Универзитет Св. Кирил и Методиј – Скопје  
  
Факултет за информатички науки и компјутерско инженерство

**Предмет: Вовед во науката за податоци  
  
Тема: Избор на податочно множество за Алцхајмерова болест и негова анализа преку соодветна методологија.**

Github линк до проектот : <https://github.com/majkltancin/proekt-vnp.git>

Ментор: Изработил:

Ана Тодоровска Мајкл Танчин 193187

Скопје, 2025

Содржина

[**Вовед** 3](#_Toc201527060)

[**Користени алатки и библиотеки** 4](#_Toc201527061)

[**Опис на избраното податочно множество** 4](#_Toc201527062)

[**Вчитување на датасет и подготовка на податоците** 5](#_Toc201527063)

[**Анализа и визуелизација на податоците** 5](#_Toc201527064)

[**Предпроцесирање на податоците** 6](#_Toc201527065)

[**Поделба на податочно множество** 6](#_Toc201527066)

[**Креирање на модели** 7](#_Toc201527067)

[**Споредба на модели** 8](#_Toc201527068)

[**Искористување на најдобриот модел за проверка на Алцхаемерова болест** 9](#_Toc201527069)

[**Заклучок** 10](#_Toc201527070)

[**Користена литератира** 10](#_Toc201527071)

# **Вовед**

Целта на овој проект е да се обработат податоци за алцхаемерова болест и да се проба со различни модели на машинско учење да се проучи податочното множество и да се предвиди дали некој пациент ја има или ја нема таа болест.

Зашто е битно да се создаваат тренираат и користат вакви модели на машинско учење за предвидување на оваа болест поред специјалисти во оваа област би се запрашале некои.

Ова е болест која го напaѓа мозокот кај поголем процент од повозрасната генерација и се работи за болест која за краток временски период од нејзиното добивање доведува до губење на конгнитивните функции, памтење, промена на личноста и способноста за функционирање. Со тоа наjчесто кога пациентите ќе се здобијат со симптоми веќе е доцна односно болеста е развиена до некој степен, затоа овој начин на предвидување на оваа болест е доста корисен односно им помага на специјалистите да ја потврдат или порано да ја увидат оваа дијагноза и да им помогнат на луѓето.

Со тоа овој проект ќе има за цел:

* Да го визуализираме и анализиреме податочното множество за алцхаемерова болест
* Да провериме дали има некои вредности кои фалат и да се дополнат доколку има таков случај
* Да се доведат потребните податоци во ист формат пред креирањето на моделите
* Креирање на повеќе модели
* Финални споредба на моделите и увидување на битни карактеристики за истите
* Одбирање на моделот со најдобри резултати и тестирање на истиот на одредени податоци за предвидување на алцхаемерова болест

# **Користени алатки и библиотеки**

За изработка на проект од овој тип и за споредба на неколкуте модели користени се следните алатки и библиотеки.

* Програмски јазик : Python
* CSV за вчитување на податоците
* Pandas за креирање на дата фрејм предобработка и манипулација со истиот
* Numpy за справување со нумерички вредности
* Seaborn за визуализација на податоците

# **Опис на избраното податочно множество**

Користениот датасет е именуван како alzheimers\_disease\_data.csv и претставува збир на демографски карактеристики , навики ,фамилијарни поврзаности , когнитивни и медицински мерења кај пациенти кои се предмет на истражување поврзано со Алцхајмеровата болест. Податоците се добиени од kaggle кој се наменети за истражувачки дејности и креирање на модели за предвидување .

Датасетот содржи приближно 2,149 записа (редови), каде секој запис претставува податок за еден пациент. Секој запис содржи повеќе од 30 променливи, меѓу кои: возраст, пол, степен на образование, навики, повреди, болести, и најбитно дијагноза.

Променливите се од различен тип: нумерички (на пример, возраст) и категориски (на пример, пол дијагноза.

# **Вчитување на датасет и подготовка на податоците**

Најпрво треба да ги вчитаме податоците од csv фајлот истиот да се провери дали е вчитан и какви податоци содржи. По овој чекор се прави проверка дали во тоа податочно множество се содржат одредени нул вредности. Тоа може да го направиме на најразлични начини, но кај мене во кодот тоа е направено со помош на missingо и heatmap од библиотеката seaborn. Овој дел го правиме поради тоа што голем број на модели имаат проблем со овие вредности кои недостасуваат и истите можно е да не се извршат.

