Report

Data Analysis Project

IMDB MOVIE GENRE PREDICTION

Zahra Reihanian	810101177
Fatemeh Taherinejad	810101219
Mohammad Mahdi Mohajeri	810101355
Fatemeh Nadi	810101285

January, 2023

Contents

1 Chosen Website **1**

2 Data Crawling **2**

3 Data Cleaning and Preprocessing **3**
3.1 Question 1 3
3.2 Question 2 7

4 Visualization and EDA **8**
4.1 Question 1 8
4.2 Question 2 9

5 Feature selection and dimension reduction **17**
5.1 PCA 17
5.2 LDA 18

6 Classification Methods **19**
6.1 Question 1 19
6.2 Question 2 24
6.3 Question 3 25
6.4 Question 4 - Extra Part 27

7 Model improvement **31**

8 References **34**

1 Chosen Website

Predicting movie genres based on other features of a movie, can come in handy for recommender systems and online VOD services. The chosen website to crawl and extract information from, is the IMDB Top 1000 movies. This site contains the following information of the top rated 1000 movies:

- Name- title of the movie in English
- Year of production
- Certificate- the age rating of the movie
- Duration in minutes
- Votes- number of votes in the IMDB website
- Metascore- movie score on the Metacritic website
- Director of the movie
- 4 top stars
- Summary- a short explanation of the movie story
- Genre

Based on past experience and knowledge, certificate, duration, scores, director and actors can be used to predict the genre of a movie. Therefore, it is to be tested if genre is predictable based on the information gathered from the website. It is worth to mention that summary is probably a very good feature for genre prediction according to the words it contains but it is first tried to predict the genre without summary. In the second step, a network is used to extract features from summary and the features are combined to build new models.

2 Data Crawling

To extract the information mentioned above from the website, beautiful soap can be used as the website allows crawling there is no need to scape the website with interaction. The page is first inspected to detect where each element is written to and then extracting each element is easy to obtain with just a few lines of code.
First, all divisions containing a movie are collected and with iterating through each, different pieces of relevant information are collected. Each piece of required information is specified with its unique tag is and name and after extracting the main information, little preprocessing is done in some fields and the results are saved to a csv file for later uses. It must be mentioned that fields with missing values are filled with null and will then be dealt with in the data cleaning step.

3 Data Cleaning and Preprocessing

3.1 Question 1

Every piece of collected data can contain missing values and concepts. By investigating the dataset, it was observed that for some movies, certificate values are not collected and there are also some missing values for Metascores.
In order to obtain better data, the website was once again crawled with the help of selenium and using element paths.
This data was not perfect either and after inspection of each part of both datasets, it was decided to combine the two and use best parts of each. This is the first step of data cleaning.

```
print("#null values in df1:\n", df.isna().sum(axis=0))
```

```
#null values in df1:
 Name            0
Date             0
Certificate    450
Duration         0
Votes            0
IMDB             0
Metascore      160
Director         0
Star1            0
Star2            0
Star3            0
Star4            0
Genre            0
dtype: int64
```

Figure 1: missing values of the first dataset

To see which column is best represented in each dataset, unique values of columns are printed and compared. Feature with the greatest number of missing values was certificate with a total of 450. Unique values of certificate were inspected in both datasets and it was found that there are some irrelevant values in the second dataset and some rows that do have a rating in the second dataset that are missing in the first dataset. It must also be mentioned that for this particular feature, there are some movies that are not rated in the website at all.
These movies are mostly from countries other than America and European countries. Correct values were replaced in the first dataset.

```
df['Certificate'].unique()
```

```
array(['PG-13', 'R', nan, 'PG', 'G', '18+', '(Banned)', 'NC-17',
       'Approved', 'M/PG'], dtype=object)
```

```
list1 = ['148 min','163 min','175 min', '157 min', '152 min', '109 min', '128 min']
```

```
for index, row in df2.iterrows():
    if (row['certificate'] not in list1):
        df.loc[df['Name'] == row['title'], 'Certificate'] = row['certificate']
```

```
df['Certificate'] =df['Certificate'].replace(['Unrated','6','12','18','16','(Banned)'],
                                             ['Not Rated','6+','12+','18+','16+','Banned'])
```

```
df['Certificate'].unique()
```

```
array(['PG-13', 'R', 'PG', 'G', nan, 'Not Rated', 'Approved', 'Passed',
       'NC-17', 'TV-MA', 'GP', '18+', 'TV-PG', 'TV-14', 'M/PG', '12+',
       'Banned', '16+', '6+'], dtype=object)
```

Figure 2: cleaning certificate column

Certificate values were also checked and some values were replaced with standard notations. After these steps, there exists 125 rows with null as the certificate value. To fill these values, it was first observed which certificate is most common in each genre group and also in the entire dataset. It was observed that "R" is the most repeated in most genres and with further searches over the internet, it was decided to fill null values with "R". This decision was made based on the fact that movies with "R" certificate are not only the majority in the dataset but also it is safer for movies to be rated as "R" according to its meaning.

```
df.groupby('Genre')['Certificate'].agg(pd.Series.mode)
```

```
Genre
Action                              R
Adventure                          PG
Animation                          PG
Biography                           R
Comedy                              R
Crime                               R
Drama                               R
Family                              G
Fantasy                [Not Rated, R]
Film-Noir          [Approved, Passed]
Horror                              R
Mystery                      [PG, R]
Thriller                   Not Rated
Western       [PG, PG-13, Passed, R]
Name: Certificate, dtype: object
```

Figure 3: mode of certificates in each genre

Before grouping by genre, unique values of it were observed and it was found that there are some extra spaces in the name of some genres which makes them incompatible and categorized as another unique class. Therefore, stripping is done on the column and grouping is done based on the cleaned values. As mentioned before, another feature with missing values in the Metascore value which is numerical and dealing with missing values of a numerical variable can be much easier. To find a meaningful value to fill missing values with, its distribution is plotted in a histogram and as can be seen in the below figure, the distribution is left skewed and therefore the mean can not be a good representative to fill the missing values with.

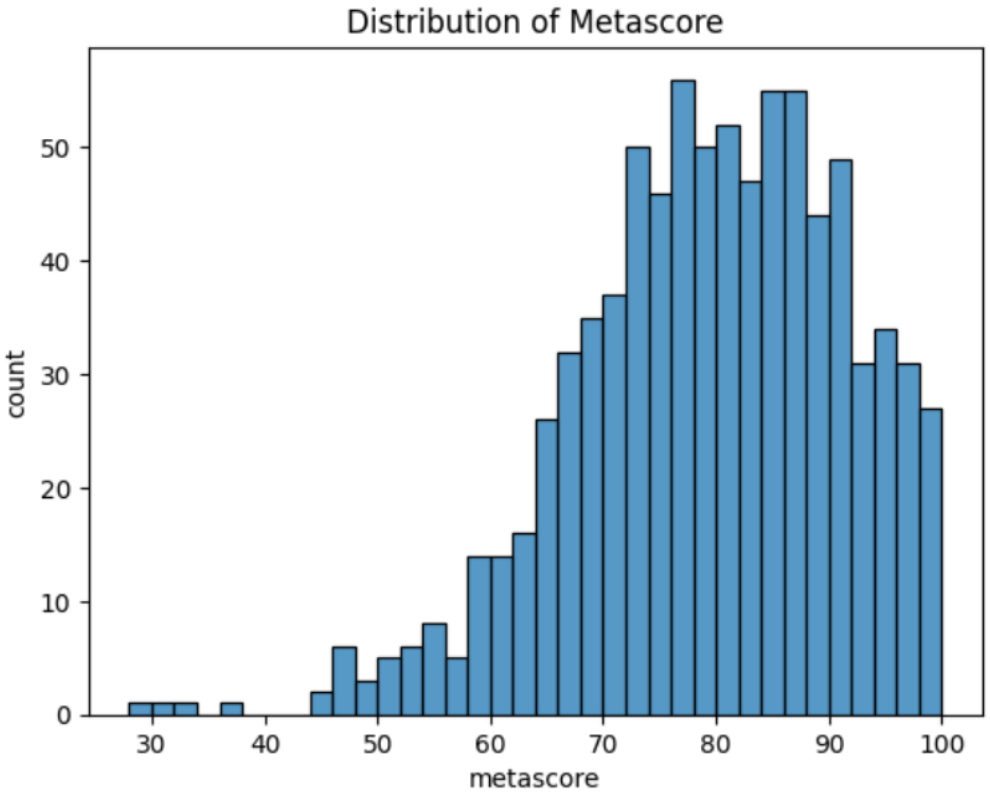

Figure 4: distribution of Metascore values

It was guessed that Metascore can somehow be correlated with IMDB score and there might be a regression formula to fill the values of Metascore with the corresponding IMDB value. To inspect the guess, scatter plot of Metascore is plotted according to IMDB score and surprisingly, there is no meaningful correlation between the two variables. The resulting plot is shown in figure 5. Mean, median and mode of the column is printed and according to the reasons mentioned, median is used to fill the missing values.

