Predikcija i rangiranje uspjeha studenata na veleučilištu na osnovu rezultata državne mature, uspjeha iz srednje škole i drugih pokazatelja

Rijetković, Mario

Undergraduate thesis / Završni rad

2023

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **Polytechnic of Međimurje in Čakovec / Međimursko veleučilište u Čakovcu**

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:110:276384

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2024-05-19**

Repository / Repozitorij:

Polytechnic of Međimurje in Čakovec Repository - Polytechnic of Međimurje Undergraduate and Graduate Theses Repository

MEĐIMURSKO VELEUČILIŠTE U ČAKOVCU
STRUČNI STUDIJ RAČUNARSTVA

MARIO RIJETKOVIĆ

PREDIKCIJA I RANGIRANJE USPJEHA STUDENATA NA VELEUČILIŠTU NA OSNOVI REZULTATA DRŽAVNE MATURE, USPJEHA IZ SREDNJE ŠKOLE I DRUGIH POKAZATELJA

ZAVRŠNI RAD

ČAKOVEC, 2023.

MEĐIMURSKO VELEUČILIŠTE U ČAKOVCU
STRUČNI STUDIJ RAČUNARSTVA

MARIO RIJETKOVIĆ

PREDIKCIJA I RANGIRANJE USPJEHA STUDENATA NA VELEUČILIŠTU NA OSNOVI REZULTATA DRŽAVNE MATURE, USPJEHA IZ SREDNJE ŠKOLE I DRUGIH POKAZATELJA
PREDICTION AND RANKING OF UNIVERSITY STUDENTS BY SUCCESS BASED ON RESULTS OF STATE SECONDARY SCHOOL LEAVING EXAMINATION, MIDDLE SCHOOL SUCCESS AND OTHER FACTORS

ZAVRŠNI RAD

Mentor:
mr. sc. Željko Knok

ČAKOVEC, 2023.

MEĐIMURSKO VELEUČILIŠTE U ČAKOVCU
ODBOR ZA ZAVRŠNI RAD

Čakovec, 27. ožujka 2020

država: **Republika Hrvatska**
Predmet: **Baze podataka I - izvanredni**

ZAVRŠNI ZADATAK br. 2019-RAČ-I-192

Pristupnik **Mario Rijetković (0313019105)**
Studij: izvanredni preddiplomski stručni studij Računarstvo
Smjer: Programsko inženjerstvo

Zadatak: **Predikcija i rangiranje uspjeha studenata na veleučilištu na osnovu rezultata državne mature, uspjeha iz srednje škole i drugih pokazatelja**

Opis zadatka:

Na osnovu usporedbe rezultata ispita DM, ocjena iz srednje škole, ocjena iz odabranih predmeta na prvoj godini studija, predviđa se uspjeh u nadolazećim godinama. Koristiti alate otvorenog koda.

Zadatak uručen pristupniku: 27. ožujka 2020.
Rok za predaju rada: 20. rujna 2020.

Mentor:

mr. sc. Željko Knok, v. pred.

Predsjednik povjerenstva za završni ispit:

ZAHVALA

Zahvaljujem se svome mentoru mr. sc. Željku Knok na usmjeravanju i vođenju prilikom pisanja. Svojim mi je savjetima omogućio da temu ovog završnog rada sročim u strukturiranu cjelinu.

Također, zahvaljujem se svim djelatnicima Međimurskog veleučilišta u Čakovcu koji su nam prenijeli znanja unutar struke. Posebno se želim zahvaliti svojoj obitelji koja me podupirala tijekom studiranja u drugom gradu i omogućila mi sva novostečena poznanstva i znanja.

SAŽETAK

Tema ovog završnog rada zahtijeva stvaranje baze podataka s ocjenama iz srednje škole te s državne mature kako bismo mogli na temelju tih ocjena napraviti predikciju uspjeha na veleučilištu. Da bismo napravili predikciju kao web-aplikaciju, potrebno je koristiti program Visual Studio Code. Podaci koji su korišteni u ovome radu nasumično su generirani te je generirano 5000 slučajeva. Kako bi predikcija bila uspješna, potrebne su određene karakteristike za pojedini slučaj, ocjene iz srednje škole te ostvareni bodovi na državnoj maturi. Na temelju ocjena iz srednje škole i ostvarenih bodova na državnoj maturi aplikacija će predvidjeti uspjeh na veleučilištu za sve tri godine. Sama predikcija napravljena je pomoću TensorFlow biblioteke koja radi kao neuronska mreža. Napravljena web-aplikacija sastoji se od generiranja baze slučajeva, treniranja modela za predikciju, same predikcije, učitavanja CSV datoteke te samostalnog upisa ocjena.

Ključne riječi: TensorFlow, Visual Studio Code, CSV, predikcija

Sadržaj

1. UVOD 5
1.1. Opis 5
1.2. Strojno učenje 5
2. CILJ RADA 6
3. SOFTWARE 7
3.1. Visual Studio Code 7
3.2. Google Chrome – web-preglednik 8
4. PROGRAMSKI JEZICI 9
4.1. HyperText Markup Language (HTML) 9
4.2. Cascading Style Sheets (CSS) 9
4.3. JavaScript (JS) 10
4.4. TypeScript (TS) 10
5. BIBLIOTEKE – MODULI 11
5.1. React 11
5.2. TensorFlow 11
6. RUNTIME 12
6.1. Node.js 12
7. ALAT ZA PREDIKCIJU 13
8. WEB-STRANICA ZA PREDIKCIJU 14
8.1. Generiranje baze slučajeva 15
8.2. Treniranje modela 25
8.3. Predikcija 30
8.4. Generiranje CSV datoteke za izvoz 32
8.5. Uvoz CSV datoteke 34
8.6. Samostalan unos ocjena 36
9. ANALIZA REZULTATA 40
10. ZAKLJUČAK 44
11. LITERATURA 45
12. PRILOZI 47
12.1. Popis slika 47
12.2. Popis kodova 48

1. UVOD

1.1. Opis

U ovom završnom radu bit će uspoređeni rezultati s ispita državne mature, ocjene iz srednje škole te pomoću prediktivnog modela i pomoći linearne regresije bit će predviđen uspjeh u budućim godinama studija. Za izradu rada korišten je razvojni alat Visual Studio Code, biblioteke TensorFlow za predikciju ocjena, React i komponenta razvojnog okruženja (engl. Component Framework).

1.2. Strojno učenje

Strojno učenje je grana istraživanja posvećena razumijevanju i izgradnji metoda koje „uče", to jest metoda koje iskorištavaju podatke za poboljšanje izvedbe na nekom skupu zadataka. Na njega se gleda kao na dio umjetne inteligencije. Algoritmi strojnog učenja izgrađuju model temeljen na uzorcima podataka, poznatim kao podaci o obuci, kako bismo donosili predviđanja ili odluke bez eksplicitnog programiranja za to. Algoritmi strojnog učenja koriste se u raznim aplikacijama, kao što su medicina, filtriranje e-pošte, prepoznavanje govora i računalni vid, gdje je teško ili neizvedivo razviti konvencionalne algoritme za obavljanje potrebnih zadataka. [1]

2. CILJ RADA

Cilj ovog rada je izrada aplikacije kojom će se predvidjeti uspjeh studenata na veleučilištu. Bit će uspoređeni rezultati s ispita državne mature, ocjene iz srednje škole te pomoću prediktivnog modela i pomoći linearne regresije bit će predviđen uspjeh tijekom budućih godina studija.

Za izradu rada korišten je razvojni alat Visual Studio Code, biblioteke TensorFlow za predikciju ocjena, React i komponenta razvojnog okruženja (engl. Component Framework). Krajnji cilj izrade aplikacije je prikaz prosjeka ocjena na veleučilištu kroz sve tri godine studija te prikaz najpovoljnijeg smjera za pojedini slučaj.

3. SOFTWARE

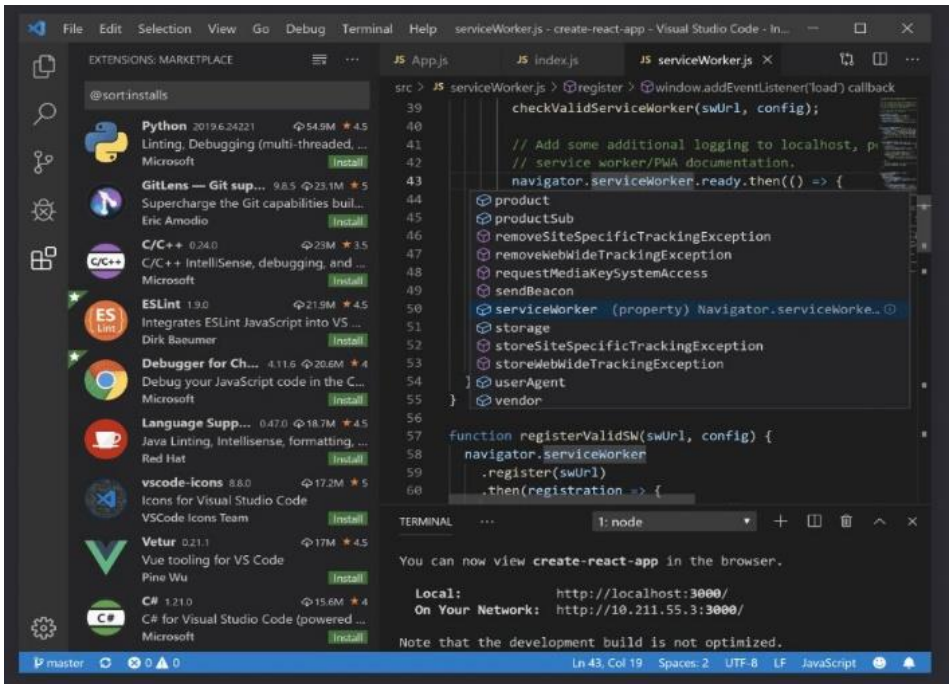
3.1. Visual Studio Code

Slika 1. Visual Studio Code – logo

Izvor: https://python-climbing.com/wp/wp-content/uploads/2019/10/vscode_icon-1024x576.png (23.6.2022.)

