**LAB 5**

**PREDICTION ALGORITHMS**

<https://www.kaggle.com/datasets/timoboz/tesla-stock-data-from-2010-to-2020>

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sb

from sklearn.model\_selection import train\_test\_split

from sklearn.preprocessing import StandardScaler

from sklearn.linear\_model import LogisticRegression

from sklearn.svm import SVC

from xgboost import XGBClassifier

from sklearn import metrics

import warnings

warnings.filterwarnings('ignore')

from google.colab import files

uploaded = files.upload()

**Reading Data**

df = pd.read\_csv('TSLA.csv')

df.head()

df.shape

df.describe()

df.info()

**Exploratory Data Analysis**

plt.figure(figsize=(15,5))

plt.plot(df['Close'])

plt.title('Tesla Close price.', fontsize=15)

plt.ylabel('Price in dollars.')

plt.show()

df.head()

df[df['Close'] == df['Adj Close']].shape

df = df.drop(['Adj Close'], axis=1)

df.isnull().sum()

**Distribution Plot**

features = ['Open', 'High', 'Low', 'Close', 'Volume']

plt.subplots(figsize=(20,10))

for i, col in enumerate(features):

  plt.subplot(2,3,i+1)

  sb.distplot(df[col])

plt.show()

**Box plot**

plt.subplots(figsize=(20,10))

for i, col in enumerate(features):

  plt.subplot(2,3,i+1)

  sb.boxplot(df[col])

plt.show()

**Feature Engineering**

splitted = df['Date'].str.split('-', expand=True)

df['year'] = splitted[0].astype('int')

df['month'] = splitted[1].astype('int')

df['day'] = splitted[2].astype('int')

df.head()

df['is\_quarter\_end'] = np.where(df['month']%3==0,1,0)

df.head()

**Bar graph**

data\_grouped = df.groupby('year').mean()

plt.subplots(figsize=(20,10))

for i, col in enumerate(['Open', 'High', 'Low', 'Close']):

  plt.subplot(2,2,i+1)

  data\_grouped[col].plot.bar()

plt.show()

df.groupby('is\_quarter\_end').mean()

df['open-close'] = df['Open'] - df['Close']

df['low-high'] = df['Low'] - df['High']

df['target'] = np.where(df['Close'].shift(-1) > df['Close'], 1, 0)

**Pie chart**

plt.pie(df['target'].value\_counts().values, labels=[0, 1], autopct='%1.1f%%')

plt.show()

**Heat Map**

plt.figure(figsize=(10, 10))

sb.heatmap(df.corr() > 0.9, annot=True, cbar=False)

plt.show()

**Data Splitting and Normalization**

features = df[['open-close', 'low-high', 'is\_quarter\_end']]

target = df['target']

scaler = StandardScaler()

features = scaler.fit\_transform(features)

X\_train, X\_valid, Y\_train, Y\_valid = train\_test\_split(features, target, test\_size=0.1, random\_state=2022)

print(X\_train.shape, X\_valid.shape)

models = [LogisticRegression(), SVC(

kernel='poly', probability=True), XGBClassifier()]

for i in range(3):

  models[i].fit(X\_train, Y\_train)

  print(f'{models[i]} : ')

  print('Training Accuracy : ', metrics.roc\_auc\_score(Y\_train, models[i].predict\_proba(X\_train)[:,1]))

  print('Validation Accuracy : ', metrics.roc\_auc\_score(Y\_valid, models[i].predict\_proba(X\_valid)[:,1]))

  print()

**Confusion matrix**

metrics.plot\_confusion\_matrix(models[0], X\_valid, Y\_valid)

plt.show()