Pattern Recognition Project

Movie Popularity Prediction

Milestone 2

- Team Members:

نادين مصطفى علي أحمد
20201700902
sec 9

نانسي ابراهيم عطية ابراهيم
20201700906
sec 9

نوران وليد محمد أحمد المرسي
20201700936
sec 10

عبد الله يحيى سليمان يحيى
20201700466
sec 4

عبد العزيز حسام عبد العزيز عبد الفتاح
20201700457
sec5

عبد الرحمن محمد نجيب محي الدين محمود
20201700449
sec 5

- Preprocessing:
    1. Applying preprocessing on numerical columns ‘budget’, ‘‘viewercount’’, ‘revenue’, ‘runtime’, ‘votecount’ and ‘releasedate’ by replacing any 0 value with the mean of each column. Also scaling has been made on these columns by subtracting the minimum value of each column from the corresponding column and then dividing it by the range of each column to transform the values in each column to a range between 0 and 1.

    2. The next application of preprocessing was encoding some columns, we’ve worked with ‘homepage’ column, ‘status’ column and ‘original language’ column. We have checked for nulls and returned binary values 0 for false and 1 for true for ‘homepage’. In the same function we’ve used label encoder to encode the columns ‘original language’ and ‘status’ into numerical.

    3. In the columns that are text data ‘overview’, ‘title’, ‘tagline’ and ‘original title’, we’ve made some NLP techniques like making all words lowercase, tokenization, removing stops words, lemmatization, and removing non-alpha numeric characters. Then we’ve used TfidfVectorizer for converting text data into numerical representation, where the importance of each word is based on their frequency and rarity across the data, where every word has a weight. For the training data, the vectorizer transforms and fits, while in the test data the vectorizers only transforms.

    4. On the columns ‘genres’, ‘keywords’, ‘production companies’, ‘production countries’, ‘spoken languages’, that contain lists of dictionaries Columns are converted from lists of dictionaries to comma separated strings. We’ve used MultiLabelBinarizer to transform these columns into binary-encoded representations.

We have used a global MultiLabelBinarizer to use in two functions, ‘dictionaryPreprocessing’ one for the training data that contains fit and transform, and ‘dictionaryPreprocessing_test’ for the test data that contains only transform

5. For representing the ‘release date’ column, we’ve updated it with time differences in days to be represented as a numerical feature. We’ve calculated it by subtracting the two dates ‘release date’ and ‘reference date = datetime.now’.

6. After reading data from the csv file, we’ve dropped duplicates from our dataframe.

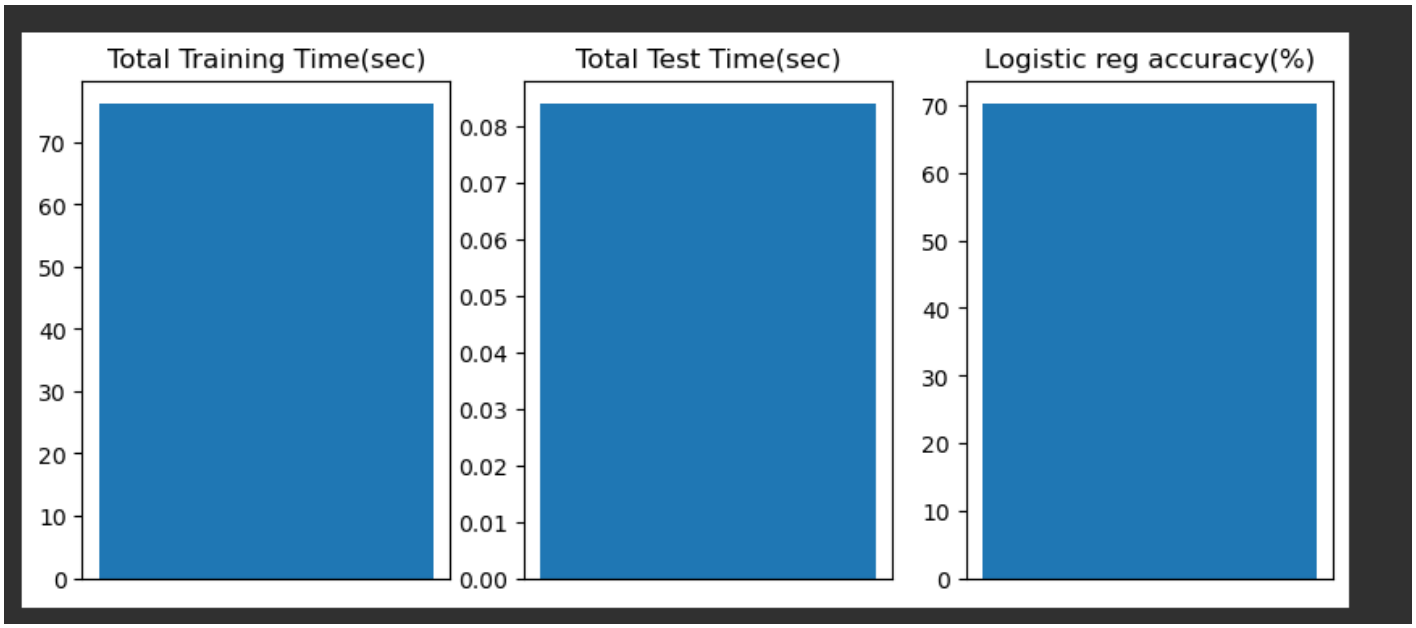
- Feature Selection:
  In the feature selection we've used 'VarianceThreshold' feature selection technique to remove the columns with low variance from the preprocessed data. By comparing the model performance metrics such as accuracy or other relevant metrics, the effect of this feature selection is way better.

