Machine learning

Milestone1 - Report

Data Preprocessing

1. preprocessing techniques
    a. Missing values:

        Dataset contained Some missing values so we dropped any row that contained at least one missing value, but we have done this after choosing the features that we are going to use.

```
data.dropna(how='any',inplace=True)
```

    b. Text values:

        Since computers can’t deal with string values, so we need to convert any text data to numbers, so we have changed Prime_genre column to numeric values using One-Hot-Encoding technique.

```
one_hot = pd.get_dummies(data['prime_genre'])
data = data.join(one_hot)
data = data.drop(['prime_genre'],axis=1)
```

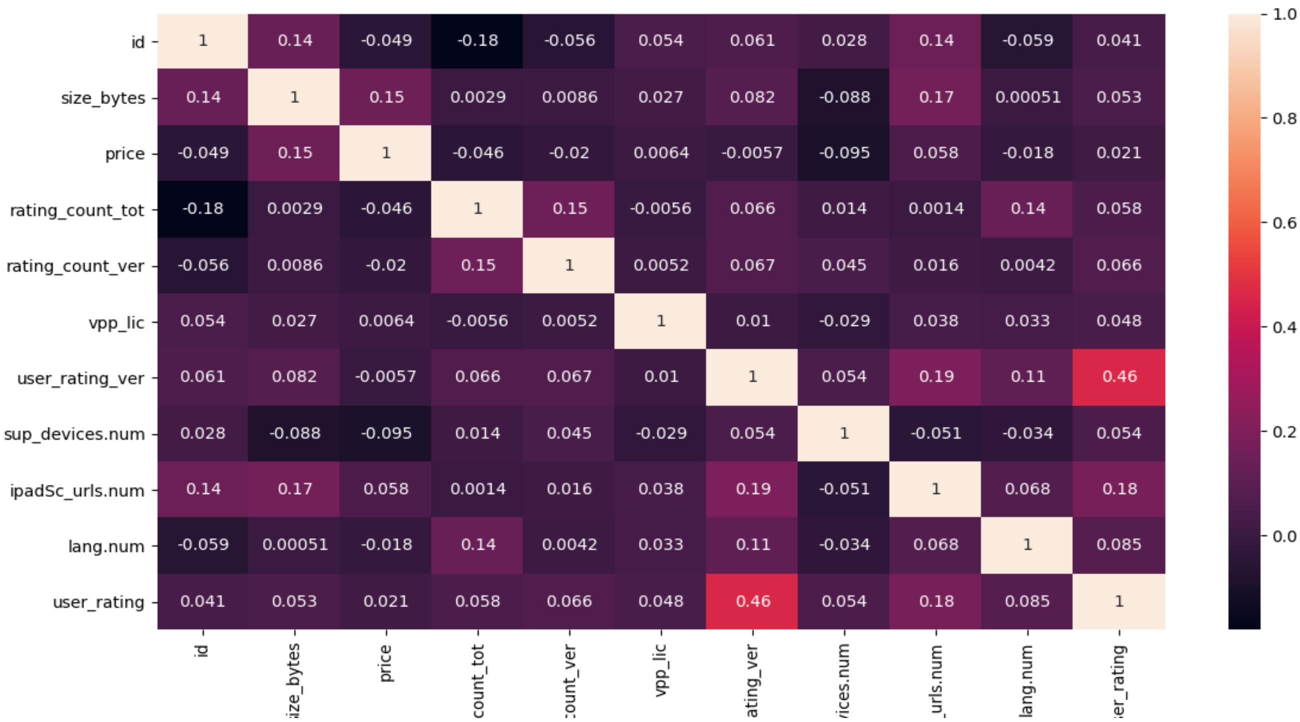
    c. Feature scaling:

        Features values differ very much in ranges between them so som attributes would affect the output more than other attributes only because it has higher ranges, so it is important to apply feature scaling. We applied min-max scaling on two columns ['rating_count_tot','rating_count_ver']

```
scaler = MinMaxScaler()
data['rating_count_tot'] =
scaler.fit_transform(np.array(data['rating_count_tot'])
.reshape(-1,1))
```

```python
data['rating_count_ver'] =
scaler.fit_transform(np.array(data['rating_count_ver'])
.reshape(-1,1))
```

2. Analysis
    a. Applied correlation between features to see which features have the most effect on the output (User_Rating) and decide which features we are going to use in our regression models



    b. After Applying correlation it was observed that the user_rating_ver has the most effect on the target value

3. Features Used

Feature	Description
rating_count_tot	User Rating counts (for all version)
rating_count_ver	User Rating counts (for current version)
user_rating_ver	Average User Rating value (for all version)
prime_genre	Primary Genre
sup_devices.num	Number of supporting devices
lang.num	Number of supported languages
ipadSc_urls.num	Number of screenshots showed for display

Regression Techniques

As we know that regression models are used to describe relationships between variables by fitting a line to the observed data, Regression allows you to estimate how a dependent variable changes as the independent variable changes.

So ,we used two techniques, Multiple linear regression and Polynomial Regression.

We also tried **enhancing** the results using **L2** regularization.