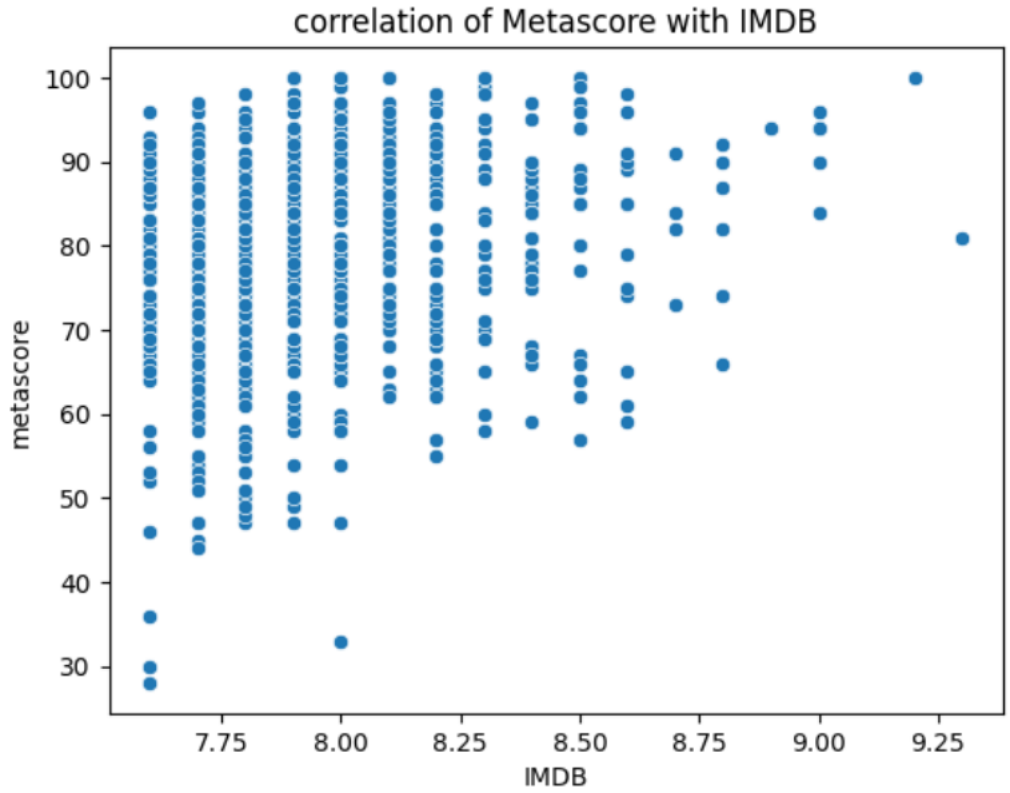

Figure 5: scatter plot of IMDB and Metascore

Next step taken is to change data types for memory efficiency reasons and also to make it easier to work with the dataset. With observing the values, integer values of votes, year and both scores fall in ranges that data type can be reduced. Also, genre and certificate are changed to category as there are only a few categories that repeated many times. There is no point of changing the data types of stars and director as there are many common names and categorizing them will not change the taken space much.

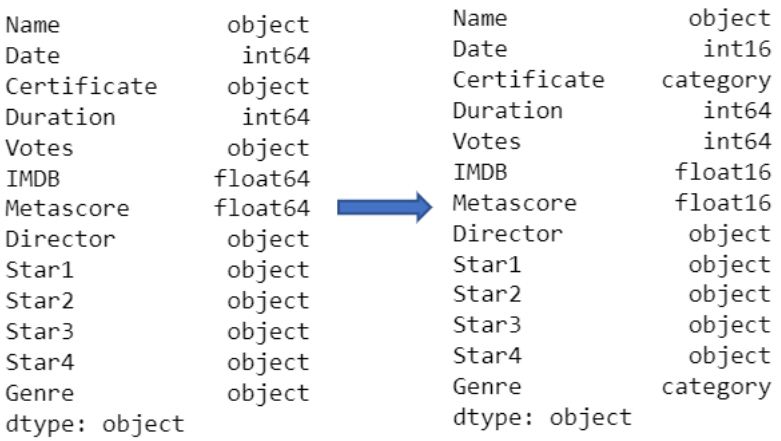

Figure 6: data type conversion

3.2 Question 2

Last but mot least, number of records of each genre is inspected and as can be seen in figure 7, classes are not totally balanced. Unfortunately, as dataset is not huge, records can not be deleted to balance the dataset and type of data is not some kind to generate new records easily. Therefore, it was decided to combine the minor classes with each other into a new class as others. This will not balance the data fully, but it is still better than nothing. It is also bore in mind that with this class distribution, classifier accuracies will be lower and not as good as it would have been if there existed more data on minor classes.

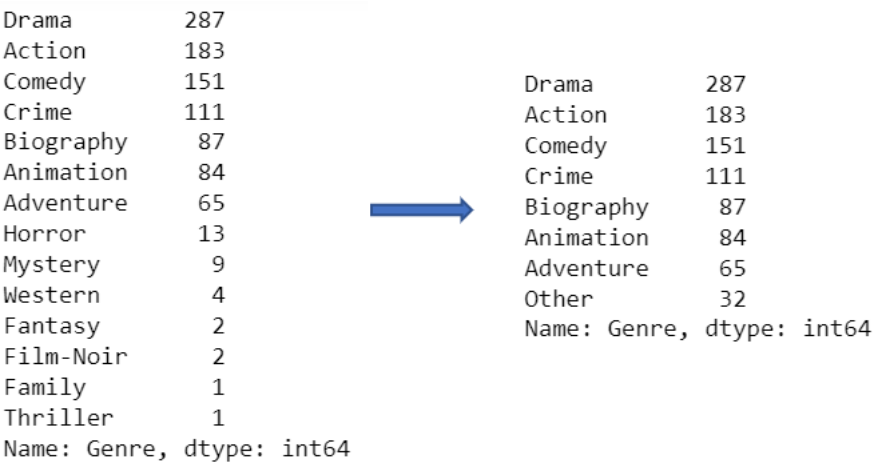

Figure 7: class balancing

4 Visualization and EDA

4.1 Question 1

In order to obtain features with higher importance in model, RandomForestRegressor is used from the sklearn library. To be able to use the function, categorical values must be encoded and to numbers. All variables are turned to categorical values and then coded using .cat.codes. features and labels are separated and given to the model. Output results are shown in figure 8 and indicate surprising results. It was expected that certificate would be important in predicting genres as there are certain certificates for certain types of movies, but as can be seen in the plot, it is the least important feature of all. Surprisingly, three of the stars are more important than other values. Overall, it can be indicated from the plot that these features are not very good in helping to predict the genre of a movie as almost all features have an importance of less than 0.1. The results are an introduction to getting a not so good classifier. Top 4 important features will later be used in the classification step to see how accuracy changes with eliminating less important features from the dataset.

