



Movie Popularity Prediction

Milestone 1 Report

Team ID	CS_6
---------	------

Name	ID
محمد محروس محمد احمد عبدالرحمن	20201700731
صالح عادل صالح محمد	20201700415
شهاب مصطفى فهمي علي	20201700405
عبدالرحمن حسني محمد كامل	20201700434
علي محمد علي شارب	20201701108
عمر ايمن حسن غباشي	20201701112

The Report

1) Preprocessing Techniques

1- We dropped Columns based on Certain reasons for Certain Columns

These columns were (homepage, id, status).

How: We got the number of nulls from the homepage column and found it more than half of the data.

We got the most frequent value from the id column and found that there is no repeated value.

We got the most frequent value from the status and found that its number is about all the data.

2- Drop any row that any value on it is null.

3- Split the data by 70 for the train and 30 for the test.

4- (budget):

1- we apply feature scaling on it.

2- We replace the zeros with the column's median. 5- (genres):

1-we got a list containing ids of dictionary and implemented algorithm on it.

2-an algorithm is going to relate the genres' rows with each corresponding runtime value using the ID of each genre to get a numeric value instead of each row of the genres.

6-(keywords):

1- we extracted a list of values of the names from the column.

2- feature encoding.

3- feature scaling.

4- replacing zeros with median.

6- (original language): we applied feature encoding.

6- (Original title):

1- using RE, we removed the stop words, applied the stemming on it.

2- feature encoding.

3- Feature scaling.

7- (overview):

1- using RE, we removed the stop words, applied the stemming on it.

2- feature encoding.

3- Feature scaling. 8- (viewer count):

1- Feature scaling.

2- replaced the outliers with the median.

9- (production companies):

1- we got a list containing ids of dictionary and implemented algorithm on it.

2- an algorithm is going to relate the product company' rows with each corresponding runtime value using the ID of each product company to get a numeric value instead of each row of the product company.

10- (production countries):

1- we got list containing iso_3166_1's of dictionary and implement algorithm on it.

2- an algorithm is going to relate the production countries' rows with each corresponding runtime value using the iso_3166_1 of each production countries to get a numeric value instead of each row of the production countries.

11- (release date): algorithm to calculate the difference (in years) between the current local date and the release date.

12- (revenue):

1- We apply feature scaling on it.

2- We replace the zeros with the column's median.

13- (runtime)

We apply feature scaling on it and use it on dictionaries columns.

14- (spoking language)

1- we extracted a list of values of the iso_639_1 from the column.

2- feature encoding.

3- feature scaling.

4- replacing zeros with median.

15- (tagline)

- 1- using RE, we removed the stop words, applied the stemming on it.
 - 2- feature encoding.
 - 3- Feature scaling.
- 16- (title)

- 1- using RE, we removed the stop words, applied the stemming on it.
- 2- feature encoding.
- 3- Feature scaling.

17- (vote count)

- 1- Feature scaling.
- 2- replaced the outliers with the median.

18- The Preprocessing is done on the training data and the test data but the different between them that the train data is doing fit and transform in feature encoder and feature scaling while the test data is doing transform only.

2) Feature Selection (Perform Analysis)

- We apply Correlation on dataset after the preprocessing techniques.

- 1- Apply concatenation on training data and testing data.
- 1- Apply correlation that the correlation of target column > 0.1.
- 2- We got on the top features from the correlation to train data and test data.

3) Regression techniques

1- Apply Linear regression.

- a. Fit train data and target train data.
- b. Predict test data.
- c. Calculate the mean square error on this data.
- d. $MSE = 0.5127449030301584$.
- e. Calculate the $r^2_score = 33.49379138051339$.

2- Apply Polynomial regression.

- a. Apply polynomial regression.
- b. Fit and transform on the train data.
- c. Fit the transform features to the linear regression.
- d. Fit the polynomial data and target train data.
- e. Predict on test data.
- f. Calculate the mean square error = 0.4688330884532879 .
- g. Calculate the $r^2_score = 39.18942732706475$.

3- Apply Ridge regression.

- a. Fit train data and target train data.
- b. Predict test data.
- c. Calculate the mean square error on this data.
- d. $MSE = 0.513919416135518$.
- e. Calculate the $r^2_score = 33.3414497128542$.