- **Multiple Linear Regression:**

  “Y=bo+b1x1+b2x2+..........+BnXn”

  Multiple Linear regression are based on the assumption that there is a linear relationship between both the dependent and independent variables or predictor variable and target variable, and it also assumes that there’s no major correlation between the independent variables.

  Multi linear regression can be linear and nonlinear, and it has one y and one or more x.

  After Implementation of the multiple linear regression

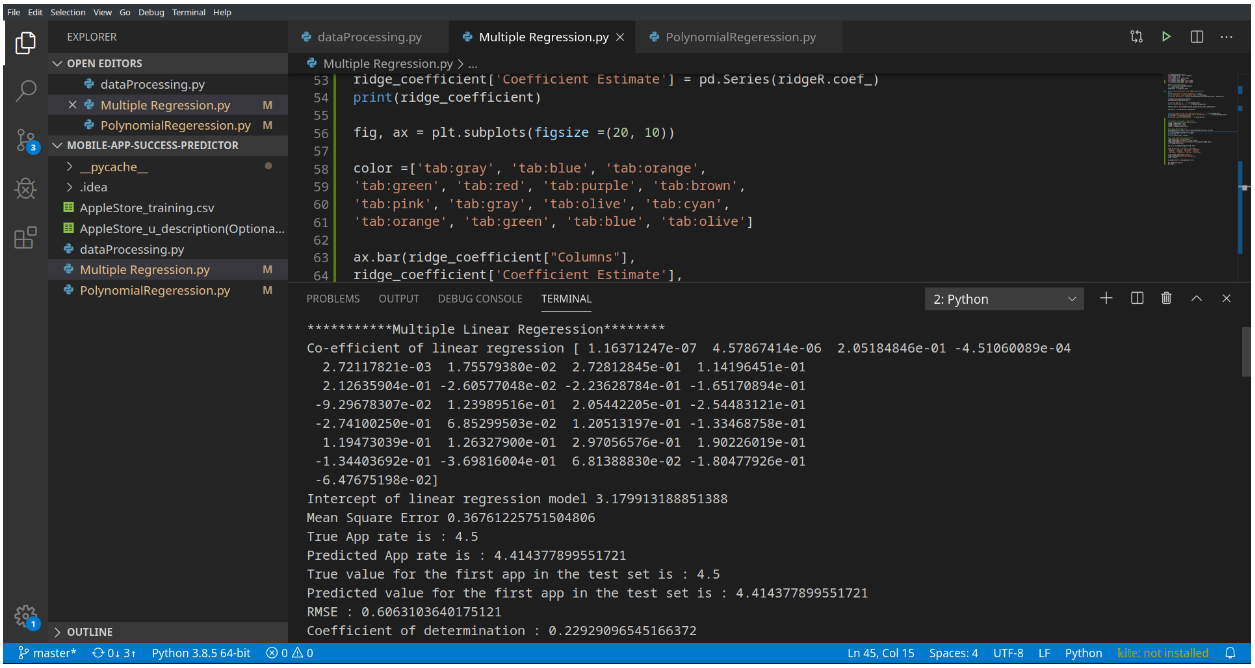
  We got a mean square error between 0.37652360484071495 and 0.35761225751504806

  The root of mean square error is between

  0.6136151928046721 and 0.5963103640175121

  The R2 score is between

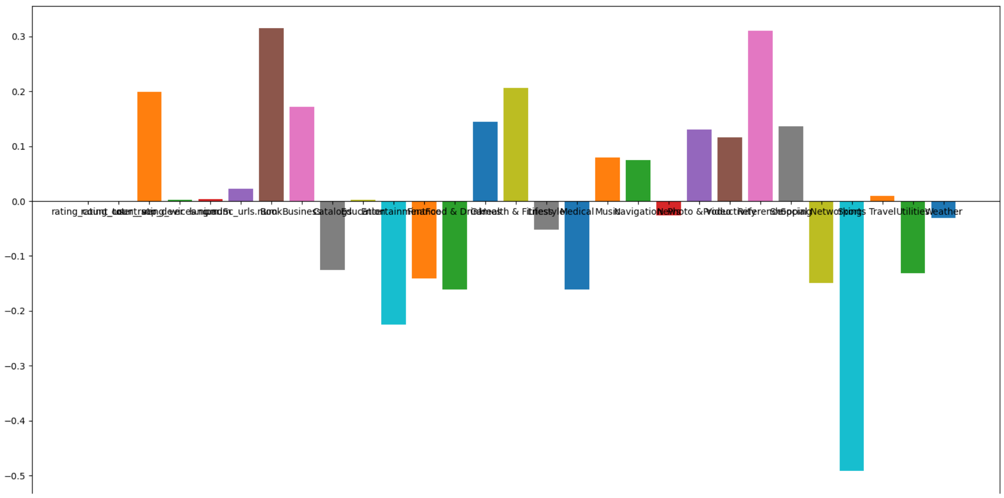
  0.22929096545166372 and 0.21844404907891346



