Pattern Recognition Milestone2

CS_57 – Movies Popularity Prediction

Name	ID	Section	Department
مينا عادل لويز متى جرجس	**20201700894**	**9**	**CS**
مارلين تاوضروس يعقوب تاوضروس	**20201701119**	**6**	**CS**
مارتينا صبرى مسعد عجايبى	**20201700626**	**6**	**CS**
مينا نبيل اسعد نجيب	**20201700897**	**9**	**CS**
محمد تامر محمد محمدى	**20201700677**	**7**	**CS**

Preprocessing Techniques :-

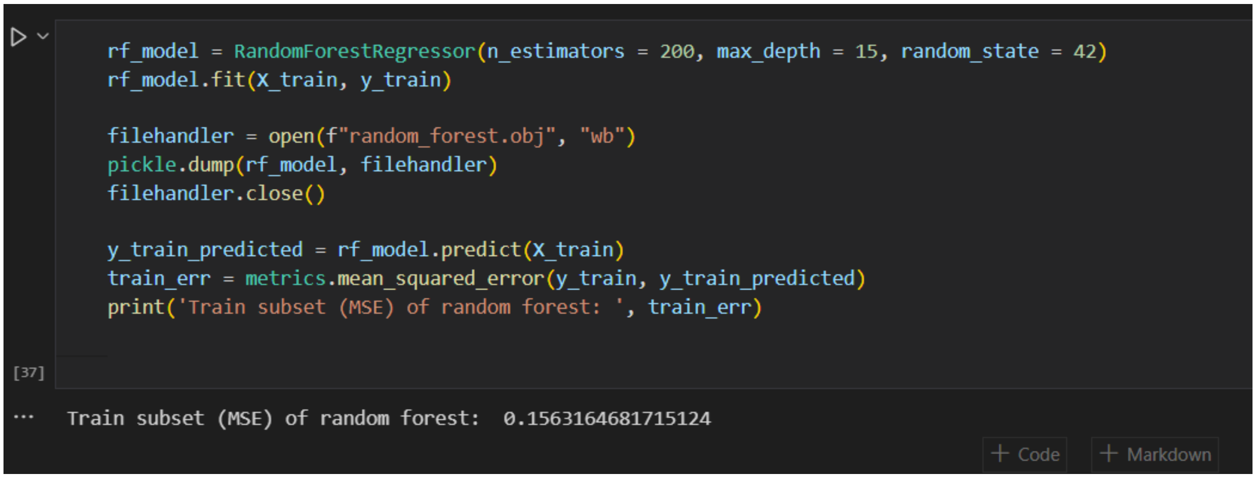
- We Applied **One hot Encoding** Technique on the following Columns **{Genres – Production Countries – Production Companies – Keywords – Spoken Language}** To determine the most effective values of these columns by separating each unique value into a new column and assign to it value = 1 in each row it appeared within otherwise the value = 0 so we can deal with categorical value in a numerical shape which make it more computable.
- We Applied **Feature Encoding** Technique on the following Columns **{Original Language – Original Title – Status}** To Convert Categorical Columns that have maximum of one value per record to numerical data so we can deal with it in a more computable way.
- We Applied **TF-idf Encoding** Technique on the following Columns **{Overview – Tagline}** to deal with them as they contain free text.

Some Analysis :-

- We Applied **Pearson's Correlation** On Numerical Columns **{budget viewercount - revenue - runtime - vote_count - year- month - day}** And we took values **greater than 0.35**
- As for The Preprocessed Categorical Columns we applied the **ANOVA TEST** to get the **P-Value** and keeping the columns that have P-Values less than 0.05
- We Applied **Scaling** using **MinMaxScaler** on Numerical Columns

Regression Techniques :-

- **Random Forrest** : This regressor fits multiple decision trees on randomly extracted subsets from the dataset and averages their prediction then we choose the number of estimators = 200 which means we use 200 decision tree which was suitable to our data , we also chose max depth = 12 to limit the number of levels to prevent **Overfitting**.

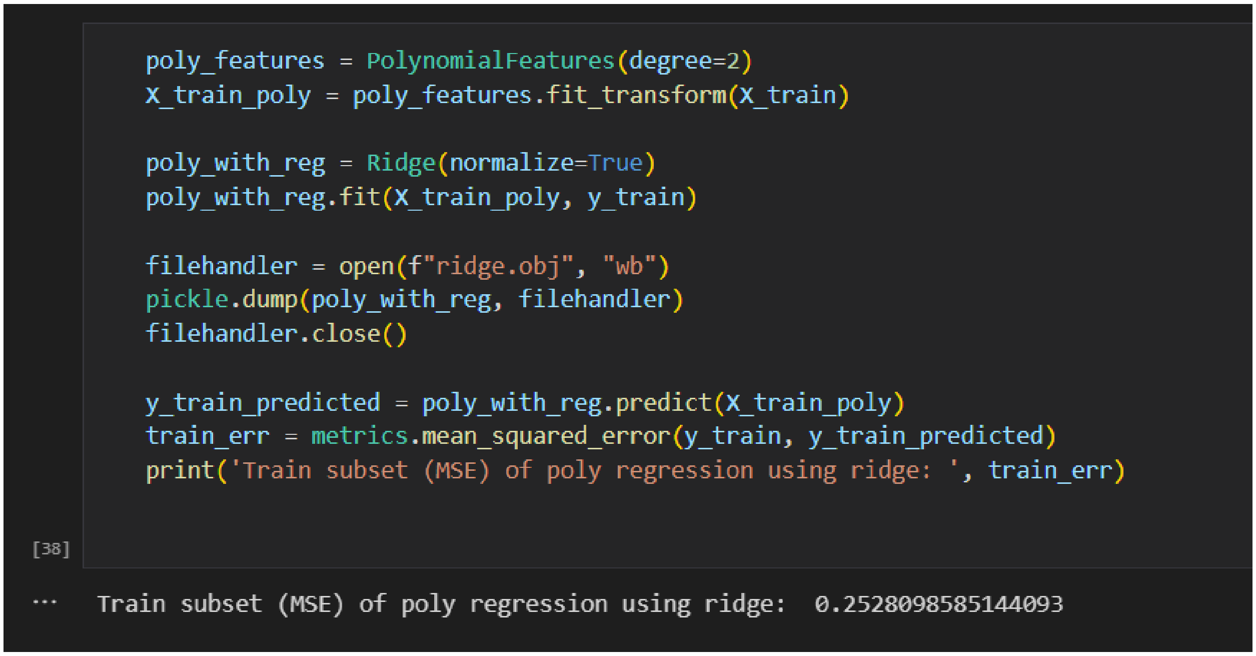


```
rf_model = RandomForestRegressor(n_estimators = 200, max_depth = 15, random_state = 42)
rf_model.fit(X_train, y_train)

filehandler = open(f"random_forest.obj", "wb")
pickle.dump(rf_model, filehandler)
filehandler.close()

y_train_predicted = rf_model.predict(X_train)
train_err = metrics.mean_squared_error(y_train, y_train_predicted)
print('Train subset (MSE) of random forest: ', train_err)
```

