pre vzdelávanie.

Dneska je množstvo materiálov aj o strojovom učení a väčšina z nich je spracovávaných prostredníctvom internetových článkov, kurzov či tutoriálov. Kurzy sa narozdiel od tutoriálov snažia pokryť širší záber z danej oblasti, pričom tutoriály sa dajú chápať ako jeho súčasti, ktoré poskytujú návod len k jednej podoblasti. Najvzornejším a v súčasnosti najlepšie ohodnocovaným príkladom internetového kurzu strojového učenia je od spoluzakladateľa vzdelávacej spoločnosti Coursera Andrew Ng, ktorý zachytáva všetky fundamentálne témy strojového učenia. Jeho dlhoročné skúsenosti ako dodatočný profesor strojového učenia priniesli kvalitný výklad prostredníctvom videí, ktoré boli obohacované aj intuitívnymi vizualizáciami. Výklad však vo všeobecnosti nieje jedinou metrikou pre meranie efektivity internetových materiálov. V prípade tutoriálov pre strojové učenie by bolo potrebné dbať na niekoľko metrík:

- použitie relevantného obsahu pre pochopenie hlavného konceptu, pomocou ktorého bude následne používateľ schopný implementovať algoritmus podľa svojho problému.
- presnosť, zrozumiteľnosť a jednoduchosť výkladu aj prostredníctvom prezentačných pomôcok akými môžu byť napríklad aj interaktívne vizualizácie.

- Vhodné použitie štandardných nástrojov známych na implementáciu strojového učenia pre príslušný problém. Pod nástrojmi môžeme napríklad chápať programovacie jazyky a knižnice. [3]

Metriky nám však samotne nemusia byť k úžitku pokiaľ ich nemáme ako merať. Na sledovanie týchto metrík je preto treba zaviesť aj niekoľko testovacích metód. Nasledovné testovacie metódy sa vzťahujú viac na kurzy ako na tutoriály, no dajú sa použiť aj pri tutoriáloch pokiaľ ich rozsah je dostatočne veľký.

- **čas strávený pri úlohách** - vyskytujú sa pri úlohách problémy pri ktorých sú používatelia zdržaný neadekvátne dlho? Je úloha príliš ťažká a vyžaduje si väčšiu pozornosť a podporu?
- **čas absolvovania tutoriálu** - vyžaduje tutoriál príliš dlhý čas na jeho dokončenie? Je celkovo príliš obtiažny či ľahký?
- **miera úspešnosti tutoriálu** - koľko používateľov dokáže dokončiť tutoriál do konca?
- **miera úspešnosti úloh** - ktoré úlohy niesú používateľmi dokončené a prečo?
- **miesta odchodov používateľov** - v ktorých častiach tutoriálu odchádza väčšina používateľov? [4]

V strojovom učení sa vyvíjajú stále nové netriviálne metódy pri ktorých nieje vždy intuitívne jasné ako pri nich fungujú použité algoritmy vo vnútri. Preto sa na zjednodušenie takýchto typov problémov manipulujúce s veľkým množstvom dát používajú vizualizačné techniky. Existujú rôzne konfigurácie či parametre algoritmov spolu aj s rôznymi dátami a nechať možnosť s nimi experimentovať používateľa pri chápaní kľúčových konceptov cez interaktívne vizualizácie zvyšuje efektivitu a pozitívnu skúsenosť s tutoriálom. Mnoho tutoriálov neponúka takúto možnosť zrejme kvôli tomu, že to vyžaduje množstvo práce s nástrojmi, ktoré niesú špecificky vytvárané na tento problém. Časom však strojové učenie nachádzalo stále väčšiu dôležitosť aj v schopnostiach explanácie, preto začali vznikať knižnice a frameworky špeciálne zamerané na túto problematiku.

Potenciál E-learningu preto boduje práve aj pri explanačnej sile. Poskytuje predovšetkým slobodu miesta, času a zastavovania sa podľa potreby myšlienkových pochodov v procese učenia. Jednou z ďalších nápadov na zvýšenie efektivity by mohol byť aj prostredníctvom spracovania prirodzeného jazyka. Pri riešení práce ma k výbere algoritmov viedla práve táto oblasť.