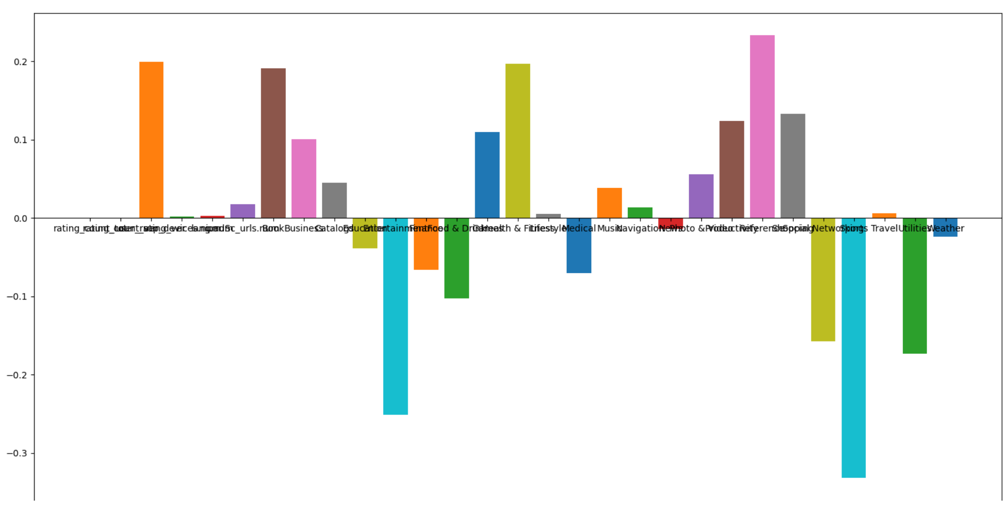
- ## L2 regularization

  Ridge Regression added a term in ordinary least square error function that regularizes the value of coefficients of variables. This term is the sum of squares of coefficient multiplied by the parameter The motive of adding this term is to penalize the variable corresponding to that coefficient not very much correlated to the target variable. This term is called L2 regularization.

alpha = 1



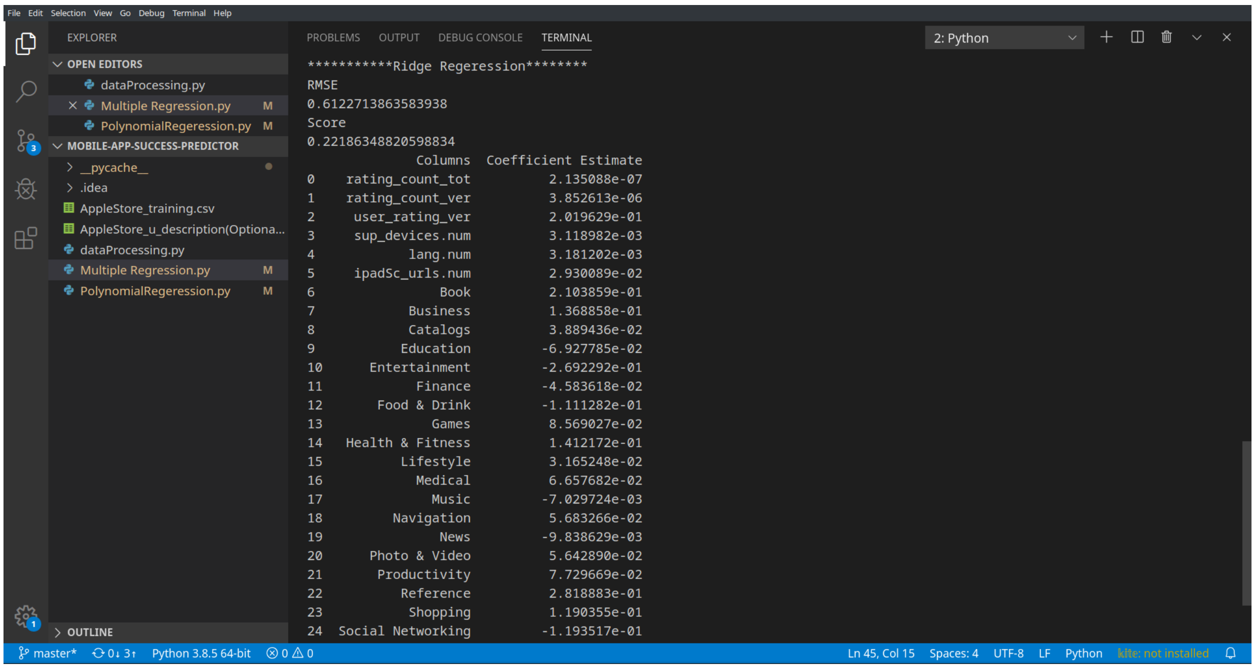
alpha = 10



As we can observe from the above plots that alpha helps in regularizing the coefficient and make them converge faster.