[37]

```
Train subset (MSE) of random forest:  0.1563164681715124
```

- **Polynomial Regression** : we used polynomial features to increase the degree of features to get more accurate results from our model.



```
poly_features = PolynomialFeatures(degree=2)
X_train_poly = poly_features.fit_transform(X_train)

poly_with_reg = Ridge(normalize=True)
poly_with_reg.fit(X_train_poly, y_train)

filehandler = open(f"ridge.obj", "wb")
pickle.dump(poly_with_reg, filehandler)
filehandler.close()

y_train_predicted = poly_with_reg.predict(X_train_poly)
train_err = metrics.mean_squared_error(y_train, y_train_predicted)
print('Train subset (MSE) of poly regression using ridge: ', train_err)
```

[38]

```
Train subset (MSE) of poly regression using ridge:  0.2528098585144093
```

Comparison between Models :-

Result / Model	**Random Forrest**	**Polynomial Regression**
Train MSE	0.1563164681715124	0.2528098585144093
Test MSE	0.2316859812514649	0.3175056876194303
R^2	0.7148543648768815	0.6092324599770211

Discarded Features :-

- **Home Page :** discarded because it contains many NULL Values (~=50% or more of the data).
- **ID :** discarded because it is irrelevant column.
- **Title :** discarded because it is a duplicate of the Original Title.
- **Any other feature selection was handled in the analysis section.**

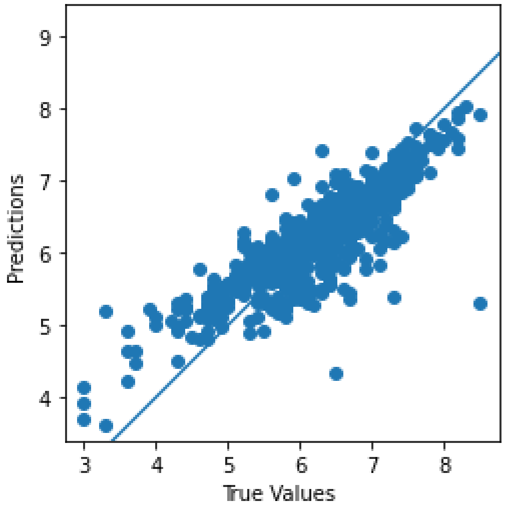
Important sizes : **Train size = 80% , Test size = 20% , Validation = 0%.**

Other Improvements :-

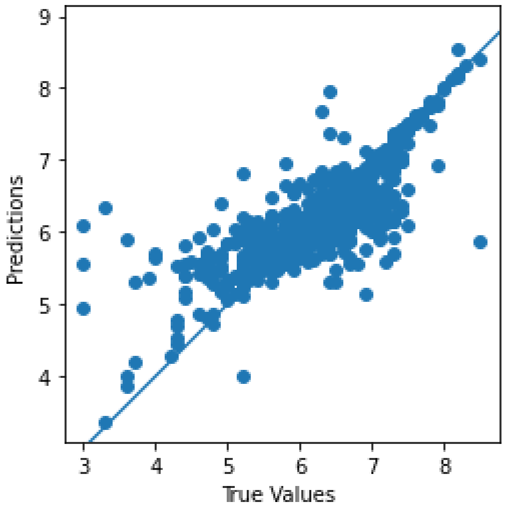
- We dropped the resulting columns from **one hot encoding** that had a sum of 1 or 2 .
- We used **Regularization** on **Polynomial Regression** with **Ridge** and we used the Parameter **Normalize = True** to scale the data to be faster and more accurate.

Some Important Plots :-

- **Random Forrest :-**



- **Polynomial Regression :-**



Conclusion : As for this milestone of the project, it was clearly obvious from the beginning that it is hard to predict an accurate prediction for Movie Rating based on the information given in the data set , lots of Columns were irrelevant and even the relevant ones were with weak relation with the target column, and our intuition was proved by the Correlation at first and then in the results of the models it was clearly proved with the R^2 Score that these models are not so reliable and hard to get accurate prediction based on the given data.