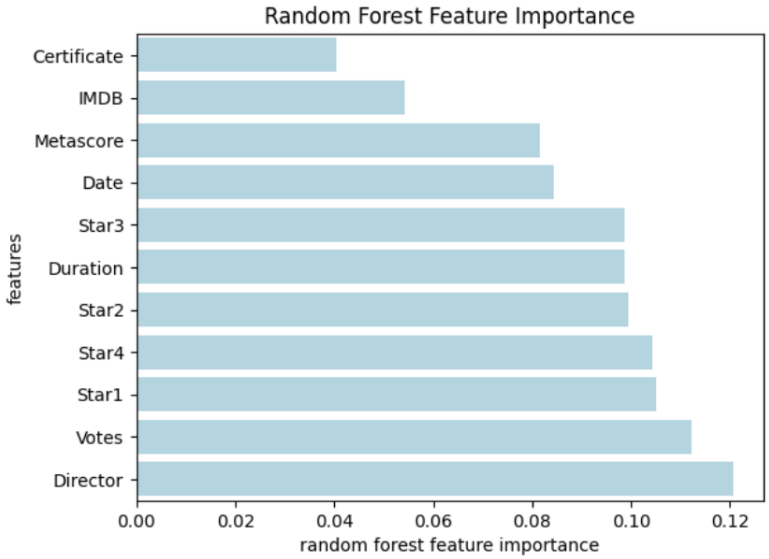

Figure 8: feature importance

Genres distribution

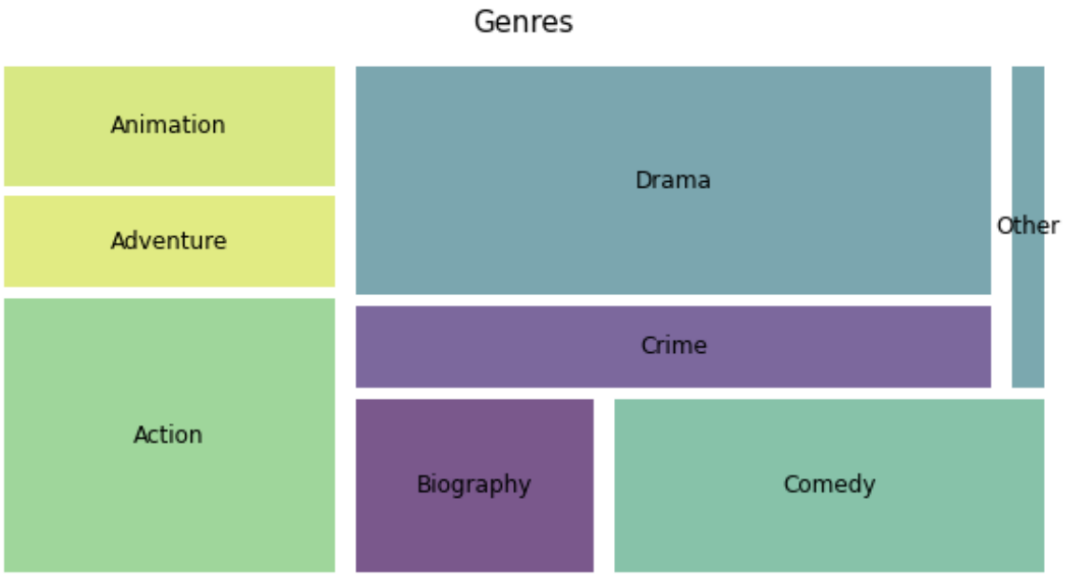

Figure 9: frequency of each genre using mosaic plot

4.2 Question 2

Now, it is time to go through data more carefully and inspect different features to find interesting results and patters, if there exists any.

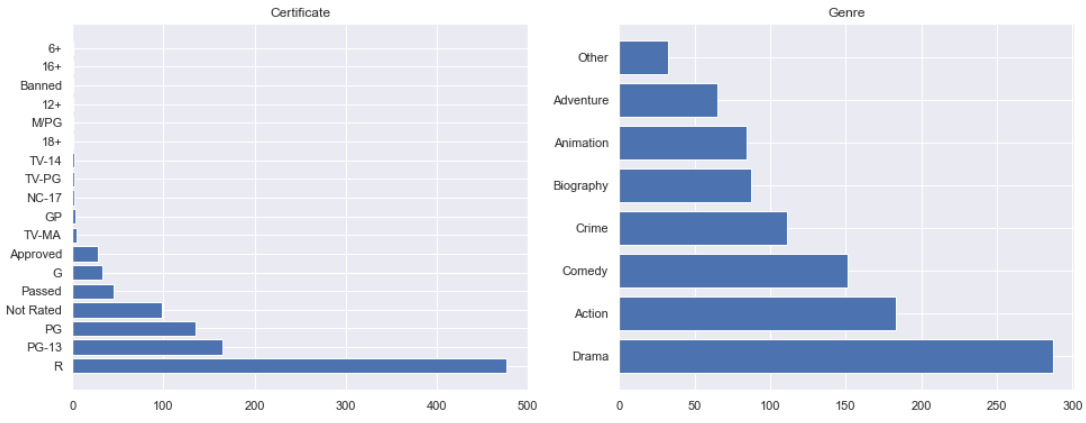

Figure 10: exploratory data analysis for categorical variable

According to the figure 10, it can be inferred that the most frequent and maybe popular genre is drama and after that the action movies are so common. The most of the movies has R certificate.

Exploratory Data Analysis for numerical variable

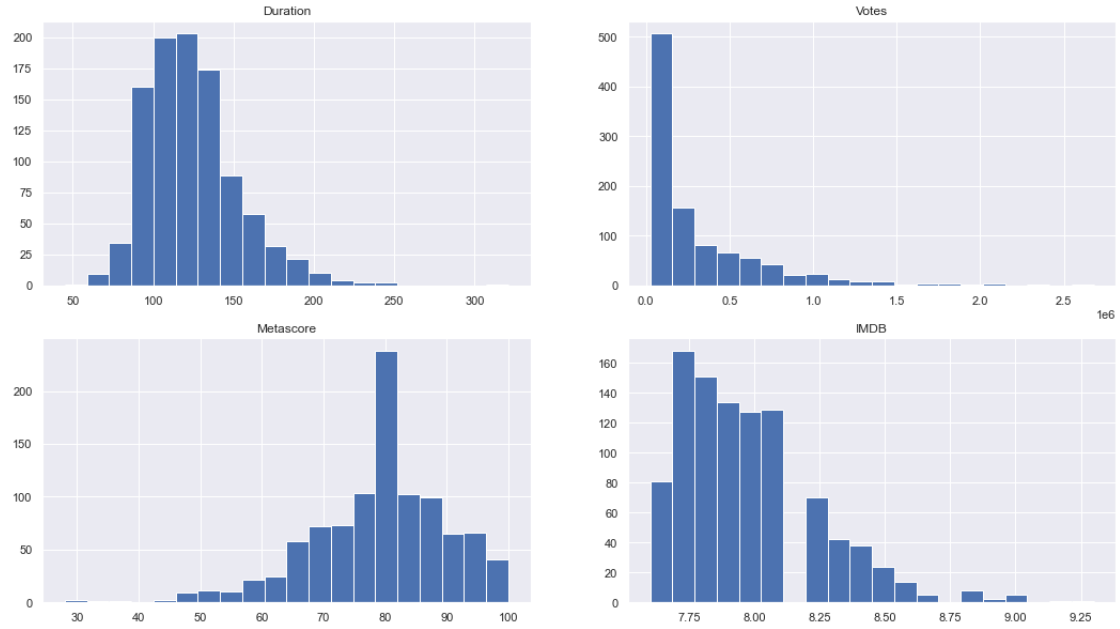

Figure 11: exploratory data analysis for numerical variable

As Duration plot shows, most of the movies have the duration between 100 and 130. This plot has a right skewness distribution.
Most of the movies has many votes, so we can trust the scores that a movie has got.
Metascore has a left skewness distribution.
IMDB has a right skewness distribution. It is so rare that rating of a movie is more than 9.

Figure 12 shows the years that have the most frequency in the dataset. As you see, the most one is the year 2014.

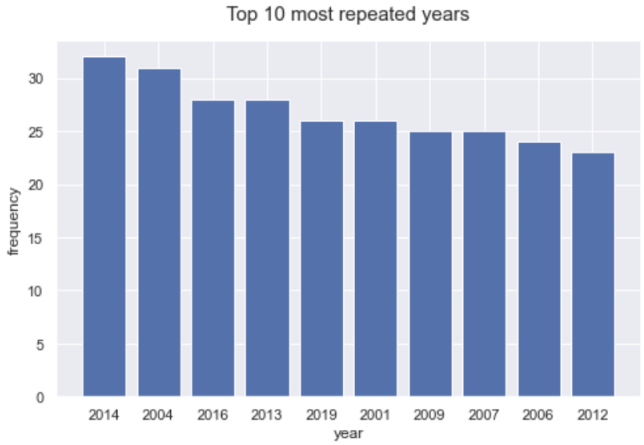

Figure 12: Top 10 most repeated years

Figure 13 shows the top 10 most frequent director in the dataset. As you see, the most frequent movies belongs to Alfred Hitchcock with 13 movies.