The score value is increased by (0.002 to 0.05)

And the error value is decreased by (0.01 to 0.03)



- **Polynomial Regression**

“Y=bo+b1X+b2X^2+……..+bnX^n”

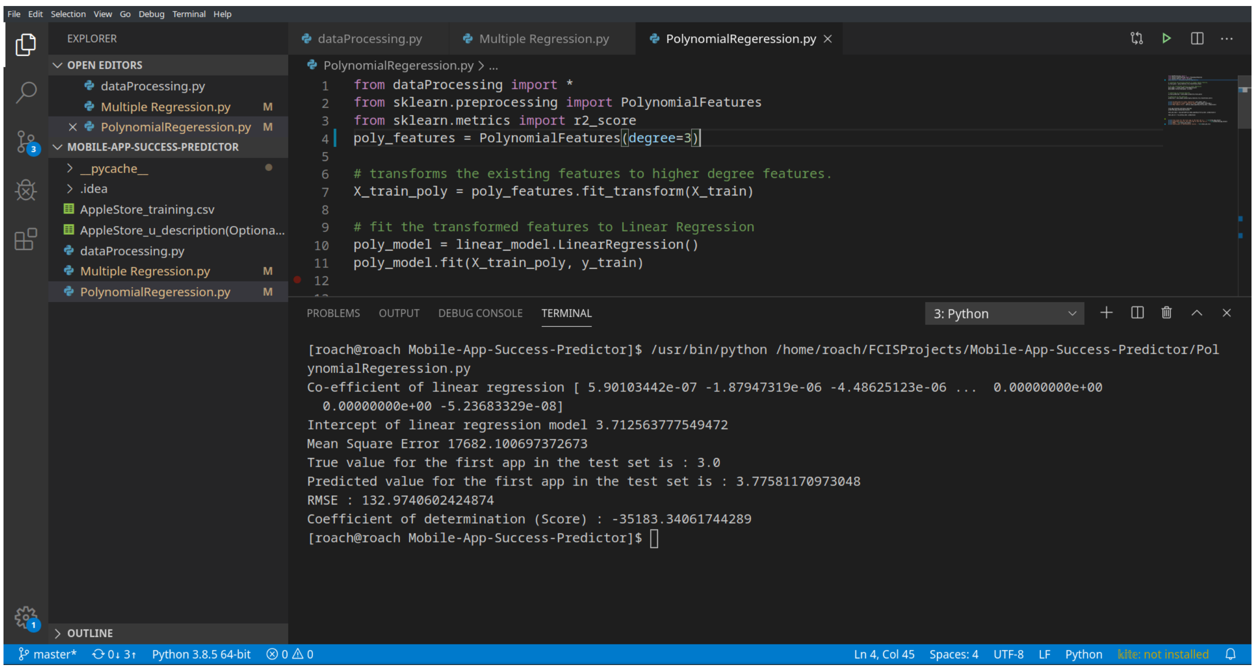
Polynomial regression is one of the types of linear regression in which the relationship between the independent variable x and dependent variable y is modeled as nth degree polynomial regression fits a nonlinear relationship between the value of x and the corresponding conditional mean of y,donated E(y|x).

Polynomial regression provides the best approximation of the relationship between the dependent and independent variable.

→ <u>First : Degree of 3</u>

Produced very high root mean square error

Resulting in overfitting.



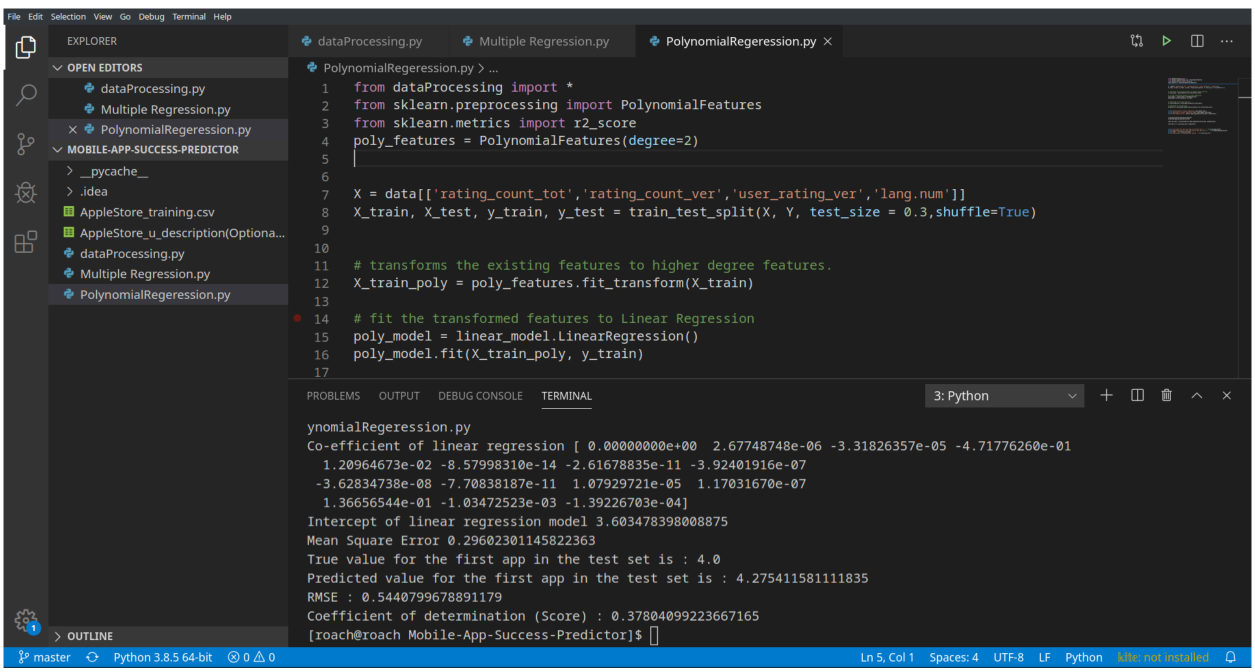
→ <u>Second : Degree of 2</u>

Produced lower root mean square error and higher R2 score than multiple linear regression.