- Before training the models, we first need to create a dictionary to map categories to numbers.

- Models training:
  We've trained three different models to classify each sample into distinct classes:
  1. Logistic Regression:

```
log_model()
```

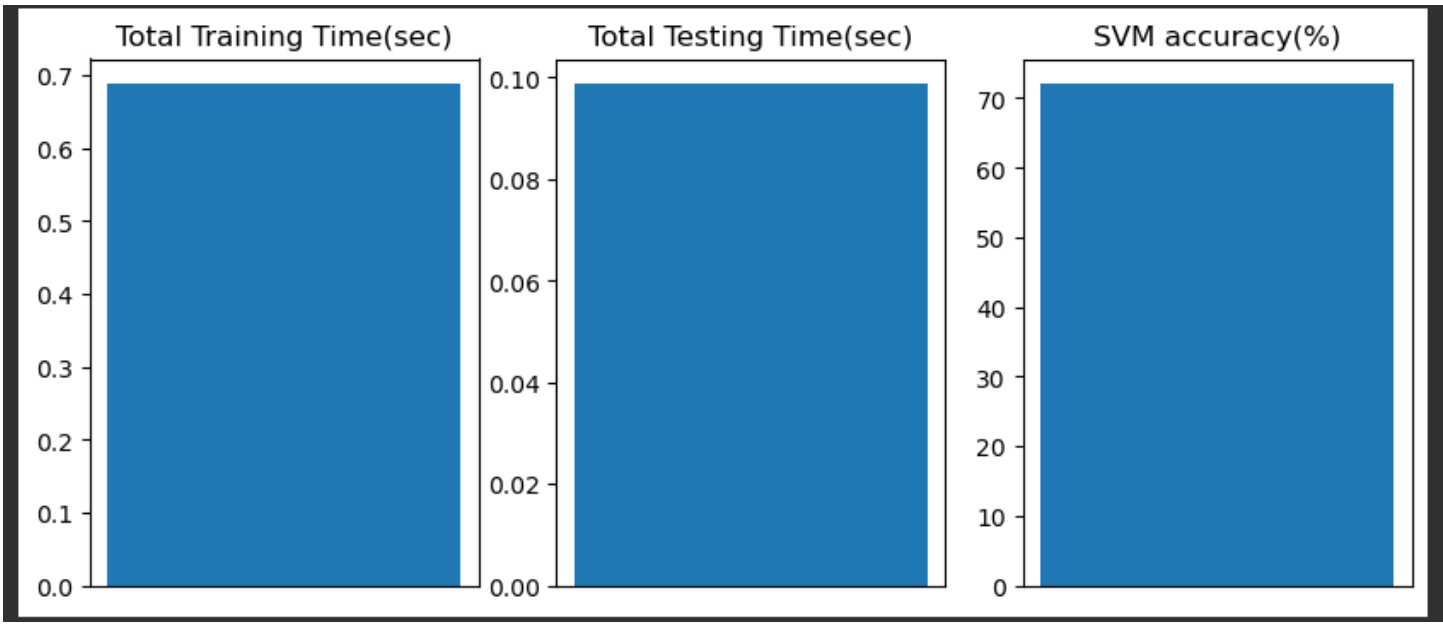
```
Creating and training a new model
Model training started
LOGISTIC
trianing time: 76.18266797065735
test time: 0.08399271965026855
Accuracy: 0.7023026315789473
```