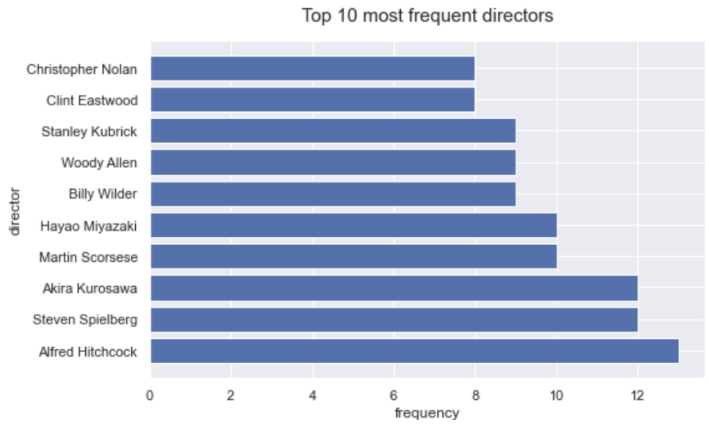

Figure 13: Top 10 most frequent director

Here, to get top 10 most frequent stars, all 3 columns that are related to stars, were considered. As the plot shows, the most repeated star is Robert De Niro.

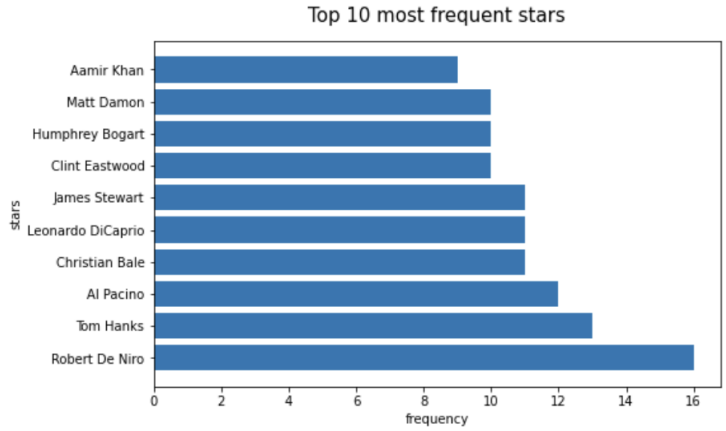

Figure 14: Top 10 most frequent stars

Box Plot

Here is another EDA for numerical variables in the dataset using box plots:

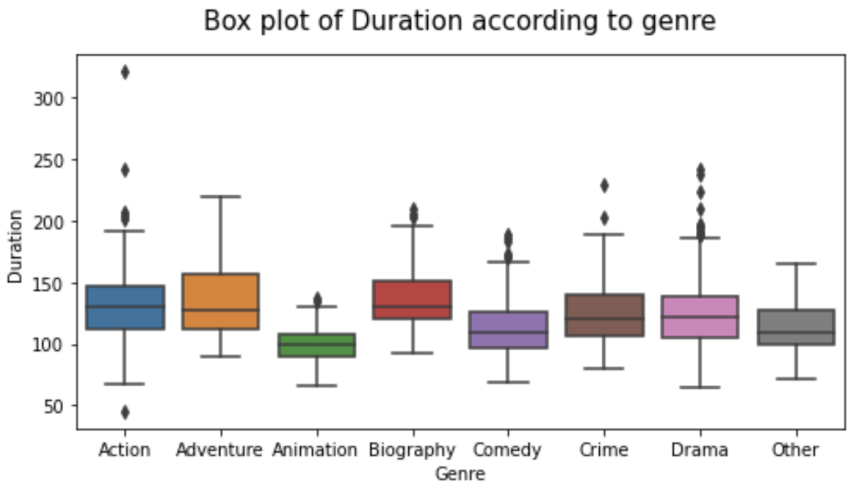

Figure 15: Box plot of Duration according to genre

All of the boxes in plot above, are almost symmetric except adventure, biography, and other that they are right skewed.

For action genre, we see some outliers that they cause this genre has the highest variance of duration among other genres. After that, drama has the highest variance of duration.

Different genres have different median. Animation genre has the lowest and biography has the highest median among others. Adventure genres has the highest lower and upper whisker and it has no outlier.

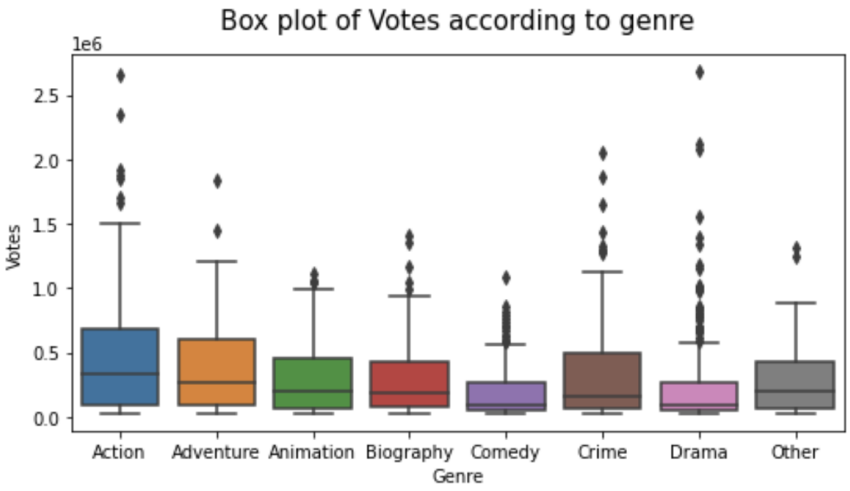

Figure 16: Box plot of Votes according to genre

Comedy, drama, and crime genres has the explicit right skewed distribution. It seems that drama has the most outliers. It can be said many people vote drama movies. Animation and comedy especially comedy has the lowest variance. Action has the highest IRQ, median, and upper whisker. So it can be inferred that this genres attracts a lot of viewers and in comparison to drama it has lower outliers.

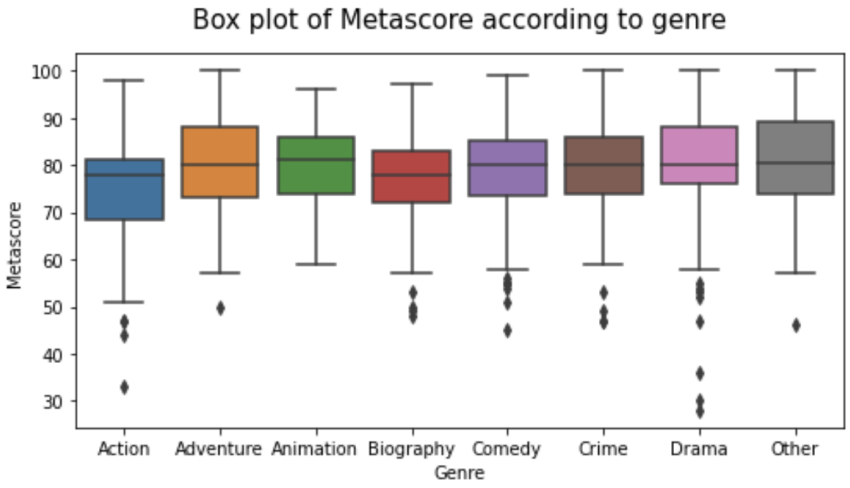

Figure 17: Box plot of Metascore according to genre

The boxes above are almost symmetric except action and drama that they are respectively left and right skewed. Drama has the most outliers among others. Because this genre has many and different viewers and fans so their scores may vary and this causes

the high variance. It seems that animation has the highest median. Action and biography has the lowest median. Action movies have also the lowest $Q1$, $Q3$ and lower whisker among others.