Visual Studio Code, koji se također naziva VS[1] Code, uređivač je koda koji je napravio Microsoft za Windows, Linux i macOS. Značajke VS Codea uključuju podršku za otklanjanje pogrešaka, isticanje sintakse, inteligentno dovršavanje koda, isječke, refaktoriranje koda i implementirani Git. Korisnici mogu promijeniti temu, tipkovničke prečace, postavke i instalirati proširenja koja dodaju dodatne funkcije. Neke od značajki koje nudi VS Code su velika podrška programskih jezika kao što su Java, JavaScript, Go, Node.js, Python, C++, C. VS Code je odličan alat koji nudi već unaprijed pomoć pri pisanju koda. Pomoći koje se koriste pri pisanju koda su: isticanje sintakse, podudaranje zagrada, preklapanje koda. Također je opremljen sa značajkom IntelliSense koja pomaže tako što dovršava kod, preporučuje naredbe, nudi brze informacije o parametrima. Dodatna mogućnost VS Coda je dodavanje proširenja putem centralnog spremišta koji pomaže većoj funkcionalnosti, efikasnosti i jezičnoj podršci. U prilog superiornosti ovog alata govore i istraživanja programera na web-stranici „Stack Overflow", prema kojem je 2016. godine VS Code na ljestvici popularnosti ostvario 13. mjesto, a dvije godine kasnije na istoj ljestvici zauzeo 1. mjesto. [3]

[1] Viusal Studio



Slika 2. Visual Studio Code – radno okruženje

Izvor: https://pobierz.pl/wp-content/uploads/2020/05/Screenshot_13.jpg (23.6.2022.)

3.2 Google Chrome – web-preglednik

Slika 3. Web-preglednik Google Chrome – logo

Izvor:https://s1.ibtimes.com/sites/www.ibtimes.com/files/styles/full/public/2020/08/26/chrome.jpg (03.7.2022.)

Google Chrome je višeplatformski web-preglednik koji je razvio Google 2008. godine. Početna verzija bila je namijenjena za platformu Microsoft Windows, koja je napravljena od besplatnih (Open Source) komponenata Apple Webkit i Mozilla Firefox. Kasnije je prenesena na platformu Linux, macOS, iOS i Android. Statistički podaci stranice „StatCounter" za 2021. godinu procjenjuju da Google Chrome koristi 68 % populacije, od čega 72 % na osobnim računalima. Zbog ovog uspjeha, Google je proširio naziv marke „Chrome" na druge proizvode poput ChromeOS, Chromecast, Chromebook, Chromebox, Chromebase. [7]

4. PROGRAMSKI JEZICI

4.1. HyperText Markup Language (HTML)

Slika 4. HTML 5 – logo

Izvor: https://www.taste-of-it.de/wp-content/uploads/2014/01/html5-logo.jpg (10.7.2022.)

HyperText Markup Language ili HTML[2] je standardni jezik za dizajniranje dokumenata koji se prikazuju na web-stranici. HTML moguće je uređivati pomoću tehnologije Cascading Style Sheets (CSS)[3] i skriptnih jezika poput JavaScript. Web-preglednik prima HTML dokument s web-poslužitelja ili lokalnog spremišta te ga pretvara u multimedijske web-stranice. HTML elementi izgrađuju izgled stranice. Neki od elemenata su dugme, forma, paragraf, naslov, slike te linkovi. Godine 1993. HTML se uvodi kao novi jezični standard za web-preglednike. Nakon prvog izdanja HTML jezika uslijedile su nadogradnje HTML+, HTML 2, HTML 3, HTML 4 te današnji HTML 5. [9]

4.2. Cascading Style Sheets (CSS)

Slika 5. CSS – logo

Izvor: http://blog.brakoniecki.pl/wp-content/uploads/2015/02/css-logo.png (15.7.2022.)

Cascading Style Sheets (CSS) je stilski jezik koji se koristi za uređivanje HTML dokumenta. CSS nam omogućuje promjenu, veličinu i boju fonta, pozicioniranje elemenata na web-stranici te izgled same stranice. CSS se prvi put pojavljuje 1996. godine pod nazivom CSS 1, nakon čega su uslijedile nadogradnje CSS 2, CSS 2.1, CSS 3 te današnji CSS 4. [10]

[2] HyperText Markup Language
[3] Cascading Style Sheets

4.3. JavaScript (JS)

Slika 6. JavaScript – logo

Izvor: https://quocent.com/QUOCENT-CMS/userfiles/StaticBanner/1529923467_Javascript.png (05.8.2022.)

JavaScript (JS)[4] je objektno orijentiran programski jezik. Jedan je od temeljnih tehnologija za razvoj web-stranica, uz HTML i CSS. JS je jezik visoke razine, pravodobno kompajliran te nudi dinamičko tipkanje. Uz to podržava i sučelje (API)[5] za rad s tekstom, datumima, standardnim strukturama podataka te Document Object Model (DOM).[6] Najpopularniji dodatak za JavaScript programski jezik je Node.js. Za povećanje funkcionalnosti JS-a postoje biblioteke kao što su jQuery koji je ujedno najpopularniji i najkorišteniji (75 %), React koji je napravio Facebook te Angular koji je izradio Google. JavaScript prvi put se pojavljuje 1995. godine. [13]

4.4. TypeScript (TS)

Slika 7. TypeScript – logo

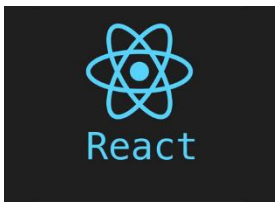
Izvor: https://humancoders-formations.s3.amazonaws.com/uploads/course/logo/230/formation-typescript.png (15.8.2022.)

TypeScript je programski jezik proizveden od tvrtke Microsoft. To je strogi sintaktički nadskup JavaScripta i dodaje izborno statičko tipkanje u jezik. Dizajniran je za razvoj velikih aplikacija i prevodi se u JavaScript. TypeScript podržava definicijske datoteke koje mogu sadržavati podatke o vrsti postojećih JavaScript biblioteka, slično kao C++. TypeScript je prvi put javno objavljen 2012. godine, nakon dvije godine internog razvoja u Microsoftu te su uslijedile nadogradnje do današnje TypeScript 4.0 verzije. [16]

[4] JavaScript
[5] Application programming interface
[6] Document object model

5. BIBLIOTEKE – MODULI

5.1. React



Slika 8. React – logo

Izvor: https://naomod.univ-nantes.io/idl/slides/images/react-logo.png (15.8.2022.)

React (također poznat kao React.js ili ReactJS) je besplatna front-end JavaScript biblioteka otvorenog koda za izgradnju korisničkog sučelja baziranog na UI[7] komponentama. React kod se sastoji od entiteta koji se nazivaju komponente. Komponente se mogu neograničeno koristiti, moraju se nalaziti u mapi src[8] i moraju slijediti Pascal Case kao svoju konvenciju imenovanja. React komponente obično su napisane pomoću JSX-a,[9] iako ne moraju biti (komponente mogu biti napisane i u čistom JavaScriptu). JSX ili JavaScript Syntax Extension je proširenje sintakse JavaScript jezika. Pruža strukturiranje i renderiranje komponenti korištenjem sintakse poznate mnogim programerima. Današnji React napravljen je kao prototip pod nazivom „FaxJS" 2011. godine, a aktualni naziv dobio je 2015. godine uz implementaciju podrške za Android, iOS i UWP[10] – Windows. [14]

5.2 TensorFlow

Slika 9. TensorFlow – logo

Izvor:https://www.w3cschool.cn/attachments/image/20170808/1502183953889061.png (7.9.2022.)