## 2.2 Zdôvodnenie voľby spôsobu riešenia a jeho stručný opis

Strojové učenie je veľmi široká oblasť ktorá okrem toho neustále napreduje a je ťažko obsiahnuť všetky jej témy do hĺbky. V rozsahu tutoriálu dokážeme však objasniť jednu zo zvolených metód spolu s konkrétnymi ukázkovými algoritmami. Zvolenie si témy učenia sa s učiteľom mne osobne prišlo predstaviteľnejšie pre ľudí, ktorí sa s touto témou prvý krát stretávajú a je v nich jasnejšie udávať konkrétne príklady použitia. Sú najčastejšie rozpoznávané problémy vo svete ktoré sú riešiteľné tradičnými metódami strojového učenia. Podobným spôsobom padlo rozhodnutie medzi klasifikáciou a regresiou. Napríklad prístupy problémov rozpoznávania obrázkov či textu pre mňa v kontexte umelej inteligencie vyvolávalo väčší pocit inteligencie pri usudzovaní záveru medzi konkrétnymi triedami sveta, resp. diskretnými hodnotami, pričom pri predikciách reálnych čísiel v regresii môže prísť zdanie, prikláňajúce sa viac k matematickým prístupom riešenia.

Klasifikácia má jedno z najširších využití v strojom učení. Príkladom toho môže byť počítačové videnie, spracovanie prirodzeného jazyka, odporúčacie systémy či internetové vyhľadávače. Práca teda bude venovať najväčšiu pozornosť klasifikácii a bude poskytovať interaktívne vizualizácie pre algoritmus Naívneho Bayesa. Zvykne sa používať pri spracovávaní prirodzeného jazyka a práve aj kvôli tomu sa budú klasifikovať textové dáta. Algoritmus je jednoduchý na pochopenie aj pre ľudí ktorí nemajú žiadnu predstavu o strojovom učení. Môže byť preto pútavou metódou aj pre laikov, naopak pre náročnejších bude taktiež k dispozícii algoritmus podporných vektorov alebo SVM (Support Vector Machines). Je to jeden z najpopulárnejších učiacich sa algoritmov v strojovom učení a pred neurónovými sieťami bol pre svoju presnosť považovaný za najväčší boom. Obe algoritmy sú vhodnými kandidátmi pre textovú klasifikáciu a porovnanie ich výsledkov nám pomôže rozhodnúť kedy ktorý použiť. Tieto algoritmy bude možné cez tutoriál postupne implementovať popri praktických cvičeniach.

Ako nástroj pre realizáciu cvičení sa v rámci tutoriálu použije interaktívne prostredie jupyter, umožňujúci používateľom editovať kódy autorov priamo v internetovom prehliadači a spúšťať ich spolu aj s vizualizáciami, do ktorých je taktiež možné zasahovať napríklad pridávaním nových dát do grafu. Prípadne pre nedostatočný rozsah cvičných úloh budú okrem dvoch spomenutých

klasifikačných algoritmov pridané ďalšie štandardne známe algoritmy pre strojové učenie ktorými sú napríklad lineárna či logistická regresia.

V jupyteri sa bude používať Python, ktorý je jeden z najpoužívanějších programovacích jazkov v strojovom učení. Je vhodnou voľbou najmä pre množstvo dostupných knižníc, ktoré flexibilne používajú nástroje pre textovú klasifikáciu. Na vizualizácie sa použije knižnica matplotlib, ktorá bude v spolupráci s jupyterom vykreslovať grafy pre väčšinu praktických úloh. Najväčší dôraz interaktívnych vizualizácií bude však kladený na Naivný Bayesovský klasifikátor, ktorý bude implementovaný detailnejšie pomocou nedávne vydaného frameworku Dash. Používa sa pri vytváraní analytických webových aplikácií a poskytuje bohaté možnosti interakčných prvkov ovplyvňujúce vizualizáciu. [5]

# Kapitola 3

## Opis riešenia návrhu

### 3.1 Návrh vizualizácii Naivného Bayesa

Nasledovné návrhy vizualizácií budú vychádzať z výpočtov pravdepodobností podľa Bayesovho pravidla. Pri každom slove sa vypočíta pravdepodobnosť vynásobením likelihoodu s priorom.

**likelihood** - počet vyskytujúcich sa príslušných slov v kategórii trénovacej množiny vydelenú celkovým počtom slov kategórie v trénovacej množine

**prior** - počet vzoriek príslušnej kategórie

Kedže Naivný Bayes vychádza z nezávislých atribútov, každý výpočet pravdepodobnosti slova sa považuje ako nezávislý od ostatných. V jednej vzorke sa teda bude iterovať cez všetky slová, ktoré sa budú navzájom vynásobovať ich vypočítanou pravdepodobnosťou.