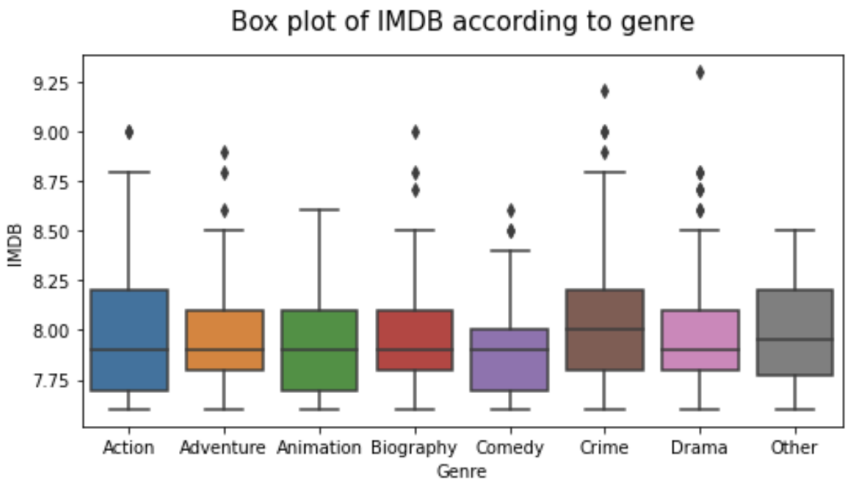

Figure 18: Box plot of IMDB according to genre

Biography, adventure, drama, and action movies have right skewed distribution. Comedy has a left skewed distribution. Drama has an outlier with the highest value among others.
Crime has the highest median. Action movies have the highest upper whisker. It can be concluded that this genre is highly rated and also according to size of IRQ, it can be guessed that it has a high variance.

The correlation between numerical features in the dataset is as follows:

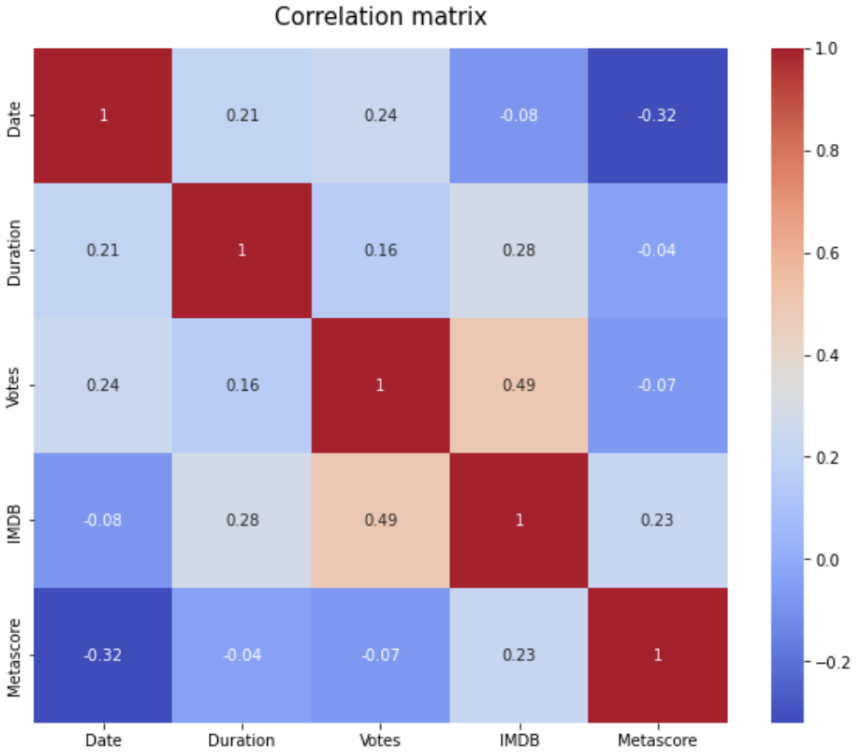

Figure 19: Correlation matrix

As you see, the most correlation is between votes and IMDB rating. It can be logical, because more votes effects on rating of a movie.

In figure 20, the distribution of each numerical feature according to genre and also the correlation of these features with each other can be seen. This plot shows how genre is distribute among numerical values. The distributions show that Drama and Action are the two genres that can be separated from other genres even with two simple features. Other classes however overlap very much and it is not easy to distinguish them using only two numerical values.

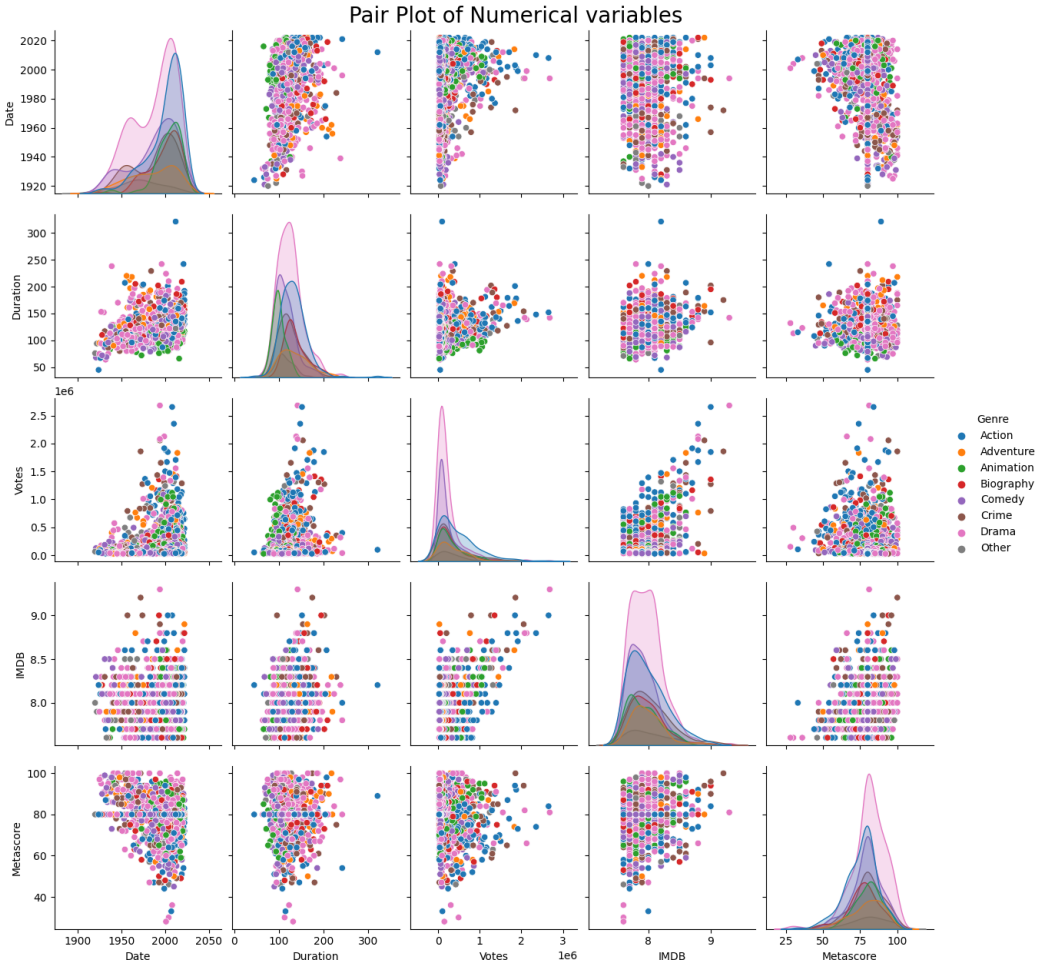

Figure 20: pair plot of numerical features

5 Feature selection and dimension reduction

Yes, we have the curse of dimensionality since there are about 5000 features for 1000 movies.

5.1 PCA

PCA is a common way to visualize a dataset in 2 dimensions. To apply PCA in the dataset, it is necessary that first convert categorical features to numerical value. Then, standardize the range of variables. It is done using StandardScaler() that standardize features by removing the mean and scaling to unit variance. After that, PCA was applied to the transformed data with 2 components. Next plot these data using plotly library that is as follows:

Figure 21: PCA of the dataset

The color of dots corresponds to the genres. As you see, PCA couldn't separate the different genres. It is because of this fact that PCA is an unsupervised method to dimension reduction. It searches for a dimension that has the most variance of the data.

5.2 LDA

Linear Discriminant Analysis is a dimensionality reduction technique that is commonly used for supervised classification problems. It is used for modelling differences in groups i.e. separating two or more classes. It is used to project the features in higher dimension space into a lower dimension space. [1]
The pictures below is the result of LDA in the dataset. Number of its components is 7. For scatter plot, it is showed first 2 components.
In the right picture, you see the kernel density estimation plot of the output of LDA transformation. It can be guessed the genre animation has the lowest variance.