TensorFlow je besplatna softverska biblioteka otvorenog koda za strojno učenje i umjetnu inteligenciju. Ima široku upotrebu za rješavanje raznih zadataka, ali posebno je fokusiran na obuku i zaključivanje dubokih neuronskih mreža. TensorFlow ima širok spektar podrške za programske jezike. Najčešće se koristi uz programski jezik Python, JavaScript, C++ i Java. TensorFlow je razvio tim Google Brain tvrtke Google 2015. godine. [20]

[7] User interface
[8] Source
[9] JavaScript Syntax Extension
[10] Universal Windows Platform

6. RUNTIME

6.1. Node.js

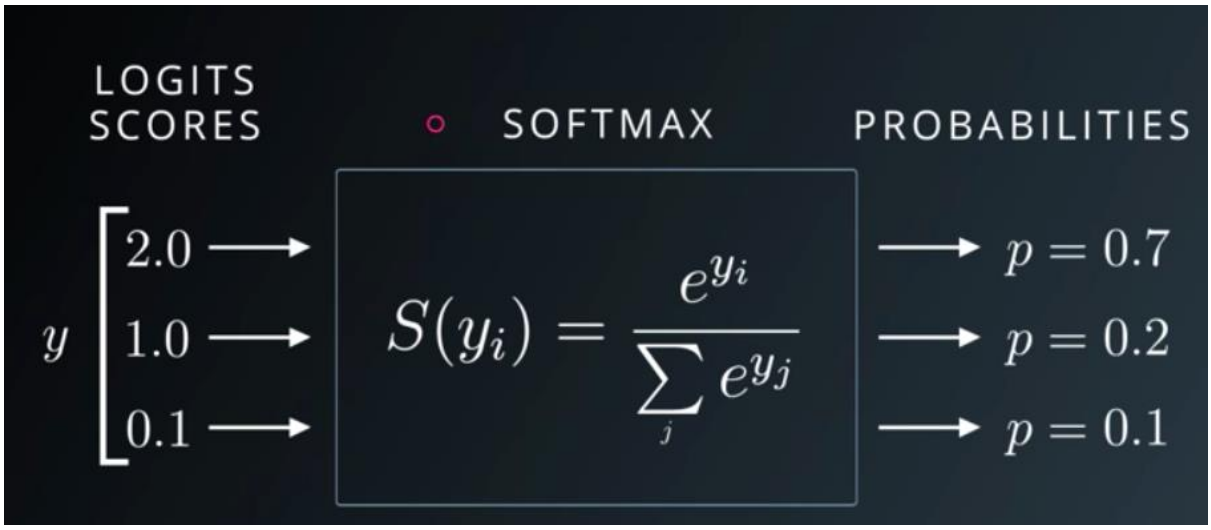
Slika 10. Node.js – logo

Izvor: https://logos-download.com/wp-content/uploads/2016/09/Node_logo_NodeJS.png (1.9.2022.)

Node.js je otvoren kod, cross-platform, back-end JavaScript runtime okruženje koje radi na V8 engineu i izvršava JavaScript kod izvan web-preglednika, koji je dizajniran za izgradnju skalabilnih mrežnih aplikacija. Node.js omogućuje razvojnim programerima korištenje JavaScripta za skriptiranje na strani poslužitelja kako bi se izradio dinamički sadržaj na web-stranici. Node.js razvijen je 2009. godine te je podržavao Linux i mac OS X. [19]

7. ALAT ZA PREDIKCIJU

U radu je korištena biblioteka TensorFlow. Pomoću nje predviđamo kakav uspjeh će biti ostvaren na osnovu ocjena iz srednje škole te ostvarenih bodova na državnoj maturi. Za predviđanje uspjeha koristi se Tensorova funkcija „softmax". Funkcija softmax, također poznata kao softargmax ili normalizirana eksponencijalna funkcija, pretvara vektor od K realnih brojeva u distribuciju vjerojatnosti od K mogućih ishoda. To je generalizacija logističke funkcije na više dimenzija i koristi se u multinomijalnoj logističkoj regresiji. Funkcija softmax često se koristi kao zadnja aktivacijska funkcija neuronske mreže za normalizaciju izlaza mreže na distribuciju vjerojatnosti preko predviđenih klasa izlaza, na temelju Luceovog aksioma izbora. [22]

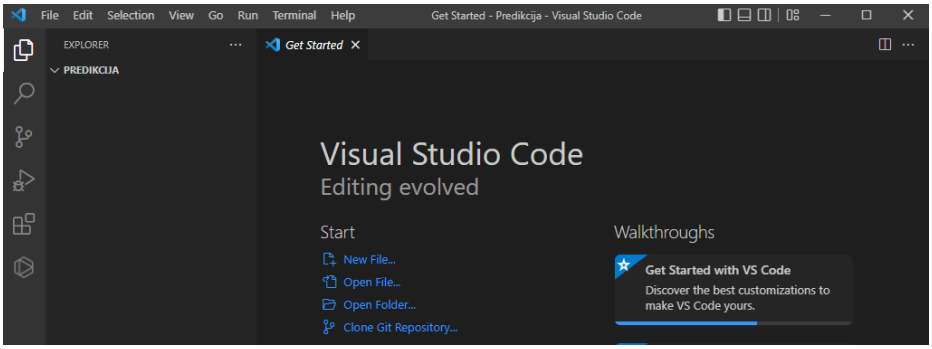


Slika 11. Funkcija softmax

Izvor: https://911weknow.com/wp-content/uploads/2020/09/understand-the-softmax-function-in-minutes-2.png (11.9.2022.)

8. WEB-STRANICA ZA PREDIKCIJU

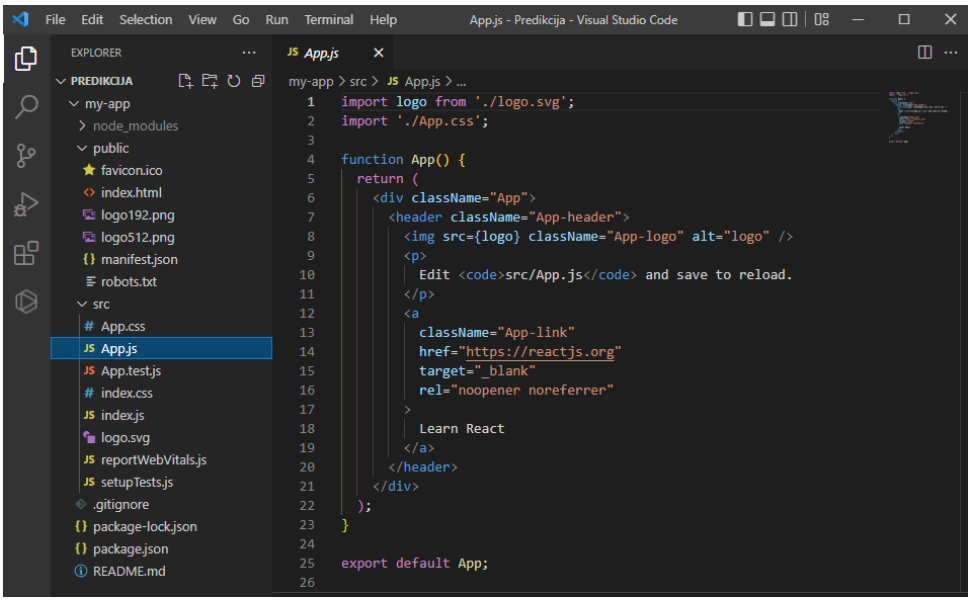
Prije početka izrade web-stranice potrebno je kreirati praznu mapu na računalu i instalirati NodeJs runtime te unutar mape desnim klikom otvoriti Visual Studio Code kao što vidimo na slici 12.



Slika 12. Kreirana mapa unutar Visual Studio Codea

Izvor : Autor

U sljedećem koraku potrebno je otvoriti terminal prozor unutar Visual Studio Codea i pokrenuti naredbu npx (create-react-app my-app --template typescript) za implementaciju React razvojnog okruženja s podrškom za programski jezik TypeScript. Nakon toga potrebno je pokrenuti naredbe za implementaciju TensorFlowa u Visual Studio Codeu (npm install @tensorflow/tfjs, i npm install @tensorflow/tfjs-vis). Po završetku instalacije web-aplikaciju pokrećemo putem naredbe (npm start).



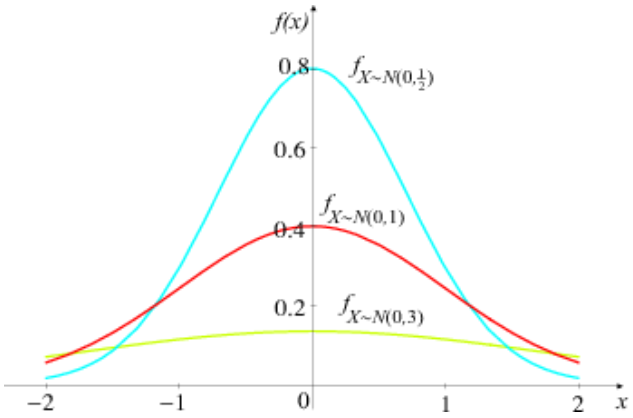
Slika 13. Prikaz implementiranog React modula

Izvor: Autor

Nakon implementacije TensorFlowa i React biblioteka, po želji se mogu dodati proširenja kao što je React Snippets u Visual Studio Codeu radi lakšeg pisanja koda.

8.1. Generiranje baze slučajeva

Kako bi strojno učenje moglo što preciznije predvidjeti uspjeh, potrebno je imati što veću bazu podataka uzoraka slučajeva. Bazu podataka za strojno učenje moguće je napraviti od već postojećih podataka ili kao u ovom radu nasumično generirati 5000 slučajeva s njihovim ocjenama i drugim pokazateljima za sve razrede srednje škole i bodova s državne mature uz upotrebu Gaussove razdiobe. Gaussova krivulja još se naziva i normalna raspodjela te se primjenjuje u raznim prirodnim znanostima, kao i u znanostima koje se bave proučavanjem ponašanja. Normalna raspodjela primjenjuje se kada rezultat oblikuje mnoštvo manjih efekata. Poznata je čuvena Gaussova krivulja, uobičajen model za prikaz varijacija. Ona govori o prirodi nasumičnosti, a predstavlja Gaussovu ili normalnu raspodjelu. [24]



Slika 14. Prikaz Gaussove krivulje

Izvor: http://www.grad.hr/vera/webnastava/vjerojatnostistatistika/html/VIS594x.png (20.10.2022.)

Da bi se mogla generirati baza slučajeva i njihove ocjene, potrebno je prvo generirati slučajeve i njihove karakteristike. Kroz sljedeće primjere prikazane su karakteristike svakog pojedinog slučaja koje nam služe za što precizniji rezultat predikcije.