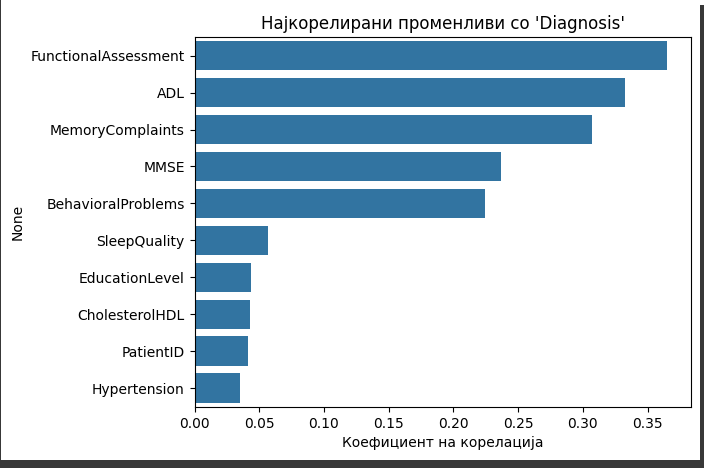
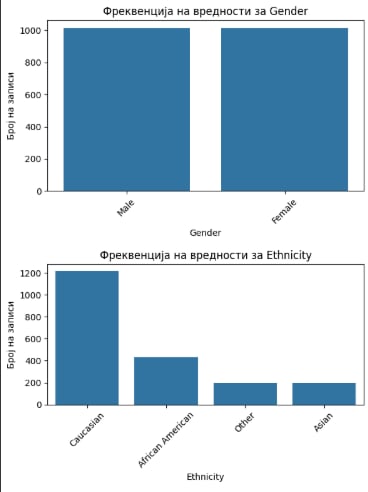
Откако ќе се изврши кодот за вредности кои недоставуваат се создаваат графици на кои се прикажува во кој дел од датасетот недостасуваат вредностите. Исто така со помош на графиците одприлика би можеле да дознаеме дали вредностите недостасуваат целосно случајно или има причина за истото.

# **Анализа и визуелизација на податоците**

Се креираат хистограми кои се генерирани врз база на податоците од кои се добиваат значајни информации како на пример како е распределено множеството на податоци какви типови на податоци имаме колкава е застапеноста на податоците од одреден атрибут. Од тука па натаму се користат овие податоци за да се одреди правецот на понатамошната обработка на податоците и кои модели би се користеле.

A graph with blue squares

AI-generated content may be incorrect.A graph of different types of data

AI-generated content may be incorrect.

# **Предпроцесирање на податоците**

Откако е направена целата анализа се продолжува со енкодирање на вредностите односно категориските податоци се претвараат во броеви. Бидејќи категориските податоци содржат класи кои не се многу бројни и истите се како на пример yes no одбрано е да се енкодираат истите со помош на label encoder. Целна на ова енкодирање односно користењето на label encoder е за понатамошната обработка на податоци односно искористување од страна на моделите кои работат со нумерички податоци за да предвидуваат други вредности. Така да во овој дел се енкодираат ситге променливи освен NaN вредностите кои остануваат како такви за да се предвидат во следниот чекор. Следниот чекор е предвидување на NaN вредностите, тоа се вредности кои недостасуваат, како што може да приметиме при проверката на null вредности од графикот кој го генерира missingo се гледа дека истите недостасуваат рандом па затоа за нив се користи KNN imputer кој ги гледа соседните вредности и според нив одлучува со која вредност ќе се пополнми тоа празно поле.

Откако овие вредности се претворени во броеви, може да искористе одредена функција за да провери кои атрибути највеќе колерираат со дијагнозата.