4) Differences Between Each Model

- 1- When we applied polynomial regression the mean square error and the accuracy became better from linear regression.
 - 2- When we applied ridge regression the mean square error and the accuracy became almost like linear regression.
- 5) Features on our regressions
-

- 1- (Use) => Based on correlation of the features used, their correlation is greater than 0.1 (genres, viewercount, production companies , revenue, runtime , vote count).
- 2- (Discard) (budget, keywords , original language , original title , overview , production countries , release date , spoken language , tagline , title).

6) Sizes of training data and testing data

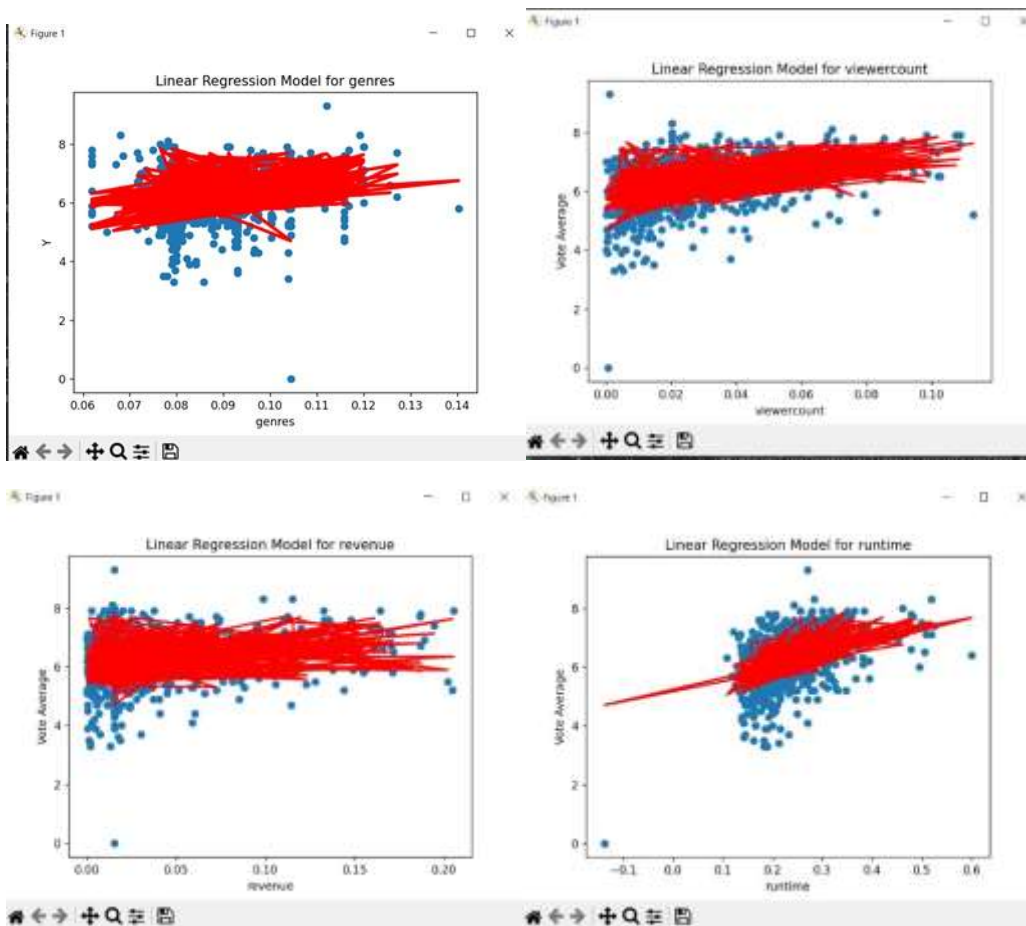
- 1- The data is divided into 70% train and 30% test and the shuffle is true when the divide.
 - 2- The number of columns for the train and test data is 16 columns before correlation.
 - 3- The number of rows for the train data is 1859 rows.
 - 4- The number of rows for the test data is 797 rows.
- 7) Techniques improve the results.
-

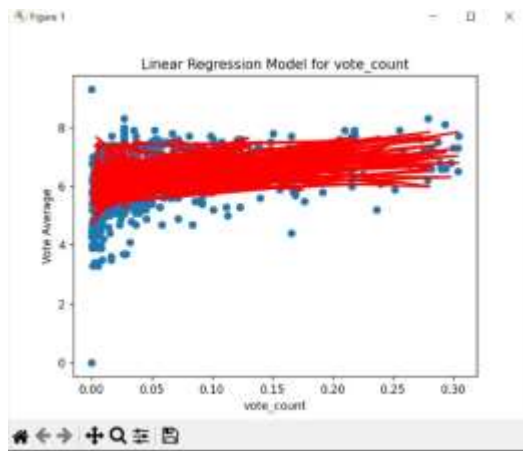
We use polynomial regression and it improves the result and improve the mean square error from 0.5127449030301584 to 0.4688330884532879 and improve the r^2_score from 33.49379138051339 to 39.18942732706475.