Primjer 1.

```
export type StudentInfo = {
    jmbag: number;
    diligence: number;
    naturalSciences: number;
    socialSciences: number;
    art: number;
    external: number;
    highSchoolQuality: number;
}
```

Kôd 1. Prikaz karakteristika slučaja

Izvor: Autor

U primjeru 1. vidljiva je struktura podataka (StudentInfo) za svaki pojedini slučaj. Iz toga vidimo da svaki slučaj sadržava JMBAG, marljivost (diligence), sposobnost u prirodnim znanostima (naturalSceinces), sposobnost u društvenim znanostima (socialSciences), sposobnost u umjetnosti (art) te kvalitetu srednje škole (highSchoolQuality) i kao takvi su fiksni brojevi. Vanjski faktor (external) je promjenjiv broj i on se mijenja kroz školovanje.

Primjer 2.

```
export type YearGrades = {

english: number;

literature: number;

math: number;

physics: number;

electro: number;

computer: number;

}
```

Kôd 2. Prikaz ocjena iz srednje škole

Izvor: Autor

Primjer 2. pokazuje strukturu podataka (YearGrades) za ocjene iz srednje škole. Ovisno o tome što se traži, ova struktura se može promijeniti. Za ovaj rad korišteno je sljedeće: engleski jezik (english), hrvatski jezik (literature), matematika (math), fizika (physics), elektrotehnika (electro) te informatika (computer).

Primjer 3.

```
export type ExitExamGrades = {
    literature: number;
    english: number;
    math: number;
}
```

Kôd 3. Prikaz ocjena s mature

Izvor: Autor

Primjer 3. prikazuje strukturu podataka za ocjene na državnoj maturi (ExitExamGrades). Podatke, odnosno ocjene koje će se koristiti za predikciju su: hrvatski jezik (literature), engleski jezik (english) te matematika (math).

Primjer 4.

```
export type HSGrades = {
    Y1: YearGrades;
    Y2: YearGrades;
    Y3: YearGrades;
    Y4: YearGrades;
    exitExam: ExitExamGrades;
}
```

Kôd 4. Prikaz strukture podataka ocjena iz srednje škole i s državne mature

Izvor: Autor

Kako bismo ocjene iz sve četiri godine srednje škole pripisali jednom slučaju, napravljena je prethodna struktura podataka. Iz nje vidimo kako smo strukturu podataka iz primjera 2. pripisali svakoj godini srednje škole i na to dodali bodove, odnosno ocjene s državne mature vidljive iz primjera 3.

Prije nego što generiramo ocjene iz srednje škole potrebno je još nekoliko koraka. Kako bismo generirali srednju vrijednost i rasipanje te donji limit, gornji limit i pomak, koristi se transformacija Box-Muller. To će omogućiti da se generira karakteristika za svaki slučaj i na osnovu toga njihove ocjene. [26]

Primjer 5.

```
export function randn_bm(mean: number, variance: number) {

    var u = 0, v = 0;

    while (u === 0) u = Math.random();

    while (v === 0) v = Math.random();

    const x = Math.sqrt(-2.0 * Math.log(u)) * Math.cos(2.0 * Math.PI
* v);

    return x * variance + mean;

}
```

Kôd 5. Prikaz srednje vrijednosti i rasipanja

Izvor: Autor

U primjeru 5. vidljiv je prikaz funkcije (randn_bm) koja će vratiti srednju vrijednost i rasipanje.

Primjer 6.

```
export function randn_bm2(min: number, max: number, skew: number) {

    let u = 0, v = 0;

    while (u === 0) u = Math.random()

    while (v === 0) v = Math.random()

    let num = Math.sqrt(-2.0 * Math.log(u)) * Math.cos(2.0 * Math.PI
* v)

    num = num / 10.0 + 0.5

    if (num > 1 || num < 0)

        num = randn_bm2(min, max, skew)

    else {

        num = Math.pow(num, skew)

        num *= max - min

        num += min

    }

    return num

}
```

Kôd 6. Prikaz gornjeg i donjeg limita, pomak

Izvor: Autor

U primjeru 6. vidljiv je prikaz funkcije (randn_bm2) koja će vratiti pomak te donji i gornji limit. Za generiranje karakteristika koristi se funkcija (createStudents) za slučajne brojeve s Gaussovom razdiobom vidljiva u primjeru 7.

Primjer 7.

```
export function createStudents(nStudents: number): StudentInfo[] {

    const students: StudentInfo[] = Array.from({ length: nStudents
}).map((v, index) => ({

        jmbag: index,

        diligence: randn_bm(0, 0.1),

        naturalSciences: randn_bm2(0.3, 1.1, 1.0),

        socialSciences: randn_bm2(0.3, 1.0, 1.0),

        art: randn_bm2(0.3, 1.0, 1.0),

        external: randn_bm(0, 0.1),

        highSchoolQuality: randn_bm(1, 0.1),

    }));

return students;

}
```

Kôd 7. Prikaz generiranja karakteristika za slučaj

Izvor: Autor

Nakon prethodnih koraka, u primjeru 8. generiraju se ocjene za slučaj. Funkcija iz primjera 8. prvo generira slučajni izlaz od 0 do 1 te ga množi s 4 i dodaje 1 pa ga zaokružuje na cijeli broj koji će u konačnici rezultirati ocjenom.

Primjer 8.

```
export const createYearGrades = (student: StudentInfo): YearGrades
=> {

    return {

        english: Math.round(limit(student.socialSciences +
student.diligence + student.external, 0, 1) * 4 + 1),

        literature: Math.round(limit(student.socialSciences +
student.diligence + student.external, 0, 1) * 4 + 1),

        math: Math.round(limit(student.naturalSciences +
student.diligence + student.external, 0, 1) * 4 + 1),

        physics: Math.round(limit(student.naturalSciences +
student.diligence + student.external, 0, 1) * 4  + 1),

        electro: Math.round(limit(student.art + student.diligence +
student.external, 0, 1) * 4 + 1),

        computer: Math.round(limit(student.art + student.diligence +
student.external, 0, 1) * 4 + 1)

    };

}
```

Kôd 8. Prikaz generiranja ocjena iz srednje škole za slučaj

Izvor: Autor

U primjeru 9. prikazan je kod generiranja bodova, odnosno ocjena za maturu. Ocjene će biti generirane na sličan način kao što je to slučaj u primjeru 8.

Primjer 9.

```
const createExamGrades = (student: StudentInfo): ExitExamGrades => {

    return {

        english: Math.round(limit((student.socialSciences +
student.diligence + student.external) * student.highSchoolQuality,
0, 1) * 4 + 1),

        literature: Math.round(limit((student.socialSciences +
student.diligence + student.external) * student.highSchoolQuality,
0, 1) * 4 + 1),

        math: Math.round(limit((student.naturalSciences +
student.diligence + student.external) * student.highSchoolQuality,
0, 1) * 4 + 1),

    }

}
```

Kôd 9. Prikaz generiranja ocjena s državne mature za slučaj

Izvor: Autor

U primjeru 10. vidljiv je prikaz koda koji generira ocjene iz sve četiri školske godine srednje škole i bodova, odnosno ocjena s državne mature. Nakon svake godine nasumično se mijenja vanjski utjecaj (student.external).

Primjer 10.

```
export const createHSGrades = (student: StudentInfo): HSGrades => {

    const Y1 = createYearGrades(student);

    student.external += (Math.random() - 0.5) * 0.1;

    const Y2 = createYearGrades(student);

    student.external += (Math.random() - 0.5) * 0.1;

    const Y3 = createYearGrades(student);

    student.external += (Math.random() - 0.5) * 0.1;

    const Y4 = createYearGrades(student);

    student.external += (Math.random() - 0.5) * 0.1;

    const jmbag = student.jmbag;

    const exitExam = createExamGrades(student);

    return { Y1, Y2, Y3, Y4, exitExam };

}
```

Kôd 10. Prikaz generiranja ocjena iz srednje škole i s državne mature

Izvor: Autor

Slika 15. Prikaz generirane baze slučajeva i njihovih ocjena

Izvor: Autor

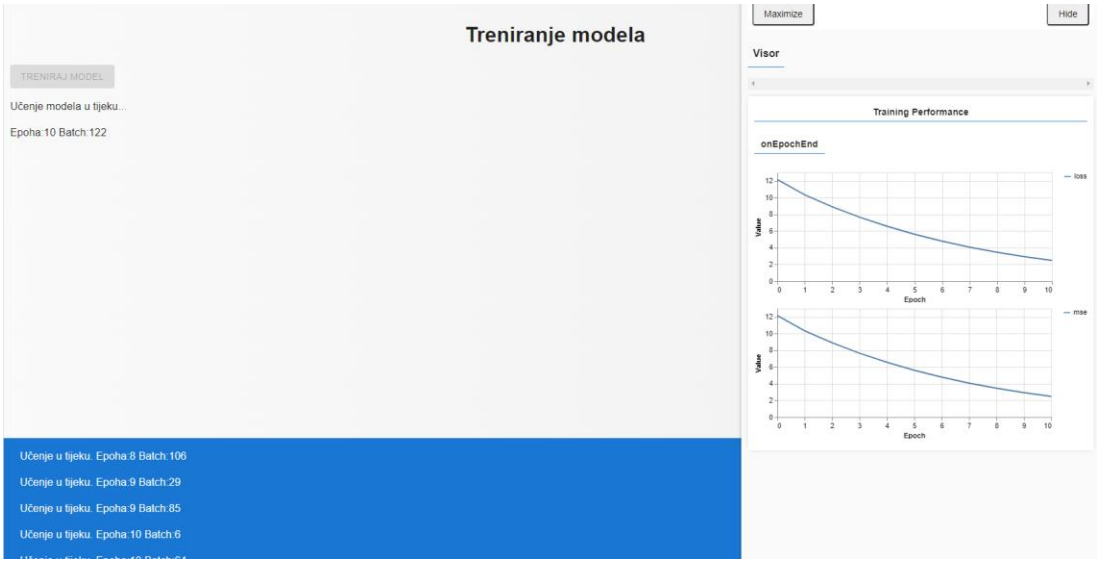
Na slici 15. vidljiv je prikaz generirane baze slučajeva i njihovih ocjena. Da bismo dobili ovakav gotov prikaz generirane baze slučajeva i njihovih ocjena, potrebno je bilo napraviti sve korake koji su navedeni u prethodnim primjerima. Klikom na gumb „Generiraj bazu" poziva se funkcija koja stvara listu od 5000 generiranih slučajeva s ocjenama iz prethodno navedenih primjera. Također, nakon klika na gumb „Generiraj bazu", pojavljuje se drugi gumb koji nudi mogućnost preuzimanja generirane baze u .CSV[11] formatu. Također postoji forma koja omogućuje listanje između generiranih slučajeva, kako bismo vidjeli generirane ocjene iz sve četiri godine srednje škole te ocjena iz mature za svaki slučaj.