# **Поделба на податочно множество**

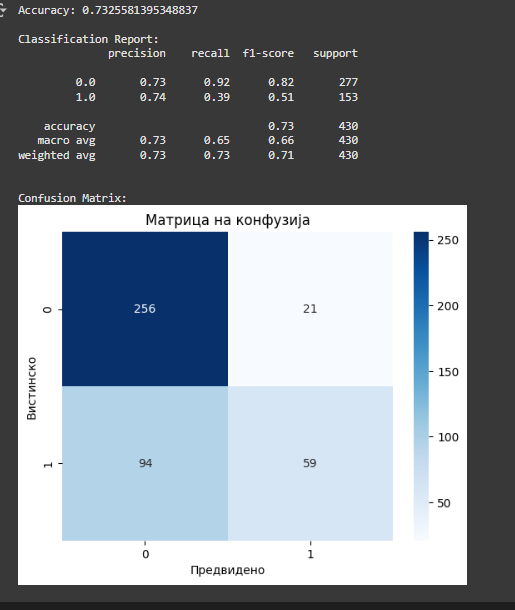
Откако ќе се направат предходните чекори, може да преминеме на поделба на податочното множество од кое предходно се исфрлени некои непотребни атрибути. Податочното множество го делиме така што имаме едно податочно множество на тестирање и едно податочно множество за тренирање на моделот и со целна карактеристика дијагнозата. Пред да го поделиме податочното множество е направено скалирање на сите атрибути освен на целната карактеристика. Скалирањето се прави поради тоа што имаме вредности кои варираат од еден до неколку во различните колони така да според тие варијации моделот може да си даде до знаење дека одредена колона е побитна од другата. По скалирањето вредностите ќе бидат во опсег помеѓу средната вредност и стандардната девијација.

# **Креирање на модели**

Се креираат најразлични модели почнувајки од не толку сложени како KNN, логистичка регресија, дрва на одлучување и RandomForest, кон посложени како xgboost и невронски мрежи. Како што наведовме најпрво започнуваме со КNN каде што првично пробуваме со помош на grid search и к folds за да одредиме со кои параметри најдобро ќе работи моделот со што потоа се креира моделот и се предвидува со моделот кој дава најдобри резултати. На крај се испишува accuracy, precision, f1 score. Користиме други метрики освен accuracy за да не добиеме лажна слика за перформанси на моделот. Поради што видовме од анализата податочното множество е небалансирано.

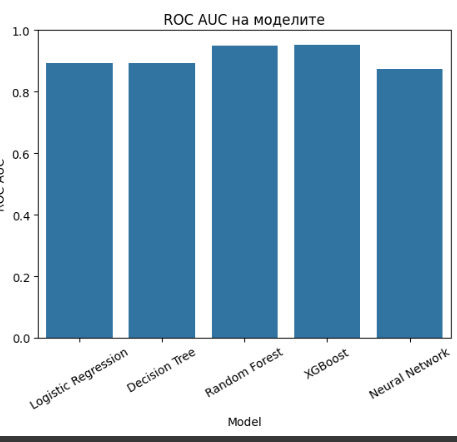
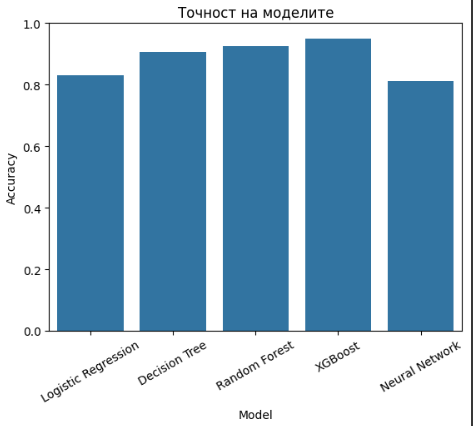
Потоа се тренира моделот логистичка регресија со максимален број на итерации 1000 со што се продолжува со предвидување на моделот и на крај се испишува неговата точност, прецизност и се креира класификацион репорт. Истото се прави за decision tree и random forest.

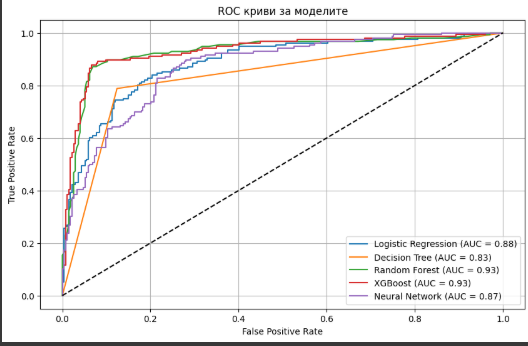
Од тука преминува на посложените модели како xgboost и невронската мрежа. Како и предходните модели се креира xgboost моделот со метрика за евалуација logloss. За крај се креира невронска мрежа која има 3 слоја и истата е од тип sequential. За оптимизатор за невронската мрежа го користиме adam а функција за загуба се користи binary crossentropy. Податочното множество се тренираат 30 епохи и batch size 32.