Na vizualizáciu týchto hodnôt sa použijú 3 rôzne kategórie, pričom každá bude vyjadrovať jeden rozmer pravdepodobnosti. Vznikne nám tak pohľad 3D kocky na ktorú je možné mapovať tieto vypočítané hodnoty. Dĺžka strán kocky bude určená maximálnou pravdepodobnosťou predikcie. To bude mať za následok mapovanie pravdepodobností na jednej vonkajšej strane kocky. Kedže sa použijú 3 kategórie, dostaneme tak tri 2D obrazy na ktoré je možné sa pozeriť z jedného pohľadu na 3D kocku. Takouto paralelnou projekciou dostaneme zredukovaný výsledný pohľad z 3D na 2D.

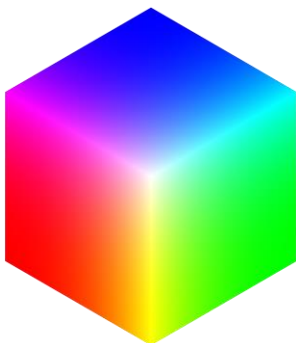
Problém však nastáva pri nastavovaní minimálnej pravdepodobnosti. Nula sa dá získať len cez vynásobenie prípadu nevyskytujúceho sa slova, takých prípadov je však veľká väčšina a mala by za následok veľmi nepresnú výslednú predikciu. Preto je pre takéto prípady potrebné zadať veľmi malé číslo, ktoré bude zaručovať nenulový konečný výsledok. Takéto číslo môže byť zadané používateľom alebo môže byť vypočítané napr. prostredníctvom Laplace Smoothingu.

Pri týchto minimálnych hodnotách sa však dostávame k ďalšiemu problému s rozsahom mapovania. Použijem svoj ilustračný príklad pri ktorom som dostal nasledovné hodnoty (po všetkých iteráciách vzorky) ktoré chcem namapovať do rozsahu čísiel od 0 po 255 (pravdepodobnosti sú záporné kvôli logaritmovaniu extrémne malých hodnôt pri násobení):

Pravdepodobnosť(P) kategórií	Mapovaná hodnota	Rozdiel P od maxima	Rozdiel P od minima
-1467 (Max. P)	255	0	1418
-1619	227	151	1266
-1604	230	165	1281
Min. P = -2885	0	1418	0

Mapované hodnoty dvoch kategórií s nemaximálnou pravdepodobnosťou budú príliš blízko maxima a to na vizualizácii spôsobí v týchto oblastiach veľkú hustotu pri väčšom množstve podobných vzoriek. Čím je väčší rozdiel medzi maximom a minimom pri kategóriách s nemaximálnou pravdepodobnosťou, tým sa stáva rozdiel irelevantnejším na mapovanej hodnote. Tento problém sa môže najčastejšie vyskytovať pri vzorkách s väčším množstvom slov a kategórie si niesú ďaleko príbuzné.

Jednou z možných riešení by bolo vyhľadať najmenšiu mapovanú hodnotu zo všetkých iterácií, ktorá by sa vybrala ako všeobecná minimálna pravdepodobnosť rovnajúca sa mapovanej hodnote nula, podľa ktorej by sa následne vytváral rozsah. Zjednodušene povedané, to by malo za následok taký zoom, pri ktorom budeme stále vidieť všetky dátové vzorky pri 2D pohľade (aj pri tomto prístupe sa môžu vyskytnúť problémy, ktoré už ale nebudem uvádzať).