8.2. Treniranje modela

Nakon generiranja baze slučajeva te s njihovim ocjenama treniramo model. Model strojnog učenja funkcija je s parametrima koji se mogu naučiti i koja preslikava ulaz u željeni izlaz. Optimalni parametri se dobivaju treniranjem modela na podacima. Obuka uključuje nekoliko

[11] Comma-separated values

koraka: dohvaćanje serije podataka u model, traženje modela da napravi predviđanje uspoređujući to predviđanje s „pravom“ vrijednošću i odlučivanje koliko promijeniti svaki parametar kako bi model mogao napraviti bolje predviđanje u budućnosti za tu seriju. Dobro uvježban model osigurat će točno preslikavanje od ulaza do željenog izlaza. Primjer treniranja modela vidi se na slici 16. U sljedećim primjerima vidi se kako izgleda treniranje modela.



Slika 16. Prikaz treniranja modela

Izvor: Autor

Primjer 11.

```
export function convertYearGradesToArray(grades: YearGrades) {

  return [grades.electro, grades.english, grades.literature,
grades.math, grades.computer, grades.physics];

}
```

Kôd 11. Pretvaranje ocjena na kraju godine u niz

Izvor: Autor

Generirane ocjene za slučajeve potrebno je pretvoriti u niz (array) kako bismo pripremili podatke za Tensorovu 2d matricu. To isto treba napraviti i s ocjenama iz mature.

Primjer 12.

```
export function convertExitExamGradesToArray(grades: ExitExamGrades)
{

  return [grades.literature, grades.english, grades.math];

}
```

Kôd 12. Pretvaranje ocjena s mature u niz

Izvor: Autor

U primjeru 13. vidi se pretvaranje ulaznih podataka u tenzor. Tenzor je 2d matrica u kojoj su redci svi podaci za 1 slučaj.

Primjer 13.

```
export function convertToTensor(records: StudentRecord[]) {
  const inputs = records.map(record =>
    [...convertYearGradesToArray(record.hsGrades.Y1),
    ...convertYearGradesToArray(record.hsGrades.Y2),
    ...convertYearGradesToArray(record.hsGrades.Y3),
    ...convertYearGradesToArray(record.hsGrades.Y4),
    ...convertExitExamGradesToArray(record.hsGrades.exitExam),
    record.collegeGrades.collegeId
    ],
  );
  const outputs = records.map(record =>
    [record.collegeGrades.Y1, record.collegeGrades.Y2,
record.collegeGrades.Y3]
```

```
  );
  const inputTensor = tf.tensor2d(inputs);
  const outputTensor = tf.tensor2d(outputs);
  return [inputTensor, outputTensor];
}
```

Kôd 13. Pretvaranje izlaznih podataka

Izvor: Autor

Iz primjera 13. vidljivo je pretvaranje izlaznih podataka u tenzor. Ulaz u funkciju su zapisi ocjena za sve studente. Izlazi funkcije:

- inputTensor – ocjene srednje škole, mature, upisani fakultet
- outputTensor – prosjek ocjene na prvoj godini fakulteta.

Primjer 14.

```
export function createModel() {
const model = tf.sequential();
model.add(tf.layers.dense({ inputShape: [28], units: 5, useBias: true
}));
model.add(tf.layers.dense({ units: 5, useBias: true, activation:
'softmax' }));
model.add(tf.layers.dense({ units: 3, useBias: true }));
return model;
}
```

Kôd 14. Kreiranje modela za učenje

Izvor: Autor

Iz primjera 14. vidi se kreiranje modela za učenje. U funkciji (createModel) vidljiva su tri sloja modela. Prvi sloj će biti ulazni i on mora biti jednak ukupnom broju ocjena. Drugi sloj je „Softmax“ neuronska mreža s nelinearnim elementima i posljednji sloj je izlazni sloj koji mora biti jednak broju izlaza modela. U ovom slučaju vidljiva su tri izlaza.

Primjer 15.

```
export async function trainModel(model: LayersModel, inputs:
Tensor2D, outputs: Tensor2D, onYield:any) {
  model.compile({
    optimizer: tf.train.adam(),
    loss: tf.losses.meanSquaredError,
    metrics: ['mse'],
  });
  const batchSize = 32;
  const epochs = 50;
  const callbacks = {
    ...tfvis.show.fitCallbacks(
      { name: 'Training Performance' },
      ['loss', 'mse'],
      { height: 200, callbacks: ['onEpochEnd'] }
    ),
    "onYield": (epoch: number, batch: number, logs: any) => {
      onYield(epoch, batch, logs);
     }
  };
  return await model.fit(inputs, outputs, {
```

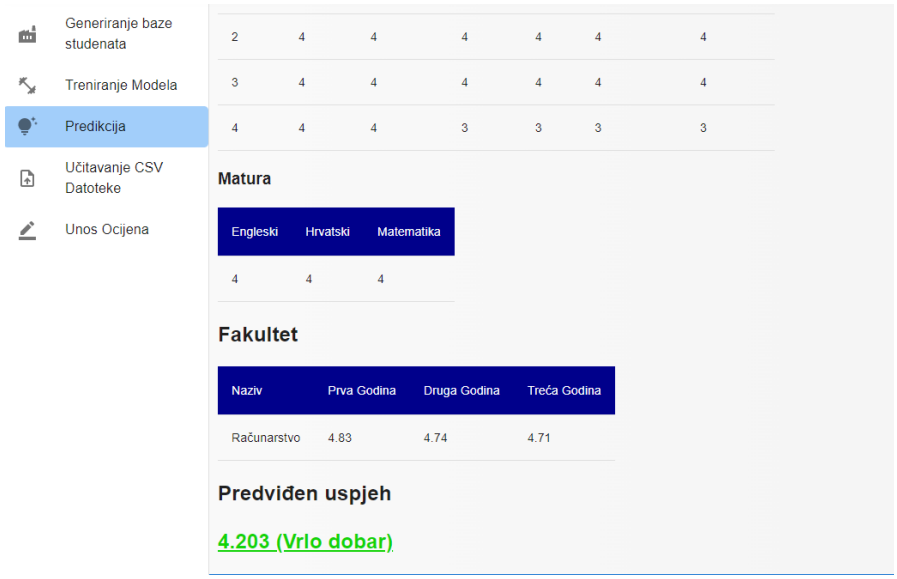
```
    batchSize,
    epochs,
    shuffle: true,
    validationSplit: 0.2,
    callbacks: callbacks,
    yieldEvery: 500,
  });
}
```

Kôd 15. Treniranje modela

Izvor: Autor

Primjer 15. prikazuje učenje modela. Za učenje modela postavljeno je pedeset koraka učenja i u svakom koraku gleda se trideset i dva slučaja. Koristi se Adam algoritam za traženje parametara mreže. U parametru (callbacks) poziva se funkcija koja iscrtava kvalitetu učenja.

8.3. Predikcija



Slika 17. Prikaz predviđenog prosjeka na veleučilištu

Izvor: Autor

Nakon što je model treniran uz pomoć generirane baze slučajeva, vidljiv je predviđen prosjek za sve tri godine fakulteta i konačno predviđanje uspjeha. Na stranici predikcija unutar aplikacije dobivamo rezultat predikcije na osnovi unošenja ocjena ili će program samostalno nasumično generirati slučaj s njegovim ocjenama, predvidjeti uspjeh te ga prikazati. Uz samu predikciju model će također prikazati najbolji odabir smjera na veleučilištu na osnovu težine samog smjera te vanjskih utjecaja. Sljedeći primjer pokazuje kako je to napravljeno.

Primjer 16.

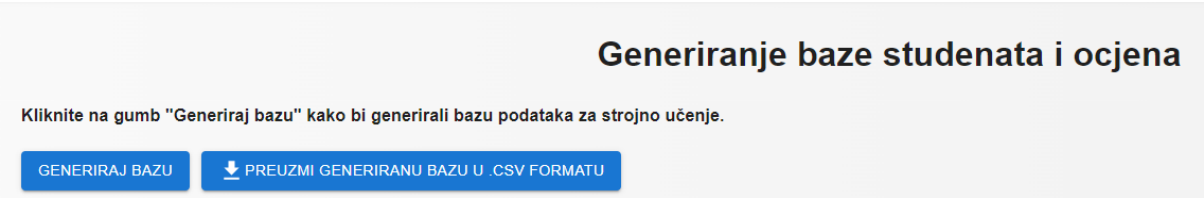
```
let records;
    if(manualStudentRecord) {
        records = [manualStudentRecord];
    }else{
        const students = createStudents(1);
        records = createStudentRecords(students);
    }
```

Kôd 16. Prikaz koda nasumično generiranog slučaja za predikciju

Izvor: Autor

Primjer 16. prikazuje kako je programu zadano da nasumično generira slučaj ako samostalno ne unesemo ocjene.