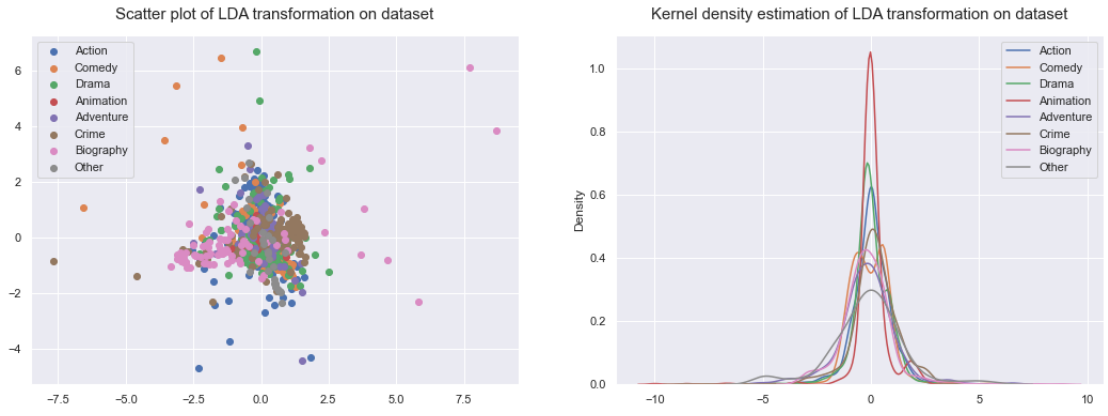

Figure 22: LDA of the dataset

6 Classification Methods

6.1 Question 1

In this part, we used classification models with the scikit-learn package to predict the genre of movies based on other features. The models used are logistic regression, SVM, decision tree, KNN, MLP, and Gaussian naive Bayes. These models are tested on three feature sets. These sets are all features, PCA features, and selected features from the random forest results in Section 5. Also, we used accuracy, precision, recall and F1 score as criteria to evaluate the classifiers. These criteria are measured based on the Kfold method with $k = 10$. We use this metric to evaluate models based on their prediction. Also with these metrics, we can evaluate models based on results for each class. For logistic regression and SVM we used one vs one and one vs rest methods for classification but for other models only used one vs one method. Results for each model are shown below.

6.1.1 Logistic Regression

Using all feature set with one vs rest has the best result based on accuracy. accuracy for this model was 38.1%.

```
Accuracy :0.381
-------------------------------------
Precision :0.3267479904233034
-------------------------------------
Recall :0.21573108739342603
-------------------------------------
F1 score :0.20407616859608776
-------------------------------------
```

Figure 23: Logistic Regression(ovr) on all features results

Model on selected features with one vs rest method for classification has the best precision result. The precision for this model is 33.65%.

```
Accuracy :0.379
-------------------------------------
Precision :0.3365377690238994
-------------------------------------
Recall :0.23665065753026915
-------------------------------------
F1 score :0.24269582922648852
-------------------------------------
```

Figure 24: Logistic regression(ovr) on selected features results

The best result for recall is measured on the model with one vs one method on the PCA feature set. Recall for this model is 29.22%.

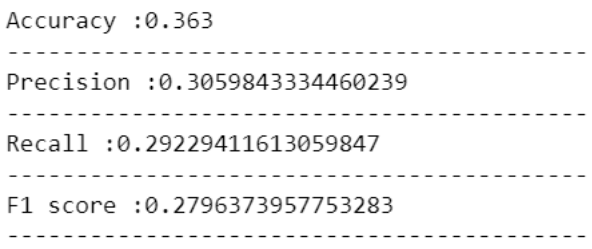

Figure 25: Logistic regression(ovo) on PCA features results

Based on the f1-score, model on PCA feature set with one vs one method has best result. f1-score for this model is 27.96%.

6.1.2 SVM

In this model, we used the RBF kernel which is the default kernel for SVM in the scikit-learn package. Using the model on the PCA feature set with one vs one method has the best result for accuracy. accuracy for this model is 42.69%. Also, this model has the best results for precision, recall, and f1-score. precision and recall for this model are 37.99% and 33.90%.

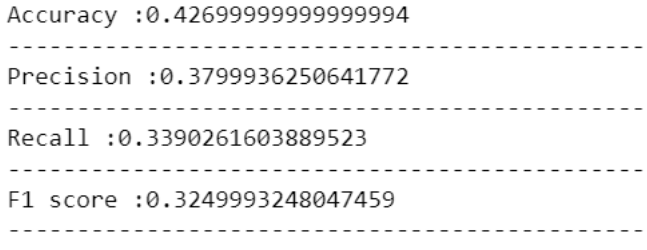

Figure 26: SVM (ovo) on PCA features results

6.1.3 Decision Tree

In comparing the model in three sets of features, the model on all features set has the best results for accuracy with 34.8% accuracy. But the model on the PCA features set has the best result for other metrics. Precision, recall, and f1-score for this model are 24.12%, 22.02% and 21.88%.

```
Accuracy :0.348
-----------------------------------------------
Precision :0.19862975672713737
-----------------------------------------------
Recall :0.2041224136657444
-----------------------------------------------
F1 score :0.15942947281724174
-----------------------------------------------
```

Figure 27: Decision tree on all features results

```
Accuracy :0.32799999999999996
-------------------------------------------------
Precision :0.24129815603157923
-------------------------------------------------
Recall :0.23022163978355
-------------------------------------------------
F1 score :0.21884411989293948
-------------------------------------------------
```

Figure 28: Decision tree on PCA results

6.1.4 KNN

We used this model with the parameter k=31. The model with this parameter had better performance. The best results for this model are on the PCA features set. results are shown in figure 29.

```
Accuracy :0.33199999999999996
-------------------------------------------------------
Precision :0.3139805560730231
-------------------------------------------------------
Recall :0.2265628898228207
-------------------------------------------------------
F1 score :0.20883251617006587
-------------------------------------------------------
```

Figure 29: KNN on PCA features results

6.1.5 MLP

For this model, we set many parameters. The activation function for this model is Logistic. The hidden layer size is (10, 15) and the solver is set to Adam. Also, the max iteration is 100, and the learning rate is 0.001. This model has the best accuracy on all features set and selected features set with a 30% result.

```
Accuracy :0.3
--------------------------------------------------
Precision :0.10738117297185423
--------------------------------------------------
Recall :0.13351209554334553
--------------------------------------------------
F1 score :0.07066882769519625
--------------------------------------------------
```

Figure 30: MLP on all features results

```
Accuracy :0.3
--------------------------------------------------
Precision :0.1468545898021242
--------------------------------------------------
Recall :0.13517114982648068
--------------------------------------------------
F1 score :0.07879591003433346
--------------------------------------------------
```

Figure 31: MLP on selected features results

Also, based on the precision, model on selected features set has best result with 14.68% precision. But model on PCA features set has the best recall and f1-score. These metrics are 13.59% and 14.04%.

```
Accuracy :0.297
--------------------------------------------------
Precision :0.1359689202333385
--------------------------------------------------
Recall :0.140469393872536
--------------------------------------------------
F1 score :0.09053395328789726
--------------------------------------------------
```

Figure 32: MLP on PCA features results

6.1.6 Gaussian Naïve Bayes

This model has the best results on selected features set. The results are shown below in figure 33.

```
Accuracy :0.30300000000000005
--------------------------------------------------
Precision :0.34516306299941135
--------------------------------------------------
Recall :0.2759938569653623
--------------------------------------------------
F1 score :0.26133989263868096
--------------------------------------------------
```

Figure 33: Gaussian Naive Bayes on selected features results

From the above results, we can compare models based on metrics and choose the best. In the below tables, the best results for each model have been shown:

Model	Best result
Logistic regression	38.1% - all features
SVM	42.69% - PCA features
Decision tree	34.8% - all features
KNN	33.19% - PCA features
MLP	30% - all features or selected features
Gaussian naïve Bayes	30.3% - selected features

Table 1: Comparison of models based on accuracy.

Model	Best result
Logistic regression	33.65% - selected features
SVM	37.99% - PCA features
Decision tree	24.12% - PCA features
KNN	31.39% - PCA features
MLP	14.68% - selected features
Gaussian naïve Bayes	34.51% - selected features

Table 2: Comparison of models based on precision.

Model	Best result
Logistic regression	29.22% - PCA features
SVM	33.90% - PCA features
Decision tree	23.02% - PCA features
KNN	22.65% - PCA features
MLP	14.04% - PCA features
Gaussian naïve Bayes	27.59% - selected features

Table 3: Comparison of models based on recall.