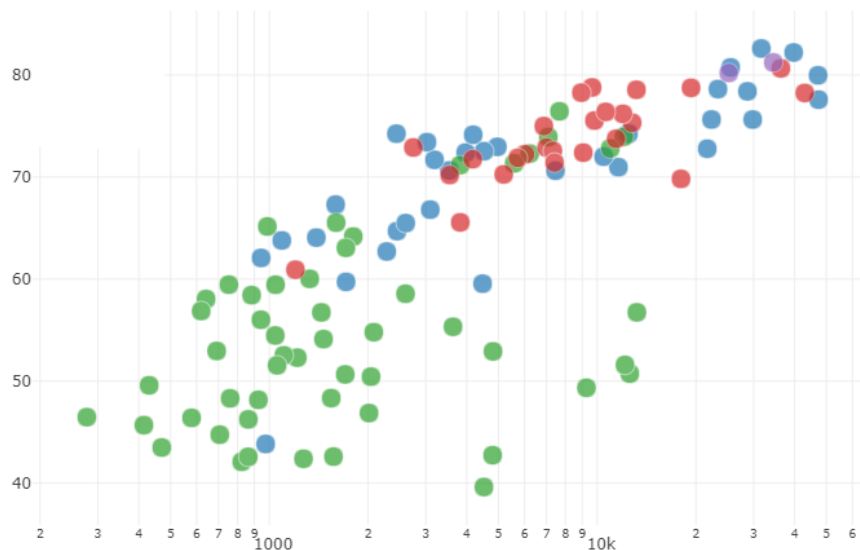


Výber mapovaných hodnôt je inšpirovaný RGB farbami, cez ktoré je možné ku každej predikcii priradiť konkrétnu farbu. Každá farba predstavuje jednu kategóriu a pri jej maximálnej pravdepodobnosti poskytuje jeden 2D obraz strany pozostávajúci z dvoch zvyšných farieb.

Obrázok 3.1: Kocka určujúca výchylky predikcií cez RGB hodnoty [6]

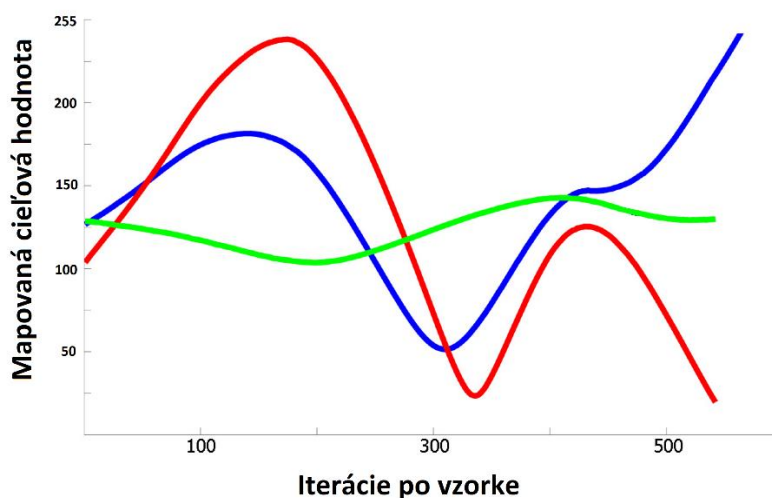


Funkcionalita zoškálovať tri obrázky do jedného sa však pravdepodobne nenájde vo vizualizačnom nástroji a alternatívny spôsob návrhu by mohol byť cez jednoduché zlúčenie všetkých obrázkov do jedného, pričom by vzorky obsahovali len svoju príslušnú farbu kategórie.



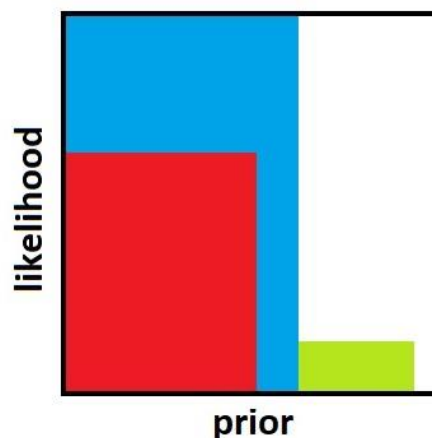
Obrázok 3.2: Zlúčené 2D obrázky vzoriek do jednej [7]

Takýmto spôsobom môžeme mať cez tento graf predstavu s akou istotou boli všetky vzorky predikované. Okrem výsledných predikcií nás môže zaujímať aj ich výpočtový priebeh pravdepodobností, ktorý by mohol byť vyjadrený nasledujúcim grafom obsahujúci mapované cieľové hodnoty.



Obrázok 3.3: Výpočtový priebeh pravdepodobností [8]

Ak sa chceme pozrieť ešte hlbšie do výpočtov, môžu nás zaujať konkrétne hodnoty likelihoodu a prioru. Keďže sú to dve hodnoty ktoré sa násobia, môžeme si ich predstaviť ako štvoruholníky v ďalšom 2D grafe, pričom hodnoty priorov budú korešpondovať s x-ovou osou a likelihood s y-ovou osou. Opäť môžeme sledovať priebeh týchto hodnôt cez nasledujúci obrázok. Pre ilustráciu aký vplyv majú priory na predikovaný výsledok môžeme zaviesť aj možnosť nebrania ich do úvahy.