8.4. Generiranje CSV datoteke za izvoz



Slika 18. Prikaz preuzimanja generirane baze slučajeva u CSV formatu

Izvor: Autor

Za mogućnost dijeljenja generirane baze između drugih aplikacija ili uvoz druge u našu aplikaciju bit će korišten tablični CSV format. CSV format je podržan u mnogim aplikacijama. Sljedeći primjeri pokazuju kako je implementirana CSV tablična forma u našu aplikaciju za izvoz i uvoz baze.

Primjer 17.

```
function csvHeader() {

    const yearCourses = ["Engleski", "Hrvatski", "Matematika",
"Fizika", "Elektrotehnika", "Informatika"];

    const grades = ["Y1_", "Y2_", "Y3_", "Y4_"].flatMap(y =>
yearCourses.map(course => y + course)).concat("Matura Hrvatski",
"Matura Engleski", "Matura Matematika");

    return ["JMBAG"].concat(grades);

}
```

Kôd 17. Kod zaglavlja CSV tablice

Izvor: Autor

U primjeru 17. napravljena je funkcija (csvHeader) koja će nam 2d listu (array) pretvoriti u string tip podatka kako bismo ih mogli prenijeti u CSV datoteku.

Primjer 18.

```
function studentArrayToCsv(student: StudentRecord) {

    return [student.jmbag,

    convertYearGradesToArray(student.hsGrades.Y1),

    convertYearGradesToArray(student.hsGrades.Y2),

    convertYearGradesToArray(student.hsGrades.Y3),

    convertYearGradesToArray(student.hsGrades.Y4),

    convertExitExamGradesToArray(student.hsGrades.exitExam)

    ].join(',')

}
```

Kôd 18. Prikaz pretvaranja ocjena u listu za CSV format

Izvor: Autor

U primjeru 18. ocjene iz srednje škole i ocjene s državne mature pretvorene su u listu te ih odvajamo zarezom.

Primjer 19.

```
function exportToCsv(records: StudentRecord[]) {

    return [csvHeader().join(','), ...records.map(r =>
studentArrayToCsv(r))].join('\r\n');

}
```

Kôd 19. Prikaz funkcije koja izvozi bazu podataka

Izvor: Autor

Iz primjera 19. vidljiva je jednostavna funkcija koja će ocjene iz srednje škole i ocjene s državne mature izvesti u CSV format.

Primjer 20.

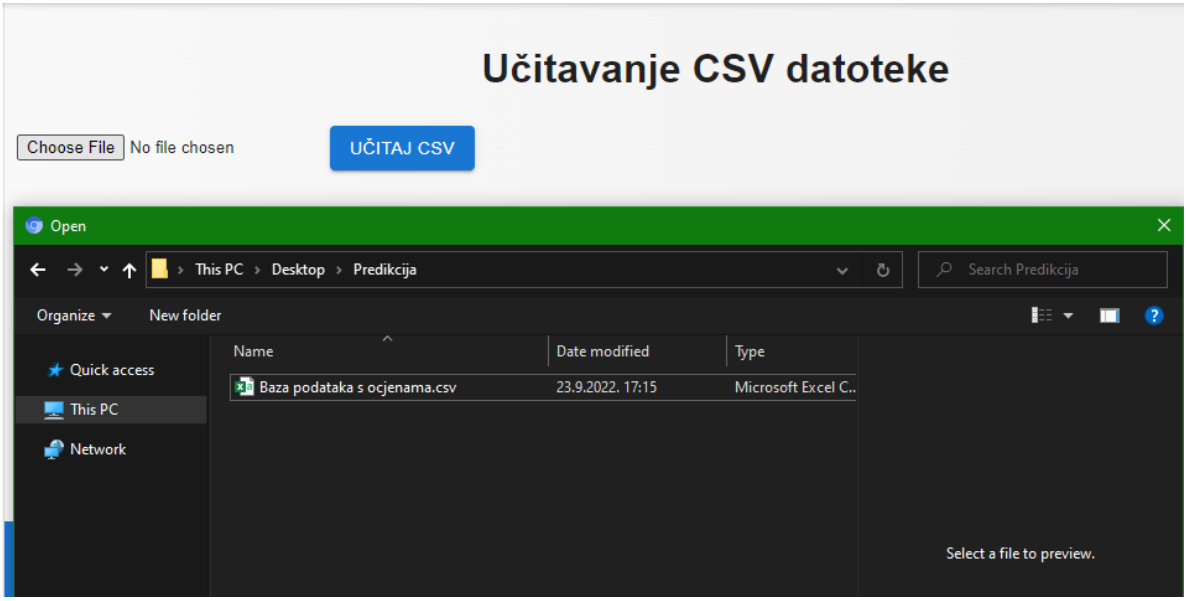
```
<Button variant="contained">

<CSVLink data={exportToCsv(studentRecords)} filename="Generirana baza.csv">

</Button>
```

Kôd 20. Prikaz koda gumba za preuzimanje baze

Izvor: Autor

U primjeru 20. napravljen je gumb koji će omogućiti preuzimanje generirane baze ocjena.

8.5. Uvoz CSV datoteke



Slika 19. Prikaz uvoza CSV datoteke

Izvor: Autor

Primjer 21.

```
export const PredictView2 = ({ model }: PredictViewProps):
JSX.Element => {
    const [file, setFile] = useState();
    const fileReader = new FileReader();
    const [students, setStudents] = useState<HSStudentRecord[]>();
    const handleOnChange = (e: any) => {
        setFile(e.target.files[0]);
    };
    const handleOnSubmit = (e: any) => {
        e.preventDefault();
        if (file) {
            fileReader.onload = function (event:
ProgressEvent<FileReader>) {
                const csvOutput = event.target!.result as string;
                setStudents(csv2Records(csvOutput));


  };
            fileReader.readAsText(file);
        }
    };
    return ( <div>
            <h1 style={{textAlign:"center", paddingTop:
"10px"}}>Učitavanje CSV datoteke</h1>
            <form>
                <input
                    type={"file"}
                    id={"csvFileInput"}
                    accept={".csv"}
                    onChange={handleOnChange}
                />
                    <Button variant="contained"  onClick={(e) => {
```

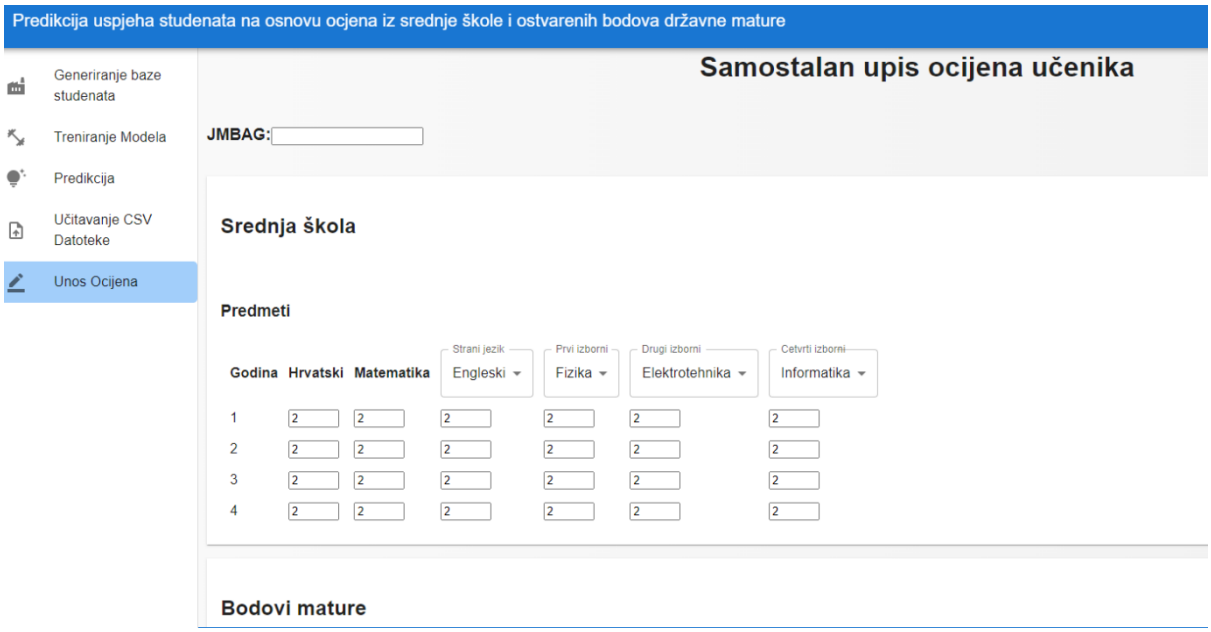
```
                        handleOnSubmit(e);

                    }}>Učitaj CSV

                    </Button>

                </form>

                {students && students.map(student =>

                    <StudentPredictView key={student.jmbag}
record={student} model={model} />

                )

                }

            </div>

        )
}
```

Kôd 21. Prikaz koda koji učitava CSV datoteku

Izvor: Autor

U primjeru 21. prikazan je kod koji omogućuje uvoz CSV datoteku u program. Također je vidljivo kako je napravljen gumb.