2. SVM:

```
svm_model()
```

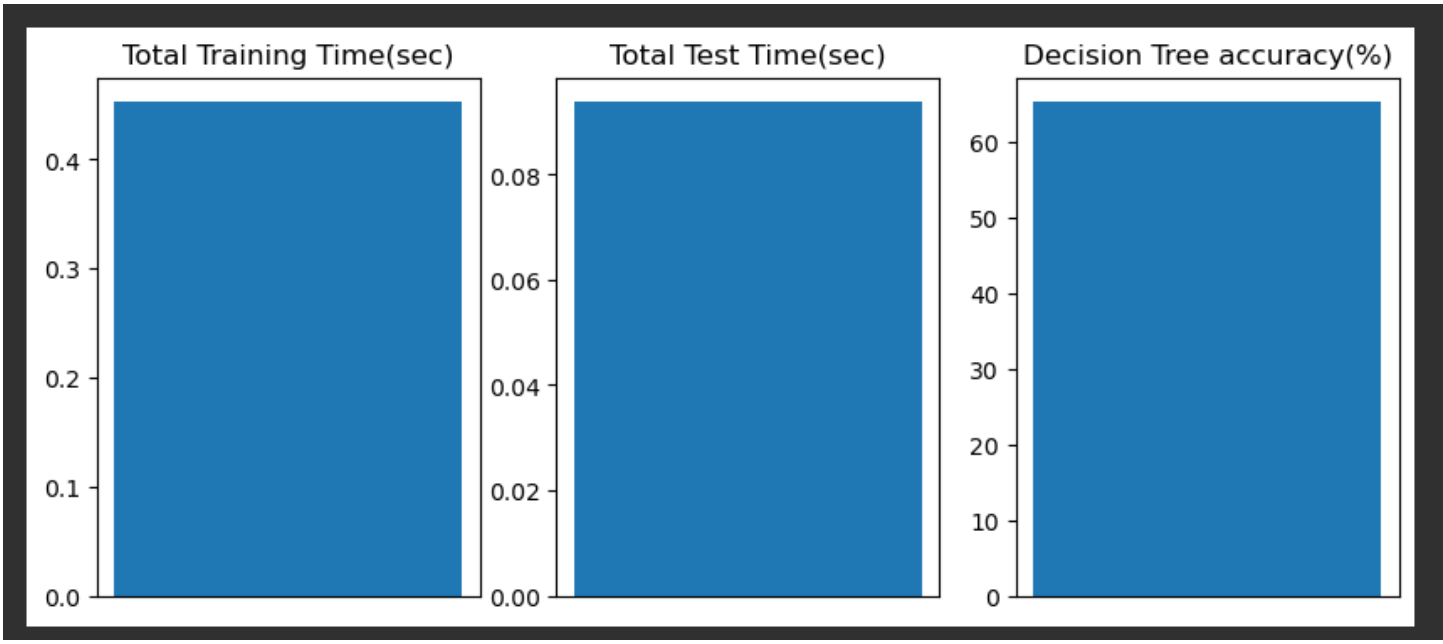
```
Creating and training a new model
Model training started
SVM
trianing time: 0.6881892681121826
test time: 0.09873557090759277
SVM accuracy:  0.7203947368421053
```



3. Decision Tree:

```
tree_model()
```

```
Creating and training a new model
Model training started
Decison Tree
trianing time: 0.4527871608734131
test time: 0.0937509536743164
Accuracy: 0.6529605263157895
```



- After training our model and trying different hyper parameters, this is the best performance we've reached.

- Conclusion:
  During the classification phase, we trained various machine learning classification algorithms, such as KNN, SVM, Naïve bayes, Random Forest, etc. These models utilized the features we selected to make predictions about movie popularity.

  It is important to note that the prediction accuracy varied across different models, and no specific model outperformed the others. This indicates that movie popularity prediction task requires careful consideration of model selection and tuning.

  Our intuition about the factors that are important for movie popularity prediction were confirmed through the analysis of dataset and working on the preprocessing and feature selection.