APPLIED DATA SCIENCE- GROUP 3

PROJECT 2 : PREDICTING IMDB SCORES

ABSTRACT

PHASE 3 :

- DATA PREPROCESSING
  - LOAD DATA IN PANDAS.
  - DROP COLUMNS THAT AREN'T USEFUL.
  - DROP ROWS WITH MISSING VALUES.
  - CREATE DUMMY VARIABLES.
  - TAKE CARE OF MISSING DATA.
  - CONVERT THE DATA FRAME TO NUMPY.
  - DIVIDE THE DATA SET INTO TRAINING DATA AND TEST DATA.

PREDICTING IMDB SCORES

PHASE 3 : PREPROCESSING

DatasetLink: https://www.kaggle.com/datasets/luiscorter/netflix-original-films-imdb-scores

Preprocessing an IMDb score dataset for use with a Random Forest Regression model involves preparing the data and then applying the model. Here are the steps you should follow:

Load and Prepare the Dataset:

Load the IMDb score dataset, as described in the previous answer, and perform the necessary preprocessing steps, including data cleaning, feature engineering, and data transformation. Ensure that the target variable (IMDb scores) and the features (independent variables) are appropriately defined.

Split the Dataset:

Split the dataset into training and test sets. The training set is used to train the Random Forest Regression model, and the test set is used to evaluate its performance. Common splits are 70-30 or 80-20.

Feature Selection (if necessary):

Depending on the size and quality of your dataset, you may want to perform feature selection to identify the most relevant features for your model. You can use techniques like feature importance from the trained Random Forest model or other feature selection methods.

Train the Random Forest Regression Model:

Use the training dataset to train the Random Forest Regression model. You can do this using a machine learning library like scikit-learn in Python. Here's a simple example:

python

Copy code

```
from sklearn.ensemble import
RandomForestRegressor

# Instantiate the Random Forest Regression model
rf_model = RandomForestRegressor(n_estimators=100,
random_state=42)

# Fit the model to the training data
```

rf_model.fit(X_train, y_train) # X_train is your feature matrix, y_train is the target variable (IMDb scores)

Adjust the hyperparameters like n_estimators and others based on your specific requirements.

Evaluate the Model:

Once the model is trained, evaluate its performance using the test dataset. Common regression evaluation metrics include Mean Absolute Error (MAE), Mean Squared Error (MSE), and Root Mean Squared Error (RMSE). You can also calculate the R-squared (R^2) score to understand how well the model fits the data.

python

Copy code

```
from sklearn.metrics import mean_squared_error, r2_score

# Make predictions on the test set
y_pred = rf_model.predict(X_test)  # X_test is your test feature matrix

# Calculate Mean Squared Error
mse = mean_squared_error(y_test, y_pred)

# Calculate R-squared (R^2) score
r_squared = r2_score(y_test, y_pred)
```

```
print(f'Mean Squared Error: {mse}')
print(f'R-squared (R^2) Score: {r_squared}')
```

Tune the Model (Optional):

You can fine-tune the Random Forest Regression model by adjusting hyperparameters, such as the number of trees, depth of trees, and other parameters. Grid search or randomized search with cross-validation can help find the best hyperparameters for your dataset.

Make Predictions:

Once you are satisfied with the model's performance, you can use it to make predictions on new data. Simply provide the model with the features of the new data, and it will return IMDb score predictions.

Save the Model (Optional):

If you intend to use the model in the future, you can save it to a file using joblib or pickle in Python.

Remember that the preprocessing steps you perform on your IMDb score dataset are crucial for the success of your Random Forest Regression model. The model's performance greatly depends on the quality of your data and feature engineering.

https://drive.google.com/drive/folders/1nH8msCJw8kcGPmD4bKjS7GR-3j8WN8Z3?usp=drive_link