Obrázok 3.4: Výpočet pravdepodobnosti

Podobne ako vizualizujeme všetky predikované vzorky v grafe, môžeme vizualizovať všetky slová v jednej vzorke. Dostaneme tak predstavu o každom slove aký mal konkrétny vplyv na výsledok cez miesto, na ktorom sa nachádza. Priemerom pozícií všetkých slov v grafe vzorky by sme dostali ťažisko vyjadrujúce aktuálny stav pravdepodobnostného výpočtu a mohli vykresliť priebeh v tomto grafe aj prostredníctvom čiar.

Pri tejto vizualizácii je možné spozorovať vplyvy jednotlivých slov vo vzorke. Napríklad najväčší rozdiel medzi maximálnou výslednou hodnotou v jednej kategórii a maximálnou druhou v druhej kategórii v k-tej iterácii nám vypovedá o najvplyvnejšom slove ktoré je na k-tom mieste. Analogickým spôsobom by sme vedeli nájsť aj druhé či tretie najvplyvnejšie slovo vo vzorke, prípadne by bolo možné vytvoriť celý histogram s percentuálnymi vplyvmi predikcie pri každom z nich. Používateľovi by bolo umožnené nastaviť dĺžku histogramu či percentuálnu hranicu vplyvu. Toto nám môže taktiež prísť vhod pri nesprávne klasifikovaných vzoriek, pri ktorých sa môžeme pozrieť ktoré slová najviac smerovali ku chybnnej predikcii. Naopak môžeme zdetekovať slová, ktoré nemali žiadny vplyv vo výslednej predikcii a môžu byť vhodným kandidátom na zahodenie tak, že sa priradia do množiny stop-slov, ktoré sa pri vypočítavaní vynechávajú.

K vizualizáciám budú použité najmenej dve rôzne datasety ktoré budú vypovedať o rozličnej obtiažnosti klasifikácie. Pod ľahším datasetom si môžeme predstaviť kategorizáciu medzi

odvetviami jedla, elektroniky, drogérie a pod. Obtiažnejšou alternatívou môžu byť kategórie s podobnou terminológiou slov napríklad pri medicíne, kde sa rozhoduje o akú diagnostiku sa jedná. Pri každom datasete bude možné určiť percentuálny pomer testovacích a trénovacích príkladov spolu aj s nastavením ľubovoľného počtu vzoriek pri každej kategórii zvlášť prostredníctvom početnostného grafu.



# Literatúra

- [1] NÁVRAT, P. -- BIELIKOVÁ, M. -- BEŇUŠKOVÁ, Ľ. -- KAPUSTÍK, I. -- UNGER, M. *Umelá inteligencia*. Bratislava : STU v Bratislave, 2002. 393 s. ISBN 80-227-1645-6.
- [2] Michael Copeland. “What’s the Difference Between Artificial Intelligence, Machine Learning, and Deep Learning?” July 2016. URL: <https://blogs.nvidia.com/blog/2016/07/29/whats-difference-artificial-intelligence-machine-learning-deep-learning-ai/>
- [3] Nikos Andriotis. “How to Use Training Metrics to Measure eLearning Effectiveness”. Nov. 2017. URL: <https://www.efrontlearning.com/blog/2017/11/how-use-training-metrics-measure-effectiveness.html>
- [4] David Venturi. “Every single Machine Learning course on the internet, ranked by your reviews”. May 2017. URL: <https://medium.freecodecamp.org/every-single-machine-learning-course-on-the-internet-ranked-by-your-reviews-3c4a7b8026c0>
- [5] Stephenkhky. “R or Python on Text Mining”. Aug. 2015. URL: <https://datawarrior.wordpress.com/2015/08/12/codienerd-1-r-or-python-on-text-mining/>
- [6] [https://commons.wikimedia.org/wiki/File:RGB\\_Colorcube\\_Corner\\_White.png](https://commons.wikimedia.org/wiki/File:RGB_Colorcube_Corner_White.png)
- [7] <https://plot.ly/dash/getting-started>
- [8] [http://www.pracsyslab.org/curvilinear\\_formation](http://www.pracsyslab.org/curvilinear_formation)