Model	Best result
Logistic regression	27.96% - PCA features
SVM	32.49% - PCA features
Decision tree	21.88% - PCA features
KNN	20.88% - PCA features
MLP	09.05% - PCA features
Gaussian naïve Bayes	26.19% - selected features

Table 4: Comparison of models based on F1-score.

Metric	Model	result
Accuracy	SVM	42.69%
Precision	SVM	37.99%
Recall	SVM	33.90%
F1-score	SVM	32.49%

Table 5: The best model for each metric.

Based on the above results, the SVM model on PCA features set with the one-vs-rest method has the best results for all metrics therefore, this model is the best model for prediction genres of IMDB movies.

6.2 Question 2

Above we see that the best result for each model (based on the criteria) is on the PCA features or selected features. Therefore, the use of all features cannot have better results. It seems that other features cannot have additional data to predict.

6.3 Question 3

Based on the above results and the graph below, SVM has the best accuracy. But it cannot be said that this is the best model. Because we have to consider other metrics and then choose the best model. Maybe a model has good accuracy but not good other metrics. Maybe a model fits one class and has good accuracy based on the large size of that class, but it cannot predict other classes well.

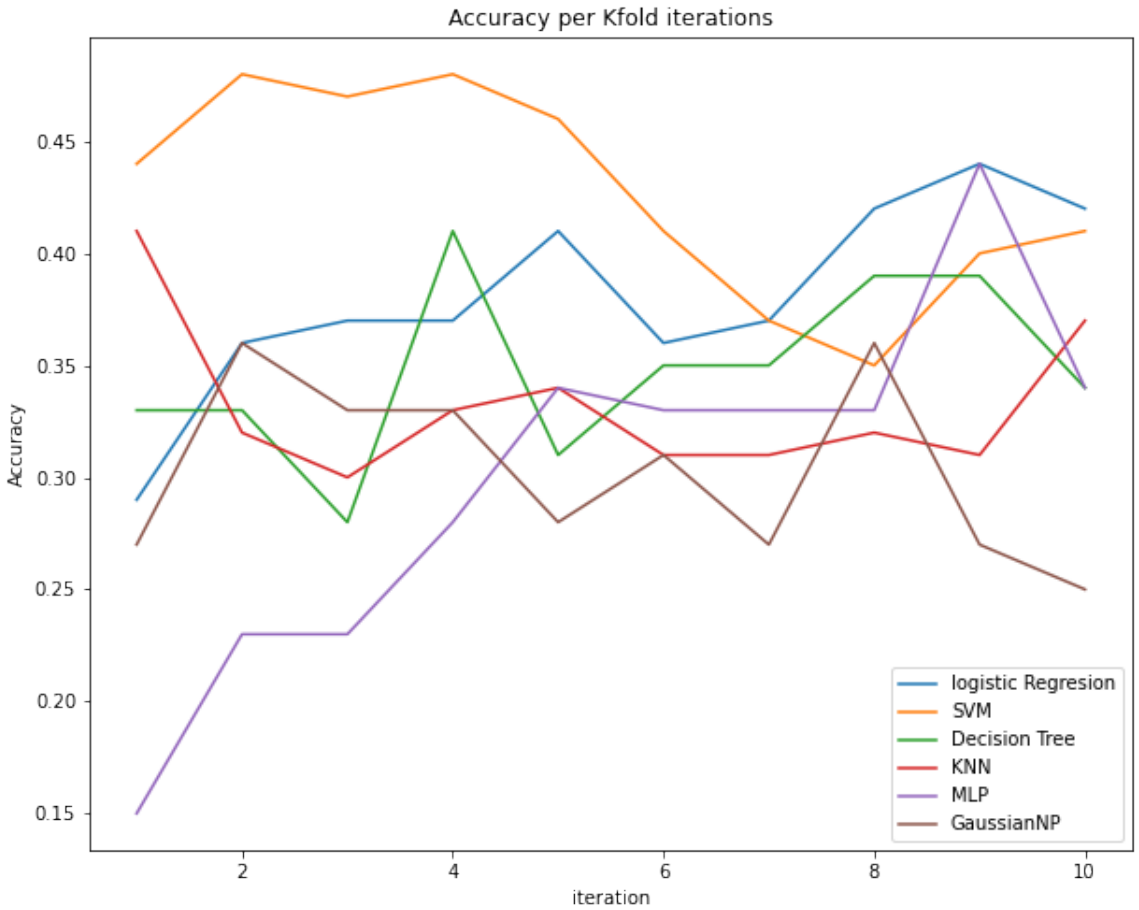

Figure 34: Models accuracy

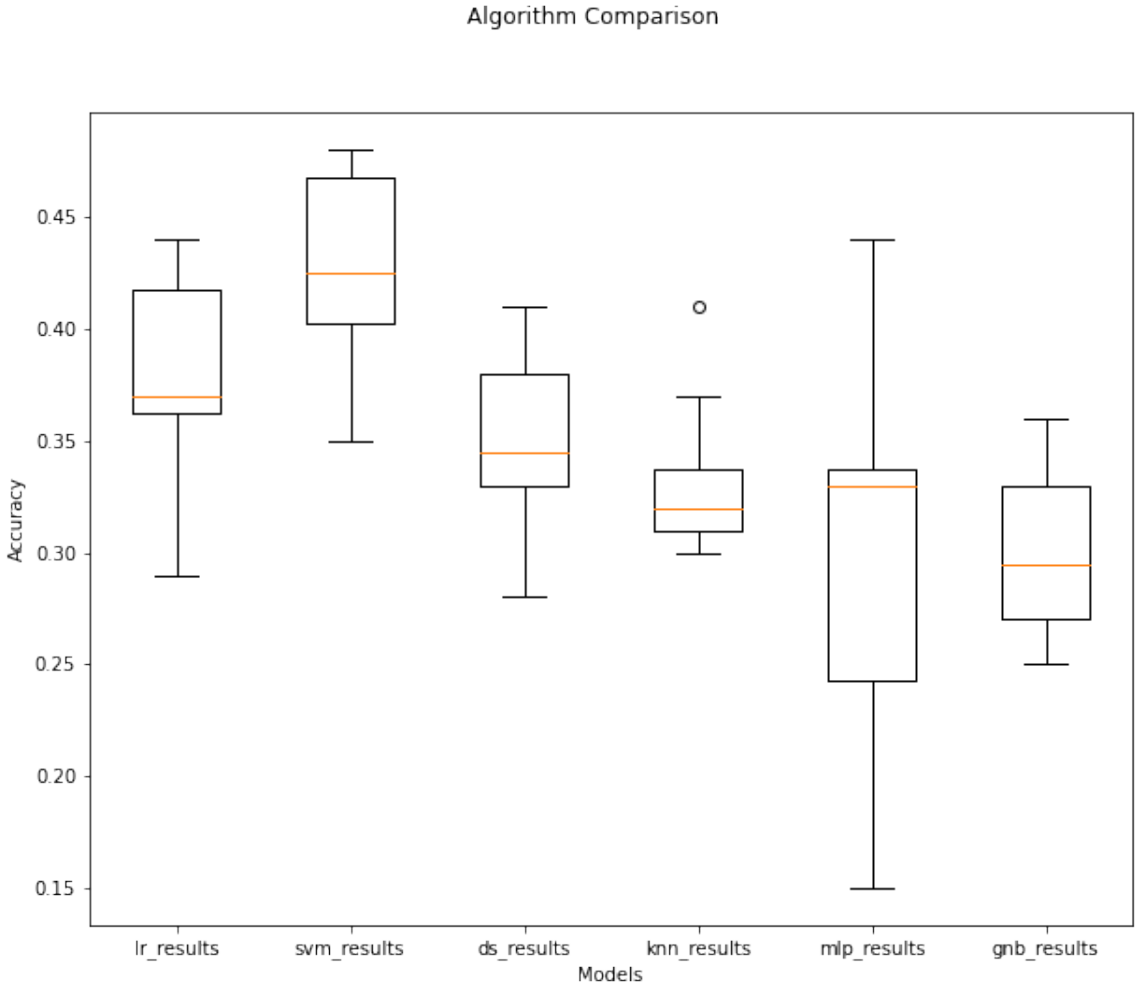

Figure 35: Accuracy distribution of models

6.4 Question 4 - Extra Part

Another methods for classification **Another methods for classification**

6.4.1 Ensemble Methods – Voting Classifier

In majority voting, the predicted class label for a particular sample is the class label that represents the majority (mode) of the class labels predicted by each individual classifier.
E.g., if the prediction for a given sample is