Providing the better solution with this data.

Features used :

[rating_count_tot, rating_count_ver, user_rating_ver, lang.num]



<u>Time of training</u>

The time of training was between 7-10 seconds

Except the polynomial of degree 3 which took around 2-3 minutes.

<u>Conclusion</u>

- Time of training increases with the increase of the complexity.
- In higher polynomial degrees overfitting can occur.
- L2 regularization increases the accuracy of training.
- Data preprocessing and good feature selection leads to better results with better accuracy.

Machine learning

Milestone2 - Report

Data Preprocessing

1. preprocessing techniques

   a. Missing values:

      Dataset contained Some missing values so we dropped any row that contained at least one missing value, but we have done

```
#drop null rows
data.dropna(how='any',inplace=True)
```

      this after choosing the features that we are going to use.

   b. Text values:

      Since ML Algorithms can't read text value data, we need to convert any text data to numbers, so we have changed Prime_genre column to numeric values using One-Hot-Encoding technique.

```
#Onehot encoding X values
dummy=pd.get_dummies(Classify_X['prime_genre'],prefix="Genre",drop_first=False)
Classify_X=pd.concat([Classify_X,dummy],axis=1)
Classify_X=Classify_X.drop(['prime_genre'],axis=1)
```

c. Feature scaling:

Features values differ very much in ranges between them so som attributes would affect the output more than other attributes only because it has higher ranges, so it is important to apply feature scaling.

We applied min-max scaling on two columns ['rating_count_tot','rating_count_ver']

```
#feature scaling
scaler = MinMaxScaler()
data['rating_count_tot'] = scaler.fit_transform(np.array(data['rating_count_tot']).reshape(-1,1))
data['rating_count_ver'] = scaler.fit_transform(np.array(data['rating_count_ver']).reshape(-1,1))
#print("************* Scaling **************")
```

Features Used

Feature	Description
rating_count_tot	User Rating counts (for all version)
rating_count_ver	User Rating counts (for current version)
prime_genre	Primary Genre
sup_devices.num	Number of supporting devices
lang.num	Number of supported languages
ipadSc_urls.num	Number of screenshots shown for display

Classification Techniques

As we know that classification and regression are two major prediction problems which are usually dealt with Data Science and Machine Learning.

Classification is the process of finding or discovering a model or function which helps in separating the data into multiple categorical classes discrete values.

Regression is the process of finding a model or function for distinguishing the data into continuous real values instead of using classes or discrete values. It can also identify the distribution movement depending on the historical data

In classification , data is categorized under different labels according to some parameters given in input and then the lapels are predicted for the data.

• Logistic Regression

It’s a classification algorithm used to assign observations on a set of discrete classes , some examples of classification like email spam or not spam.

• Simple logistic regression =logistic regression with one

predictor variable.

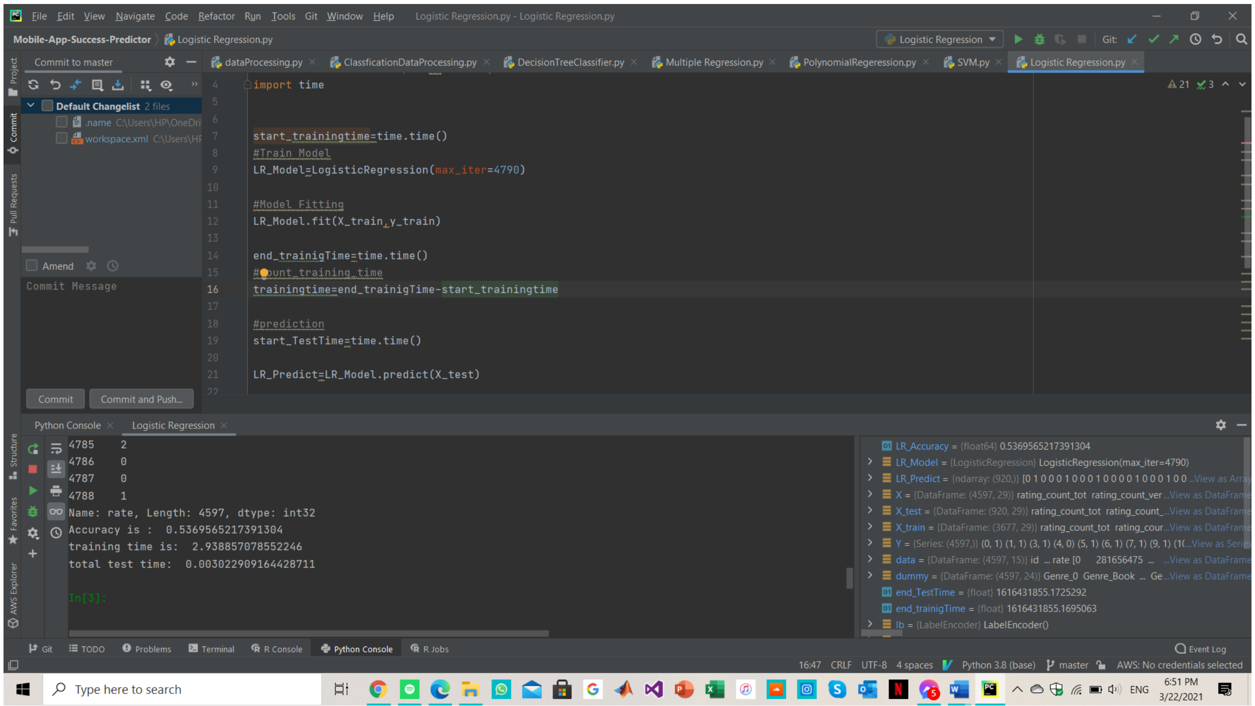
• Multiple logistic regression =logistic regression with multiple

predictor variables.