# **Споредба на модели**

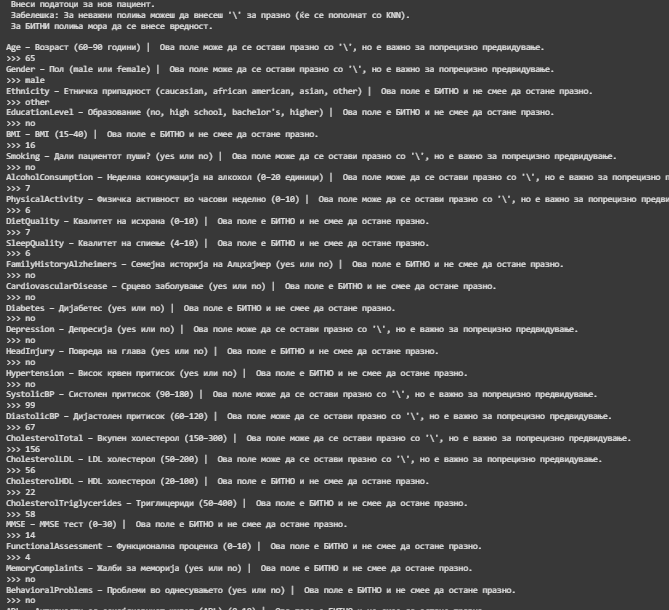
Се собираат вредностите со која точност погодуваат моделите и се прикажуваат како хистограмио и roc крива. Според овие хистограми може добро да се визуализира дека овие модели, кој подобро кој полошо, покажуваат добри перформанси така што точно ги предвидуваат вредностите. Се користи roc крива и roc auc score за да се прикажат перформансите на моделите а притоа да не бидат под влијание на небалансираното множество. Овие информации може понатаму да се искористат со тоа што ќе се извлече кое е најдобриот модел за ова податочно множество и со кој модел би можело понатаму да се предвидуваат одредени вредности.





# **Искористување на најдобриот модел за проверка на Алцхаемерова болест**

Се креира функција која одбира најдобар модел според роц кривата со која понатаму се предвидува дали одреден пациент има алцхаемер или не а за тој пациент се внесуваат податоци од наша страна. Функцијата се содржи од мапирање на податоците кои што се предходно енкодирани за да може понатаму кога корисникот ќе внесе одредени информации кои се текст да се преведат во бројки. Исто така имаме листа која содржи кои атрибути се битни. По неа има листа на фичери каде на пример имаме атрибут возраст, кој прима броеви за истиот има текст кој се прикажува при пополнување на податоците и дали е битен тој фичер. Потоа го имаме делот каде корисникот внесува податоци односно му се покажува кои податоци се битни истите се процесираат и се зачувуваат. По внесот на податоците доколку корисникот оставил некој влез празен ние го пополнуваме со помош на кнн за да може моделот да работи. Потоа се скалираат вредностите за да бидат во истите граници како и другитге податоци за на крај да се одбере моделот да се предвиди и прикаже резултатот.



# **Заклучок**

Со овој проект покажуваме дека машинското учење може да понуди прецизни алатки за поддршка на раната дијагноза на алцхамеровата болест. Највлијателни променливи кој доведоа до вакви резултати беа проблеми со меморијата и финкционирањето, проблеми со однесувањето и мрежата за секојдневни активности.

Најдобри резултати беа постигнати со XGBoost, што укажува на силниот потенцијал на напредните модели. Овие модели можат да помогнат во рана интервенција и подобрување на квалитетот на животот на пациентите.

# **Користена литератира**

<https://www.kaggle.com/datasets/rabieelkharoua/alzheimers-disease-dataset/data>

<https://www.wikipedia.org/>

<https://scikit-learn.org/stable/>

<https://courses.finki.ukim.mk/>