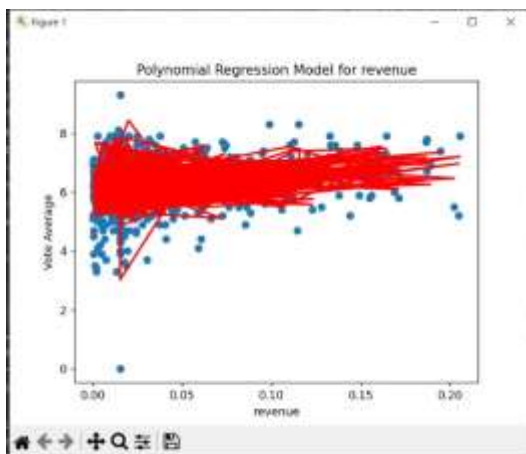
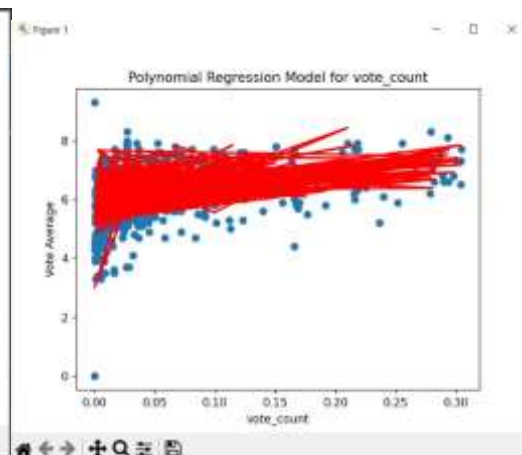
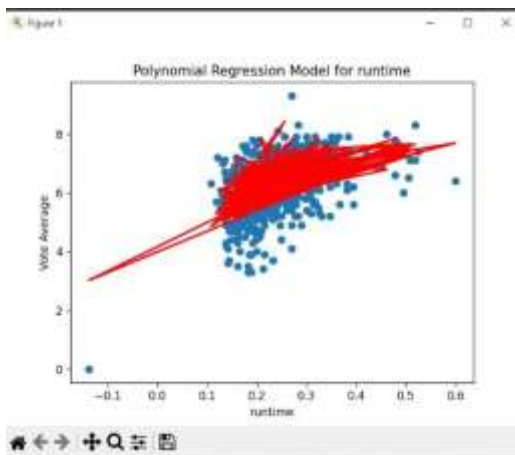
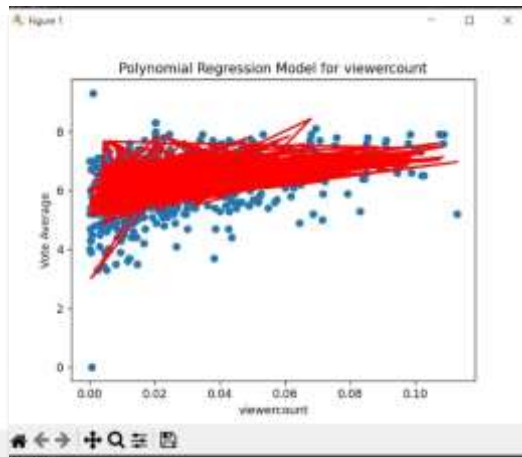
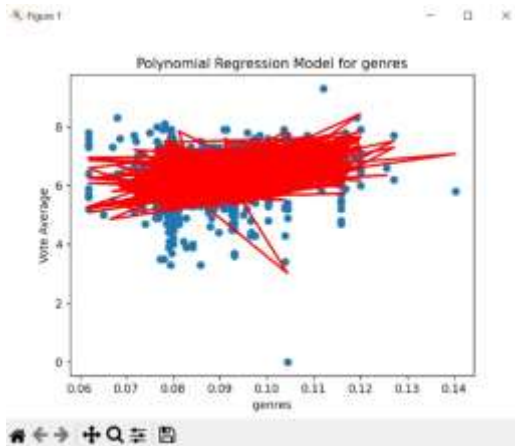
8) Screenshots