• And it also transforms its output using the logistic sigmoid

function to return a probability value.

Logistic Regression Results:



Bar Graph For Logistic regression:

Support Vector Machine

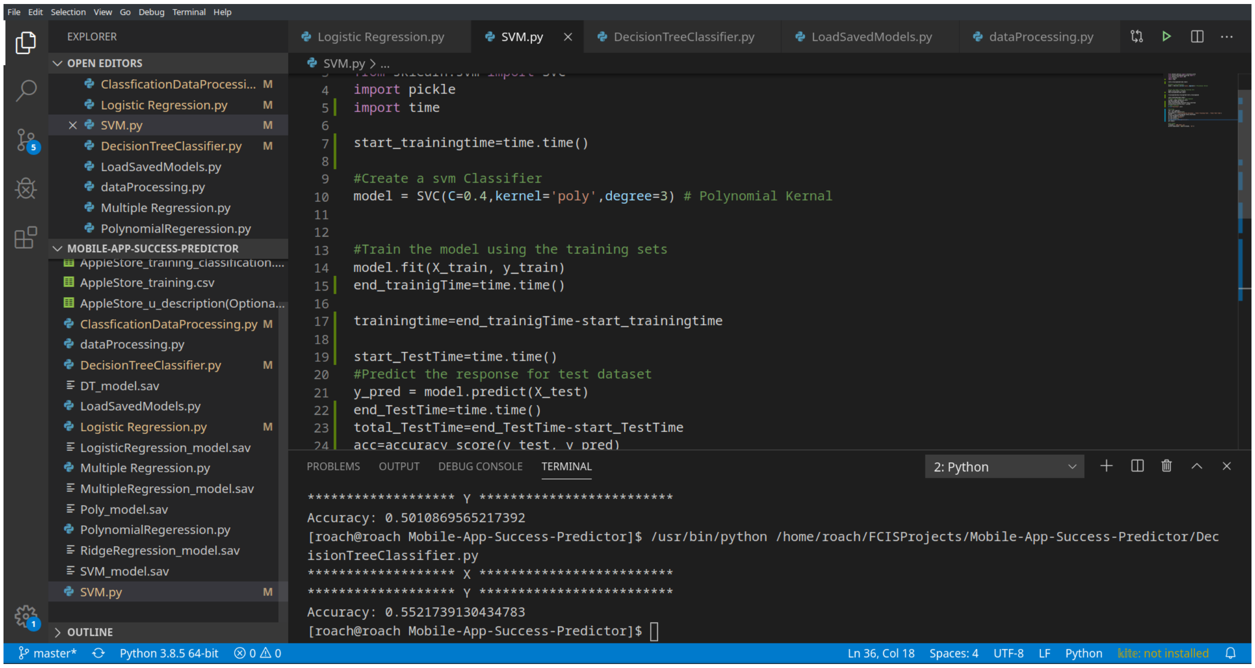
As we know SVM is supervised machine learning which can be used for both classification or regression problems. SVM is most used with linearly separable data.

It mostly used in classification problems, in this algorithm we plot each data item as a point in n-dimensional space (n-> number of features)