- classifier 1 → class 1
- classifier 2 → class 1
- classifier 3 → class 2

the VotingClassifier (with voting='hard') would classify the sample as "class 1" based on the majority class label [6]ref2.
In this case, LogisticRegression, XGBClassifier, RandomForestClassifier, and DecisionTreeClassifier are used as classifiers.

```
Accuracy :0.371
-----------------------------------
Precision :0.23811891827396042
-----------------------------------
Recall :0.21617401749836346
-----------------------------------
F1 score :0.1906089062660763
-----------------------------------
```

Figure 36: Voting Classifier - results

6.4.2 Random Forest Classifier with 30 estimators

A random forest is a meta estimator that fits a number of decision tree classifiers on various sub-samples of the dataset and uses averaging to improve the predictive accuracy and control over-fitting [3].

```
Accuracy :0.39
----------------------------------
Precision :0.2734948547332607
----------------------------------
Recall :0.24211594776468393
----------------------------------
F1 score :0.20876660529587543
----------------------------------
```

Figure 37: Random Forest Classifier - results

6.4.3 Extra Trees Classifier with 30 estimators

This class implements a meta estimator that fits a number of randomized decision trees (a.k.a. extra-trees) on various sub-samples of the dataset and uses averaging to improve the predictive accuracy and control over-fitting [4].

```
Accuracy :0.385
----------------------------------
Precision :0.27934733512653687
----------------------------------
Recall :0.24224579225861492
----------------------------------
F1 score :0.20468951912187924
----------------------------------
```

Figure 38: Extra Trees Classifier - results

6.4.4 Ada Boost Classifier with 100 estimators

The core principle of AdaBoost is to fit a sequence of weak learners (i.e., models that are only slightly better than random guessing, such as small decision trees) on repeatedly modified versions of the data. The predictions from all of them are then combined through a weighted majority vote (or sum) to produce the final prediction [5].

```
Accuracy :0.29300000000000004
----------------------------------
Precision :0.19420398166284303
----------------------------------
Recall :0.2095582648289625
----------------------------------
F1 score :0.17191564219728256
----------------------------------
```

Figure 39: Ada Boost Classifier - results

6.4.5 Hierarchical Classification

HiClass is an open-source Python library for local hierarchical classification entirely compatible with scikit-learn. It contains implementations of the most common design patterns for hierarchical machine learning models found in the literature, that is, the local classifiers per node, per parent node and per level. Additionally, the package contains implementations of hierarchical metrics, which are more appropriate for evaluating classification performance on hierarchical data [6].

```
Accuracy :0.417
----------------------------------
Precision :0.34753746923143264
----------------------------------
Recall :0.31957297832594017
----------------------------------
F1 score :0.3075901446748757
----------------------------------
```

Figure 40: Hierarchical Classifier - results

Choosing different feature subsets or performing feature reduction can not predict genre well. Although predictions are much better than random, models can still be improved to obtain higher accuracies. Genre is a complicated thing to predict and depends on the context of a movie. One of the good features that can be used to get a slight overview of movie's content is movie summary. As mentioned in the very first part, summary is one of the features available on the website and numerical features can be extracted from summary using a light NLP model. The model must be able to take in text or the tokenized form of it and output numerical features based on the whole content and not just the words. That is why using techniques such as TF-IDF and Bag of Words would not give a good output. The chosen network is BERT.

BERT is a language model based heavily on the Transformer encoder and was trained on the masked language model and next sentence prediction tasks. It takes as input the embedding tokens of one or more sentences. The first token is always a special token called [CLS] and sentences are separated by another special token called [SEP]. For each token, BERT outputs an embedding called hidden state. For classification problems, the major interest is in the hidden state associated with the initial token [CLS], which somehow captures the semantics of the entire sentence better than the others. Therefore, this embedding can be used as the input of a classifier that is built on top of it.

A lighter version of BERT, called DistilBERT is used which is 40% smaller than the original but still maintains about 97% performance on various NLP tasks. BERT's inputs are not the original words but the tokens. Simply BERT has associated a tokenizer that preprocess the text so that it is appealing for the model. Words are often split into sub words and in addition, special tokens are added: [CLS] to indicate the beginning of the sentence, [SEP] to separate multiple sentences, and [PAD] to make each sentence have the same number of tokens. [7] [8]

7 Model improvement

To improve the model, some features were extracted from the summary and other text-based features that were informative for genre prediction, such as movie titles.
As a result, only summary will be useful for the continuation of the analysis.

By applying BERT for feature extraction from the summary, about 768 features are added to the feature space. In order to use these features, you can either use all 768 features extracted from the BERT model or use a CLF feature that represents the summary.
The first method was more accurate, so all features were used in the final model.

As shown in the last part, the Logistic Regression method generally has better performance and a higher F1 score; this method also performs better here.

Here are some results

Logistic Regression without PCA

```
Accuracy :0.5176767676767676
----------------------------------
Precision :0.46676095441436366
----------------------------------
Recall :0.3948888358646194
----------------------------------
F1 score :0.39629213845948175
----------------------------------
```

Figure 41: Logistic Regression without PCA - results

I have now reduced the dimensions to 40 by applying PCA to the features.

Logistic Regression

```
Accuracy :0.4823232323232324
----------------------------------
Precision :0.46249250082884963
----------------------------------
Recall :0.4160643049853241
----------------------------------
F1 score :0.415948563394372
----------------------------------
```

Figure 42: Logistic Regression - results

Due to our imbalanced genres, the F1 score is a more useful metric. Since F1 is fixed, it's better to use fewer features in the training model to prevent the curse of dimensionality.

let's see other methods' results

SVM

```
Accuracy :0.47732323232323237
----------------------------------
Precision :0.4205847778081435
----------------------------------
Recall :0.3534868713704943
----------------------------------
F1 score :0.34916159625380855
----------------------------------
```

Figure 43: SVM after PCA - results

MLP

```
Accuracy :0.43304040404040406
----------------------------------
Precision :0.3479593356542357
----------------------------------
Recall :0.2731305062230963
----------------------------------
F1 score :0.251182092544298
----------------------------------
```

Figure 44: MLP after PCA - results

Gaussian Naive Bayes

```
Accuracy :0.4358989898989899
----------------------------------
Precision :0.38580359491810706
----------------------------------
Recall :0.3731504572643415
----------------------------------
F1 score :0.3600700764075303
----------------------------------
```

Figure 45: Gaussian Naive Bayes after PCA - results

Decision Tree

```
Accuracy :0.32332323232323235
----------------------------------
Precision :0.27105804193536986
----------------------------------
Recall :0.23042835087882993
----------------------------------
F1 score :0.22907655237200167
----------------------------------
```

Figure 46: Decision Tree after PCA - results

Just use CLF as a feature of summary.
Logistic Regression

```
Accuracy :0.39586868686868687
----------------------------------
Precision :0.36425684628920224
----------------------------------
Recall :0.29897208593587965
----------------------------------
F1 score :0.3089964925131823
----------------------------------
```

Figure 47: Logistic Regression - results

An overview of the results:

Model	PCA	F1	Accuracy	Precision	Recall
logistic regression	No	0.39	0.51	0.46	0.39
logistic regression	Yes	0.415	0.48	0.46	0.41
SVM	Yes	0.34	0.47	0.42	0.35
MLP	Yes	0.25	0.43	0.34	0.27
Gaussian Naive Bayes	Yes	0.36	0.43	0.38	0.37
Decision Tree	Yes	0.22	0.32	0.27	0.23

Conclusion

About 10% of F1 scores were increased by using summary as a feature to predict genre.

8 References

Bibliography

[1] ”geeks for geeks,” [Online]. Available: www.geeksforgeeks.org.

[2] ”scikit-learn,” [Online]. Available: www.scikit-learn.org.

[3] ”scikit-learn,” [Online]. Available: www.scikit-learn.org.

[4] ”scikit-learn,” [Online]. Available: www.scikit-learn.org.

[5] ”scikit-learn,” [Online]. Available: www.scikit-learn.org.

[6] ”arxiv,” [Online]. Available: arxiv.org.

[7] ”arxiv,” [Online]. Available: arxiv.org.

[8] ”towardsdatascience,” [Online]. Available: towardsdatascience.com.