8.6. Samostalan unos ocjena



Slika 20. Prikaz samostalnog upisa ocjena

Izvor: Autor

Kako bismo predvidjeli uspjeh na veleučilištu za samo jedan slučaj, napravljena je stranica unutar aplikacije koja omogućuje samostalan upis ocjena te na temelju tih upisanih ocjena predikcija će izračunati, odnosno predvidjeti uspjeh. Sljedeći primjeri ilustriraju načinjeno.

Primjer 22.

```
<thead>
                    <tr>
                        <th>Godina</th>
                        <th>Hrvatski</th>
                        <th>Matematika</th>
                        <th>{ !optionalSubjects ?  "Engleski" :
optionalSubjects[0]}</th>
                        <th>{!optionalSubjects ?  "Fizika" :
optionalSubjects[1]}</th>
                        <th>{!optionalSubjects ?  "Elektrotehnika" :
optionalSubjects[2]}</th>
                        <th>{!optionalSubjects ?  "Informatika" :
optionalSubjects[3]}</th>
                    </tr>
                </thead>
```

Kôd 22. Prikaz koda tablice

Izvor: Autor

Da bismo mogli unijeti ocjene u model, napravljena je tablica na kojoj će se u sljedećem primjeru dodati i mogućnost unosa brojeva odnosno forme.

Primjer 23.

```
<input value={Y1.literature} onChange={(e) => {setY1({...Y1,
literature: Number(e.target.value)})}} type="number" min="2" max="5"
></input>
```

Kôd 23. Prikaz koda forme za upis ocjena

Izvor: Autor

Iz primjera 24. vidljiva je varijabla koja omogućuje da upišemo ocjene za svaki predmet iz svake godine te isto tako mature. U primjeru 23. navedena je samo jedna forma zbog veličine koda. Da bismo unesene ocjene mogli proslijediti u predikciju, potrebno je dohvatiti te podatke koji su upisani u forme i zapisati ih u varijable. U tome će nam pomoći primjer 24.

Primjer 24.

```
const [ exitExam, setExitExam ] = useState(defaultExitExam);

    const [ Y1, setY1 ] = useState(defaultY1);

    const [ Y2, setY2 ] = useState(defaultY2);

    const [ Y3, setY3 ] = useState(defaultY3);

    const [ Y4, setY4 ] = useState(defaultY4);

    const [ jmbag, setJmbag] = useState("");
```

Kôd 24. Prikaz koda zapisivanja u varijable

Izvor: Autor

Primjer 25.

```
type StudentGradesProps = {

    setManualStudentRecord: (manualStudentRecord: StudentRecord |
null) => void;

};
```

Kôd 25. Prikaz koda upisanih ocjena spremnih za predikciju

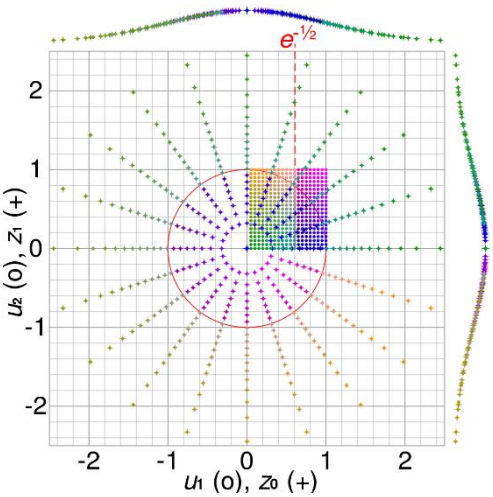
Izvor: Autor

U primjeru 25. sve upisane ocjene spremne su za predikciju odnosno učitavanje u model.

9. ANALIZA REZULTATA

U ovom radu generirana je baza od pet tisuća slučajeva s njihovim ocjenama iz srednje škole tijekom sve četiri godine. Za ocjene su uzeti predmeti iz srednje škole iz hrvatskog jezika, matematike, stranog jezika, fizike, informatike i elektrotehnike i ti predmeti su fiksni. Generiranu bazu podataka moguće je proširiti na način da se doda mogućnost izbora predmeta prije samog generiranja baze te bismo na taj način proširili upotrebu same aplikacije. Baza podataka sadrži za svaki slučaj njegov jedinstveni matični broj građana (JMBG), šest predmeta iz srednje škole za sve četiri godine te tri predmeta s državne mature koji su također fiksni. Isto tako, ako je potrebno za širu upotrebu aplikaciju proširiti, bilo bi moguće dodati mogućnost zamjene predmeta s državne mature ili dodati dodatne predmete koji bi bili uvjet za upis na veleučilište.

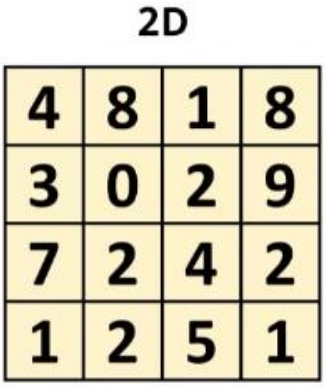
Za generiranje ocjena iz srednje škole napravljena je lista slučajeva sa slučajno odabranim karakteristikama kao što su marljivost, sposobnost slučaja, kvaliteta srednje škole te vanjski faktor koji nikada nije fiksan i mijenja se tijekom srednjoškolskog obrazovanja. Vanjski faktor možemo gledati kao nemogućnost prisustvovanja nastavi, dok je za generiranje samih karakteristika svakog slučaja korištena funkcija za slučajne brojeve s Gaussovom razdiobom. [26]



Slika 21. Box-Muller transform

Izvor: https://en.wikipedia.org/wiki/Box–Muller_transform#/media/File:Box-Muller_transform_visualisation.svg (7.11.2022.)

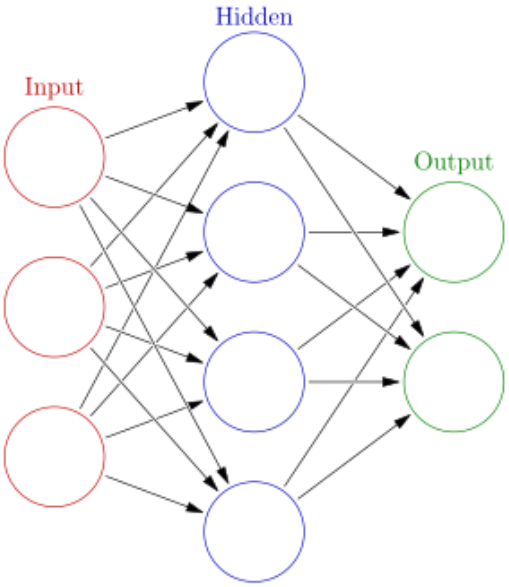
Zatim se prema karakteristikama i slučajnim varijablama određuju ocjene za srednju školu imajući na umu da nakon svake godine srednje škole dodatan utjecaj na ocjenu u sljedećoj akademskoj godini ima vanjski utjecaj koji će dati manje predvidljiv ishod. Generiranje ocjena odnosno bodova s državne mature napravljeno je tako da karakteristike svakog slučaja pomnožimo s kvalitetom srednje škole i na taj način dobijemo nasumičnu ocjenu iz državne mature. Generirani podaci spremljeni su u tabličnu bazu podataka odnosno kako ga TensorFlow naziva Tensor2D.



Slika 22. Tensor2d matrica

Izvor: https://www.techspot.com/article/2049-what-are-tensor-cores/ (22.11.2022.)

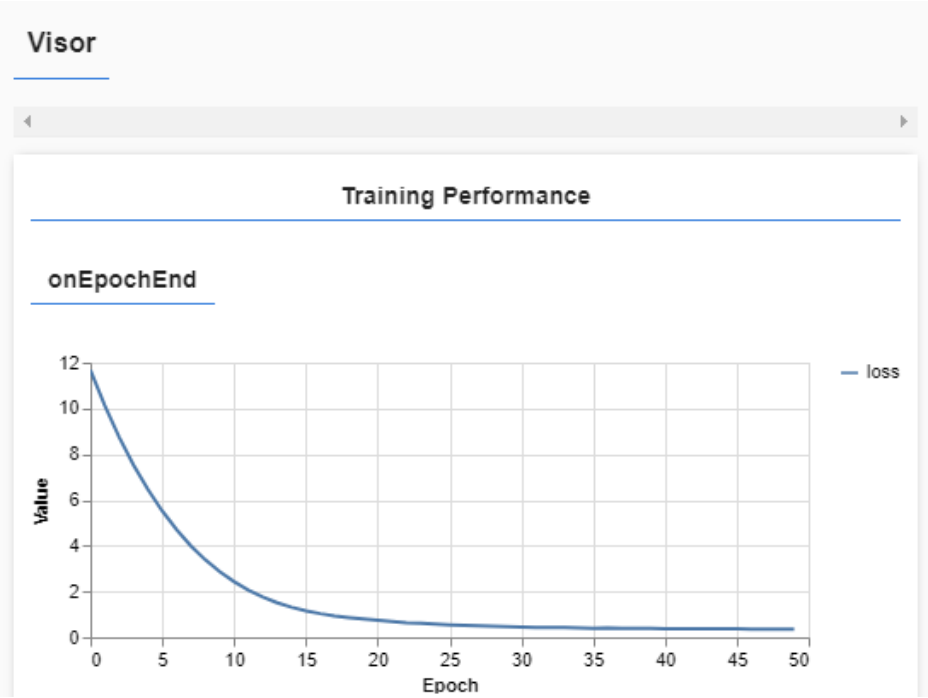
Korišten model za učenje kako bi se predvidjele ocjene na veleučilištu napravljen je kao neuronska mreža koja ima nelinearne elemente (Softmax). Također je korišten Adam algoritam za traženje parametara mreže koji nastoji minimizirati funkciju srednje kvadratne greške. Ulazni sloj mora biti jednak ukupnom broju ocjena koji je u našem slučaju 28.



