**SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I INFORMACIJSKIH TEHNOLOGIJA**

**Sveučilišni diplomski studij računarstva**

**Informacijske i podatkovne znanosti, DRD**

**PREDIKCIJA BROJA INSTALACIJA APLIKACIJE NA GOOGLE PLAY STORE-u**

**Računarstvo usluga i analiza podataka**

**Matej Dmitrović**

**Osijek, 2020.**

Contents

[1. UVOD 1](#_Toc49358580)

[2. ANALIZA PODATAKA 2](#_Toc49358581)

[2.1 Skup podataka 2](#_Toc49358582)

[2.2 Azure Machine Learning Classic 2](#_Toc49358583)

[2.2.1 Priprema podataka 2](#_Toc49358584)

[2.2.2 Treniranje modela 2](#_Toc49358585)

[2.2.3 Azure Web service 2](#_Toc49358586)

[3. PROGRAMSKO RJEŠENJE 3](#_Toc49358587)

[3.1 Azure Machine Learning Classic 3](#_Toc49358588)

[3.2 HTML/CSS 3](#_Toc49358589)

[3.3 Python/Flask i jQuery 3](#_Toc49358590)

[3.3.1 „Denormalizacija“ 4](#_Toc49358591)

[3.3.2 API 4](#_Toc49358592)

[4. ZAKLJUČAK 5](#_Toc49358593)

[5. POVEZNICE I LITERATURA 6](#_Toc49358594)

# 1. UVOD

# 2. ANALIZA PODATAKA

Kako bi bilo moguće učenje nad velikom količinom podataka, traženi podaci moraju biti dobro formatirani i uređeni kako učenje nad njima bilo što kvalitetnije.

## 2.1 Skup podataka

Skup podataka nad kojem je projekt rađen dohvaćen je sa stranice Kaggle koja drži repozitorij veliku količinu različitih skupova podataka. Odabran je skup podataka „Google Play Store“ koji u sebi sadrži mnoštvo aplikacija sa (nekim od) njihovih parametara. Na slici 2.1. prikazani su svi stupci koji su došli sa na obrađenim skupom podataka.

* *App* – ime aplikacije
* *Category* – kategorija aplikacije
* *Rating* – prosješna ocjena aplikacije
* *Reviews* – broj recenzija za aplikaciju
* *Size* – veličina aplikacije (MB)
* *Installs* – broj instalacija aplikacije
* *Type* – je li aplikacija besplatna ili ne
* *Price* – cijena aplikacije
* *Content Rating* – ocjena sadržaja aplikacije po dobi
* *Genres* – isko kao *Category*
* *Last Updated* – datum zadnjeg ažuriranja aplikacije
* *Current Ver* – trenutna verzija aplikacije
* *Android Ver* – verzija Androida za koju je aplikacija napravljena



Slika 2.1. Stupci korišteni u skupu podataka „Google Play Store“

## 2.2 Azure Machine Learning Classic

### 2.2.1 Priprema podataka

### 2.2.2 Treniranje modela

### 2.2.3 Azure Web service

# 3. PROGRAMSKO RJEŠENJE

## 3.1 Azure Machine Learning Classic

Azure ML unutar svog web servsa nudi primjer pozivanja API-ja kao i detalje kako se API korsiti. Za dovaćanje predikcija je korišten Python programski jezik koji je uklopnjen u sklop web-stranice pomoću Flask framework-a.

## 3.2 HTML/CSS

Projekt se sastoji od jedna HTML stranice u kojoj se nalazi forma za upisivanje parametara predikcije. Forma se sastoji od ulaza za:

* Ime aplikacije – tekstualno polje za ime aplikacije
* Kategorija – padajući izbornik za odabir kategorija aplikacije. Moguće opcijuesu dohvaćene sa API-ja Azure web servisa
* Prosječna ocjena – numeričko polje za prosječnu ocjenu recenzija aplikacije, ukoliko je aplikacija već na Google Play Store-u
* Broj recenzija – numeričko polje za broj recenzija aplikacije, ukoliko je aplikacija već na Google Play Store-u
* Cijena – numeričko polje za cijenu aplikacije
* Ocjena sadržaja – padajući izbornik za odabir ocjenu sadržaja po dobi. Moguće opcije su dohvaćane sa API-ja Azure web servisa

„Kategorija“ i „Ocjena sadržaja“ su dinamički popunjeni pomoću jQuery-a kako bi se stranica mogla lakše ažurirati ako dođe do promjena vrijednosti koja ta polja mogu poprimiti.

## 3.3 Python/Flask i jQuery

Dinamični dio web-stranice napravljen je pomoću Flask framework-a za Python i jQuery-a. Flask održava organizaciju web-stranica i API poziva te ima mogućnost za podizanje vritualnog servera kako bi stranica mogla pristupiti vanjskim podacima bez podizanja stranice na pravi server. Temelji se na Python programskom jeziku, te je jedan od razloga korištenja Python-a za dohvaćanja predikacije sa Azure web servisa. Cijela dinamičnost web stranice se temelji na tome da korisnik upisuje ulaze u polja za formu, te klikom na gumb submit poziva se jQuery funkcija koja šalje podatke forme podatke na API ove web-stranice. Flask tada preuzima podatke i šalje ih u Python kod za slanje podataka Azure web servisu. Ukoliko ne dođe odgovor ili je došlo do greške, Python će vratiti grešku u obliku stringa. Ako je sve u redu i predikcija je bila uspješna, Python će formatirati i „denormalizirati“ podatke te ih postali natrag jQuery-u. jQuery tada predočuje dobivenu predikciju ili grešku na određeno mjestu unutar web-stranice.

### 3.3.1 „Denormalizacija“

U ovom pod-poglavlju spomenut je pojam „denormalizacije“ – pretvorbe dobivene predikcije natrag u odgovarajući interval vrijednosti. Predikcija kao rezultat normalizacije ostalih elemnata prilikom strojnog učenja dolazi u intervalu [0, 1]. Taj interval nije dobar pokazivač broja instalacija neke aplikacije. Budući da je u trening dijelu strojnog učenja provedena LogNormal normalizacija stupca „Installs“, predikcija se mora provesti kroz inverznu LogNormal funkciju. Azure ML ne prikazuje detalje normalizacija stoga je inverzija LogNormal normalizacije teško. U tom slučaju koristi se aproksimacija inverzne funkcije koja se temelji na empirijskim vrijednostima. Dobivena aproksimacija funkcija ima izraz

gdje je x dobivena vrijednost predikcije a y konkretan broj predikcije instalacija aplikacije.

### 3.3.2 API

# 4. ZAKLJUČAK

# 5. POVEZNICE I LITERATURA

Programsko rješenje zajedno sa ovom dokumentacijom se nalazi na udaljenom GitHub repozitoriju koji se može pristupiti preko linka: <https://github.com/Matt-Tyrantin/RUAP-project>