Data Science Road Map

1.Импортираме библиотеки

2.read\_csv на датата

3.Одредуваме таргет аку не е напишано во текстот

4.Ги drop на колотине кои не ни се потребни

5.Правиме визуелизација на датата најдобро со heatmap или pairplot-и од нив ги наојгаме јаките корелаци помегу две колони

6.Гледаме дали е балансиран таргетот(не знајме сеуште)ТРЕБА ДА ДОЗНАЈМЕ

7.Detect missing values

8.Ја наојгаме причината за missing values-најцесто со dendogram

9.handle the missing values:

-Првин ја дропнуваме таргет колоната

-После ги хендламе сите колони две по две кои што имаат јака корелација(јака корелација е над 0.6)

-Вредностите аку се нешто како година,за луге поедноставно све со нормална распределба, тогаш користиме mean()

-Вредностите аку се на пример 0 и 1 само и едното преовладуват најчесто со moda().iloc[0]- Односно ја зеаме најчестата

-Вредностите аку се категориски односно стрингови, тогаш тие ги претвораме со енкодер, АМААА аку се ординални односно имаат редослед(на пр. Да запишиш факултет требит да завршис основно сдрено итн. Или за да се разведис морат да си сингл па во врска па брак и на крај разведвенје) тогас користиме Label Encoder. АААА аку се номинални односно немаат редослед како на пр. female male, red,yellow тогаш се користит One-Hot Encoder.

10.Feature Preprocesing-Овде прајме таргет променлива и таа ја дропвиме за X и додаваме таргет за Y

11.Train-Test-Split

12.Одбираме модели со кои ке ја видиме нашата точност

-Logistic Regresion & XGBoost clasifier &KNN(и за категориски и за нумерички) & Decision Tree(за двете исто) се за категорички( на пр 1 и 0 или така нешто ко ке е со female I male) АМААА морат да му направиме енцодинг аку се стрингови, па потоа да ги употребуваме моделите

-Linear Regresion & XGBoost regresion & KNN(за двете) & Decision Tree(за двете исто).

13.Cross-Validation coprareson-Ги споредвиме кај категориски accurasy или f1-score ,зависно од K-folds.За ова ни треба билбиотеката

from sklearn.model\_selection import cross\_val\_score

и после зависно кој модел го користиме на пр.

from sklearn.model\_selection import cross\_val\_score

scaler = StandardScaler()

X\_train\_scaled = scaler.fit\_transform(X\_train) –овие се исти ко кај обицниот модел

X\_test\_scaled = scaler.transform(X\_test)

# Define the KNN model

knn = KNeighborsClassifier(n\_neighbors=3)

knn\_cv\_scores = cross\_val\_score(knn, X\_train\_scaled, y\_train, cv=5, scoring='accuracy')

print(f"KNN - Mean accuracy: {np.mean(knn\_cv\_scores)}, Std: {np.std(knn\_cv\_scores)}")-овие се додаени за cross validation, и во заградата пишуваме све исто освем првата вредност, зависно кој водел го користиме

---------------------------------------------------------------------------------------------------------------------

from sklearn.model\_selection import cross\_val\_score

from xgboost import XGBClassifier

import numpy as np

# Define the XGBoost model

xgb\_model = XGBClassifier(max\_depth=5, min\_child\_weight=1, n\_estimators=100, n\_jobs=-1, learning\_rate=0.1)

# Perform 5-fold cross-validation and get the accuracy scores

xgb\_cv\_scores = cross\_val\_score(xgb\_model, X\_train\_scaled, y\_train, cv=5, scoring='accuracy')

# Print the results

print(f"XGBoost - Mean accuracy: {np.mean(xgb\_cv\_scores)}, Std: {np.std(xgb\_cv\_scores)}")

Во слујаов ги користевме овие два

ДЕФИНИЦИИ

F1-score: Мерка за перформансите на моделот особено способноста да ја балансира прецизноста и потцетувањето(појма неам)

Recall- покажува колку добро моделот успева да ги единтификува сите позитивни примери во податоците.

Кога имаме небалансирани податоци accuracy може да биде погрешно и тука настапува f1-score кој што се грижи и негативните и позитивните класи да бидат земени во предвит.

Intercept-претставува вредност на излезот кога сите влезни карактеристики се нула.