Slika 23. Izgled neuronske mreže

Izvor: https://en.wikipedia.org/wiki/Artificial_neural_network#/media/File:Colored_neural_network.svg (22.11.2022.)

Za učenje modela korišteno je 5000 slučajeva srednje škole sa 6 predmeta i 3 predmeta s državne mature, što ukupno rezultira s 9 ukupnih ocjena. Iz toga zaključujemo da se radi o 135 tisuća podataka iz kojih će naš model učiti. Model učenja podešen je tako da imamo 50 koraka učenja te u svakom koraku gledamo 32 slučaja. Također, prilikom samog učenja imamo Tensor Visor koji grafički prikazuje samu kvalitetu učenja odnosno njezinu preciznost.



Slika 24. Prikaz Tensor Visora – kvalitete učenja

Izvor: Autor

Iz slike 24. možemo zaključiti da model postaje precizniji u predviđanju uspjeha time što više koraka ubacimo u model.

10. ZAKLJUČAK

Ovim završnim radom prikazana je upotreba programskih jezika TypeScript, HTML i CSS za razvoj korisničkog sučelja (user interface) te strojnog učenja TensorFlow za predikciju uspjeha slučajeva na veleučilištu. Korišteno je razvojno okruženje Visual Studio Code te mnoštvo biblioteka.

U ovom radu korišteno je 50 koraka učenja što je dovoljno precizno, no u slučajevima gdje trebamo veću preciznost trebali bismo što veću bazu podataka uzoraka te više koraka učenja. Uz samu predikciju uspjeha na veleučilištu napravljena je i mogućnost prikaza najpovoljnijeg smjera za upis svakog pojedinog učenika. Kako bi taj prikaz bio što realističniji, napravljena je lista smjerova na veleučilištu kao što su Računarstvo, Održivi razvoj te Menadžment turizma i sporta s njihovim težinskim koeficijentima. Prema tome, kada u model učenja uračunamo kvalitetu srednje škole, marljivost učenika, vanjski faktor kao nemogućnost dolaska na nastavu i težinske koeficijente smjerova na veleučilištu, uz predviđen uspjeh isto tako možemo predvidjeti i najpovoljniji smjer za pojedinog učenika.

11. LITERATURA

[1]. Strojno učenje
https://en.wikipedia.org/wiki/Machine_learning (23.6.2022.)

[2]. Visual Studio Code Logo
https://python-climbing.com/wp/wp-content/uploads/2019/10/vscode_icon-1024x576.png (23.6.2022.)

[3]. Visual Studio Code
https://en.wikipedia.org/wiki/Visual_Studio_Code (23.6.2022.)

[4]. IntelliSense
https://code.visualstudio.com/docs/editor/intellisense (01.7.2022.)

[5]. Visual Studio Code slika
https://pobierz.pl/wp-content/uploads/2020/05/Screenshot_13.jpg (23.6.2022.)

[6]. Google Chrome Logo
https://s1.ibtimes.com/sites/www.ibtimes.com/files/styles/full/public/2020/08/26/chrome.jpg (03.7.2022.)

[7]. Google Chrome
https://en.wikipedia.org/wiki/Google_Chrome (03.7.2022.)

[8]. HTML 5 Logo
https://www.taste-of-it.de/wp-content/uploads/2014/01/html5-logo.jpg (10.7.2022.)

[9]. HTML
https://en.wikipedia.org/wiki/HTML (10.7.2022.)

[10]. CSS
https://en.wikipedia.org/wiki/CSS (15.7.2022.)

[11]. CSS Logo
http://blog.brakoniecki.pl/wp-content/uploads/2015/02/css-logo.png (15.7.2022.)

[12]. JavaScript logo
https://quocent.com/QUOCENT-CMS/userfiles/StaticBanner/1529923467_Javascript.png (05.8.2022.)

[13]. JavaScript
https://en.wikipedia.org/wiki/JavaScript (05.8.2022.)

[14]. React
https://en.wikipedia.org/wiki/React_(JavaScript_library) (15.8.2022.)
[15]. React Logo
https://naomod.univ-nantes.io/idl/slides/images/react-logo.png (15.8.2022.)
[16]. TypeScript
https://en.wikipedia.org/wiki/TypeScript (18.8.2022.)
[17]. TypeScript Logo
https://humancoders-formations.s3.amazonaws.com/uploads/course/logo/230/formation-typescript.png (15.8.2022.)
[18]. NodeJS Logo
https://logos-download.com/wp-content/uploads/2016/09/Node_logo_NodeJS.png (1.9.2022.)
[19]. NodeJS
https://en.wikipedia.org/wiki/Node.js (1.9.2022.)
[20]. TensorFlow
https://en.wikipedia.org/wiki/TensorFlow (7.9.2022.)
[21]. TensorFlow Logo
https://www.w3cschool.cn/attachments/image/20170808/1502183953889061.png (7.9.2022.)
[22]. Tensor Softmax
https://en.wikipedia.org/wiki/Softmax_function (11.9.2022.)
[23]. Tensor Softmax slika
https://911weknow.com/wp-content/uploads/2020/09/understand-the-softmax-function-in-minutes-2.png (11.9.2022.)
[24]. Gaussova krivulja
https://hr.wikipedia.org/wiki/Normalna_raspodjela (27.9.2022.)
[25]. Gausssova krivulja slika

http://www.grad.hr/vera/webnastava/vjerojatnostistatistika/html/VIS594x.png (20.10.2022.)

[26]. Box Muller transform

https://en.wikipedia.org/wiki/Box–Muller_transform (7.11.2022.)

[27]. Box Muller transform slika

https://en.wikipedia.org/wiki/Box–Muller_transform#/media/File:Box-Muller_transform_visualisation.svg (7.11.2022.)

[28]. Tensor 2d matrica slika

https://www.techspot.com/article/2049-what-are-tensor-cores/ (22.11.2022.)

[29]. Neuronska mreža slika

https://en.wikipedia.org/wiki/Artificial_neural_network#/media/File:Colored_neural_network.svg (22.11.2022.)

12. PRILOZI

12.1. Popis slika

Slika 1. Visual Studio Code – logo 7
Slika 2. Visual Studio Code – radno okruženje 8
Slika 3. Web preglednik Google Chrome – logo 8
Slika 4. HTML 5 – logo 9
Slika 5. CSS – logo 9
Slika 6. JavaScript – logo 10
Slika 7. TypeScript – logo 10
Slika 8. React – logo 11
Slika 9. TensorFlow – logo 11
Slika 10. Node.js – logo 12
Slika 11. Funkcija Softmax 13
Slika 12. Kreirana mapa unutar Visual Studio Codea 14
Slika 13. Prikaz implementiranog React modula 15
Slika 14. Prikaz Gaussove krivulje 16
Slika 15. Prikaz generirane baze slučajeva i njihovih ocjena 25
Slika 16. Prikaz treniranja modela 26
Slika 17. Prikaz predviđenog prosjeka na veleučilištu 30
Slika 18. Prikaz preuzimanja generirane baze slučajeva u CSV formatu 32
Slika 19. Prikaz uvoza CSV datoteke 34
Slika 20. Prikaz samostalnog upisa ocjena 36
Slika 21. Box-Muller transform 40
Slika 22. Tensor2d matrica 41

Slika 23. Izgled neuronske mreže 42
Slika 24. Prikaz Tensor Visora – kvalitete učenja 43

12.2. Popis kodova

Kôd 1. Prikaz karakteristika slučaja 16
Kôd 2. Prikaz ocjena iz srednje škole 17
Kôd 3. Prikaz ocjena s mature 18
Kôd 4. Prikaz strukture podataka ocjena iz srednje škole i s državne mature 18
Kôd 5. Prikaz srednje vrijednosti i rasipanja 19
Kôd 6. Prikaz gornjeg i donjeg limita, pomak 20
Kôd 7. Prikaz generiranja karakteristika za slučaj 21
Kôd 8. Prikaz generiranja ocjena iz srednje škole za slučaj 22
Kôd 9. Prikaz generiranja ocjena s državne mature za slučaj 23
Kôd 10. Prikaz generiranja ocjena iz srednje škole i s državne mature 24
Kôd 11. Pretvaranje ocjena na kraju godine u niz 26
Kôd 12. Pretvaranje ocjena s mature u niz 27
Kôd 13. Pretvaranje izlaznih podataka 28
Kôd 14. Kreiranje modela za učenje 28
Kôd 15. Treniranje modela 30
Kôd 16. Prikaz koda nasumično generiranog slučaja za predikciju 31
Kôd 17. Kod zaglavlja CSV tablice 32
Kôd 18. Prikaz pretvaranja ocjena u listu za CSV format 33
Kôd 19. Prikaz funkcije koja izvozi bazu podataka 33
Kôd 20. Prikaz koda gumba za preuzimanje baze 34
Kôd 21. Prikaz koda koji učitava CSV datoteku 36
Kôd 22. Prikaz koda tablice 37
Kôd 23. Prikaz koda forme za upis ocjena 38
Kôd 24. Prikaz koda zapisivanja u varijable 38
Kôd 25. Prikaz koda upisanih ocjena spremnih za predikciju 39