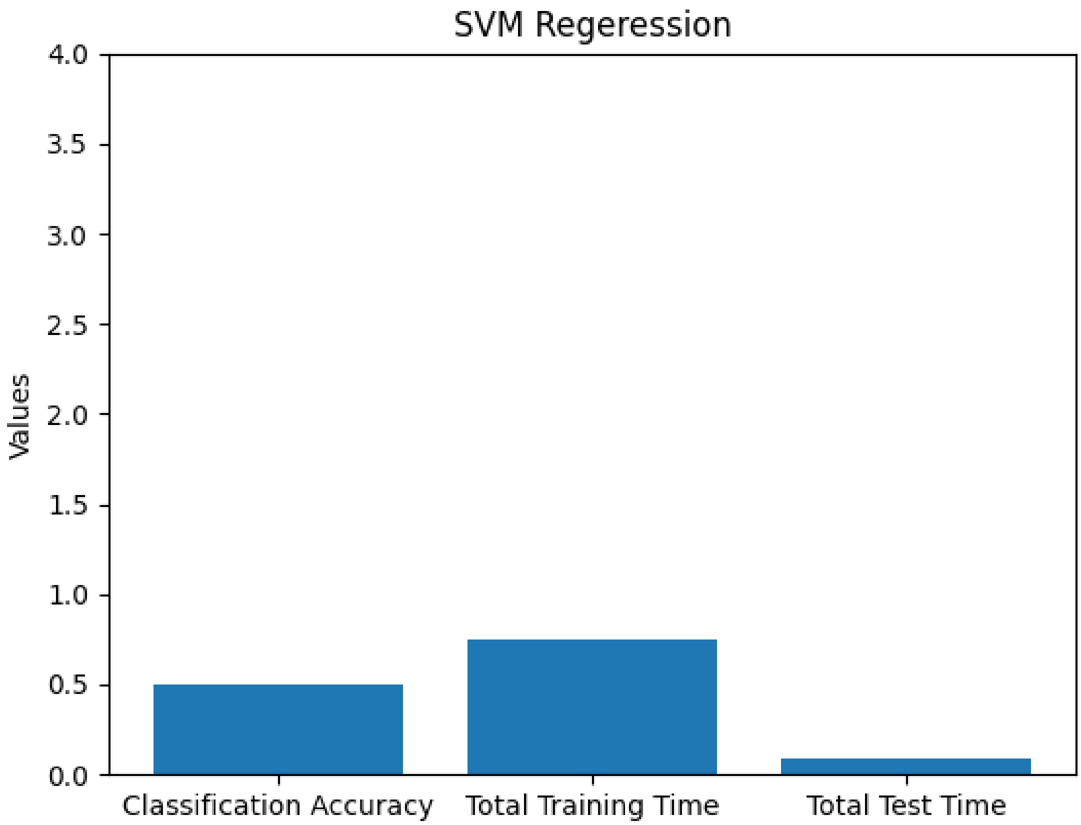
If the data isn’t linear we perform a transformation using Kernel functions.

In this model we used SVM with C=0.4 and Polynomial Kernel Function of degree 3.

SVM Results:


File Edit Selection View Go Debug Terminal Help
EXPLORER
OPEN EDITORS
ClassficationDataProcessi... M
Logistic Regression.py M
SVM.py M
DecisionTreeClassifier.py M
LoadSavedModels.py
dataProcessing.py
Multiple Regression.py
PolynomialRegeression.py
MOBILE-APP-SUCCESS-PREDICTOR
AppleStore_training.csv
AppleStore_u_description(Optiona...
ClassficationDataProcessing.py M
dataProcessing.py
DecisionTreeClassifier.py M
DT_model.sav
LoadSavedModels.py
Logistic Regression.py M
LogisticRegression_model.sav
Multiple Regression.py
MultipleRegression_model.sav
Poly_model.sav
PolynomialRegeression.py
RidgeRegression_model.sav
SVM_model.sav
SVM.py M
OUTLINE
Logistic Regression.py
SVM.py
DecisionTreeClassifier.py
LoadSavedModels.py
dataProcessing.py
SVM.py > ...
import pickle
import time
start_trainingtime=time.time()
#Create a svm Classifier
model = SVC(C=0.4,kernel='poly',degree=3) # Polynomial Kernal
#Train the model using the training sets
model.fit(X_train, y_train)
end_trainigTime=time.time()
trainingtime=end_trainigTime-start_trainingtime
start_TestTime=time.time()
#Predict the response for test dataset
y_pred = model.predict(X_test)
end_TestTime=time.time()
total_TestTime=end_TestTime-start_TestTime
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
2: Python
******************* Y ************************
Accuracy: 0.5010869565217392
[roach@roach Mobile-App-Success-Predictor]$ /usr/bin/python /home/roach/FCISProjects/Mobile-App-Success-Predictor/DecisionTreeClassifier.py
******************* X ************************
******************* Y ************************
Accuracy: 0.5521739130434783
[roach@roach Mobile-App-Success-Predictor]$
master* Python 3.8.5 64-bit 0 0
Ln 36, Col 18 Spaces: 4 UTF-8 LF Python kite: not installed

SVM Plotted Bar Graph:



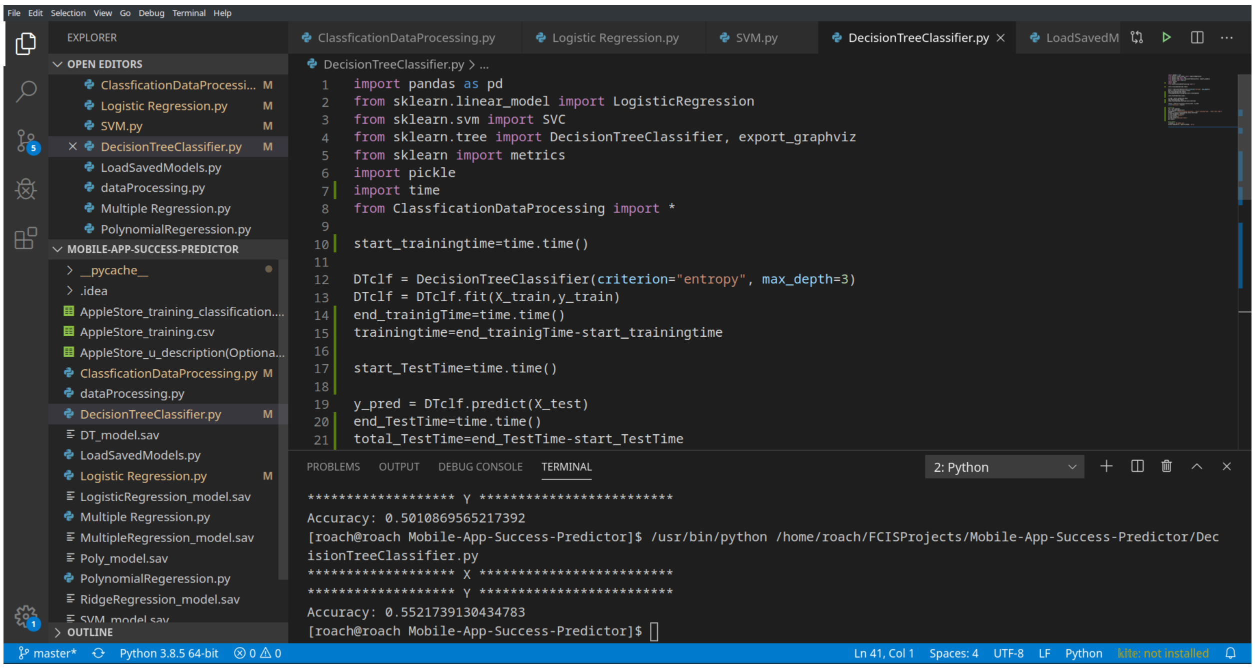
Decision Tree

Decision Tree Analysis is a general, predictive modelling tool that has applications spanning a number of different areas. In general, decision trees are constructed via an algorithmic approach that identifies ways to split a data set based on different conditions. It is one of the most widely used and practical methods for supervised learning.

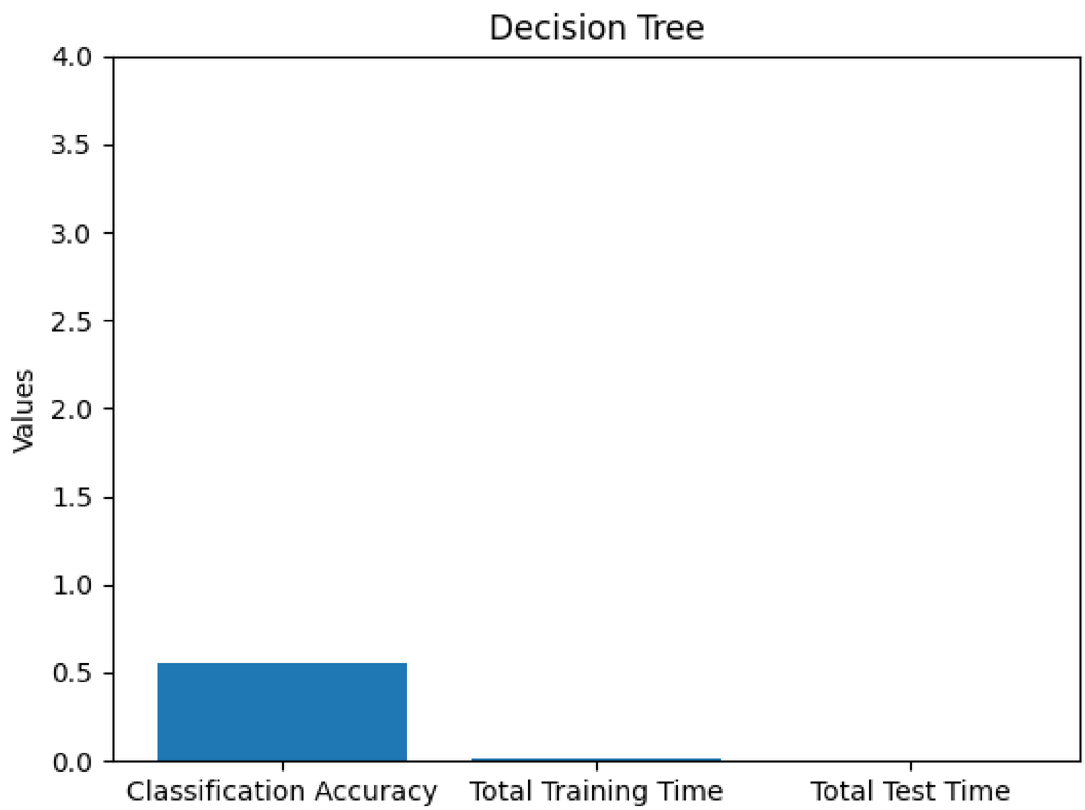
A decision tree is a tree-like graph with nodes representing the place where we pick an attribute and ask a question; edges represent the answers the to the question; and the leaves represent the actual output or class label. They are used in non-linear decision making.

In this model we used a decision tree with max depth of 3 which resulted in approximate accuracy to other models with much less training and test time resulting in much better performance.

Decision Tree Results:



Decision Tree Plotted Bar Graph:



Conclusion:

- SVM Try to maximize the margin between the closest support vectors.
- Logistic Regression maximizes likelihood.
- Data preprocessing and good feature selection leads to better results with better accuracy.
- Accuracy of the Logistic regression is slightly better than SVM in this Model but training time is much larger while logistic regression has less test time.

- Decision Tree accuracy is close slightly less but training and test time are way less than SVM and LR in this model.