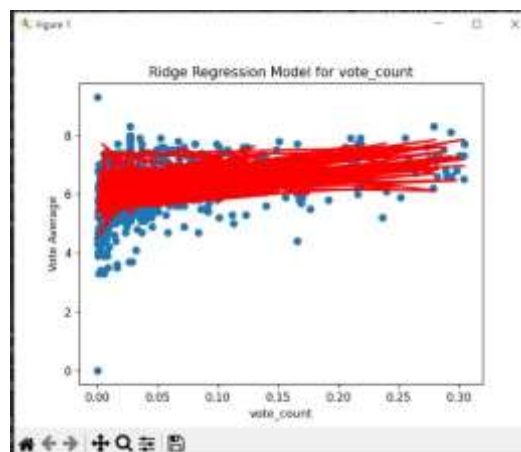
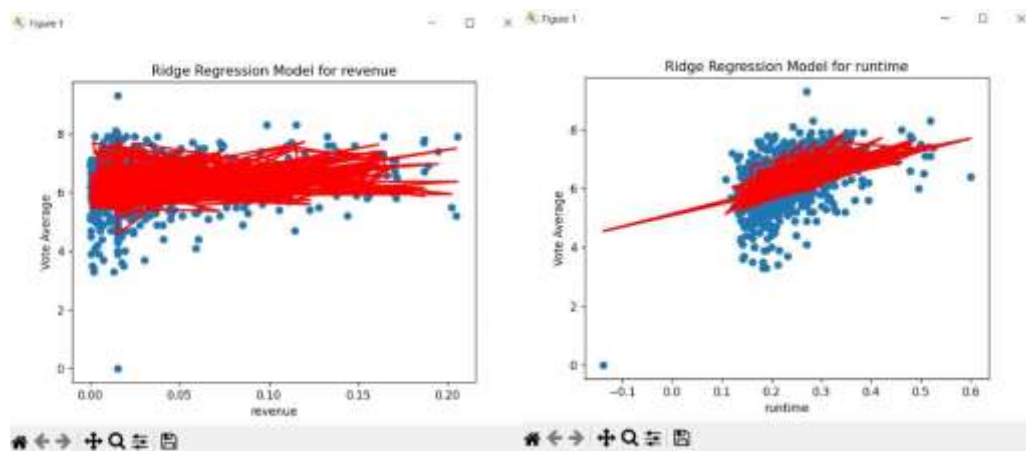
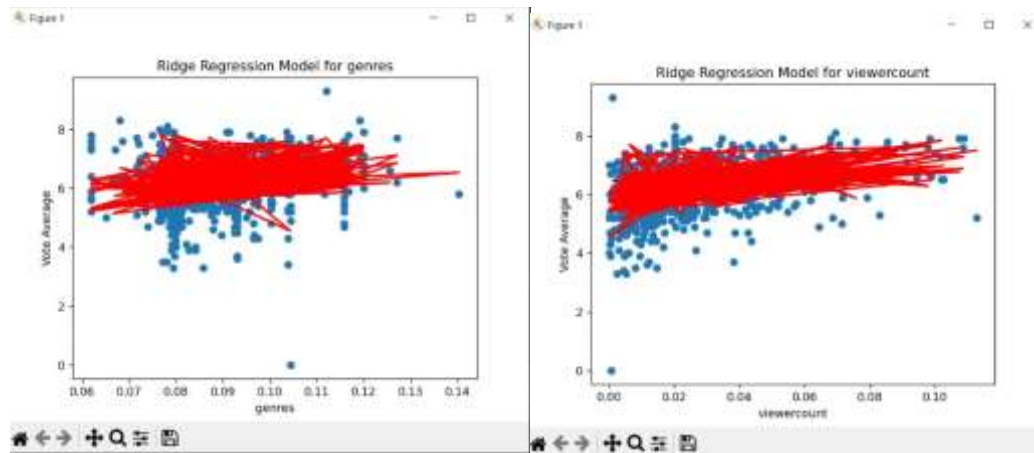
The Resultants of regression

```
Mean Squared Error for Linear Regression Model: 0.047247086011598
R2_score for linear regression model: 0.07639248517178
Mean Squared Error for polynomial regression model: 0.189819457411579
R2_score polynomial regression model: 0.0718113476188
Mean Squared Error: 0.040982181844178
R-squared Score for ridge regression: 0.04494229038088
```









The Resultants of correlation



9) Conclusion

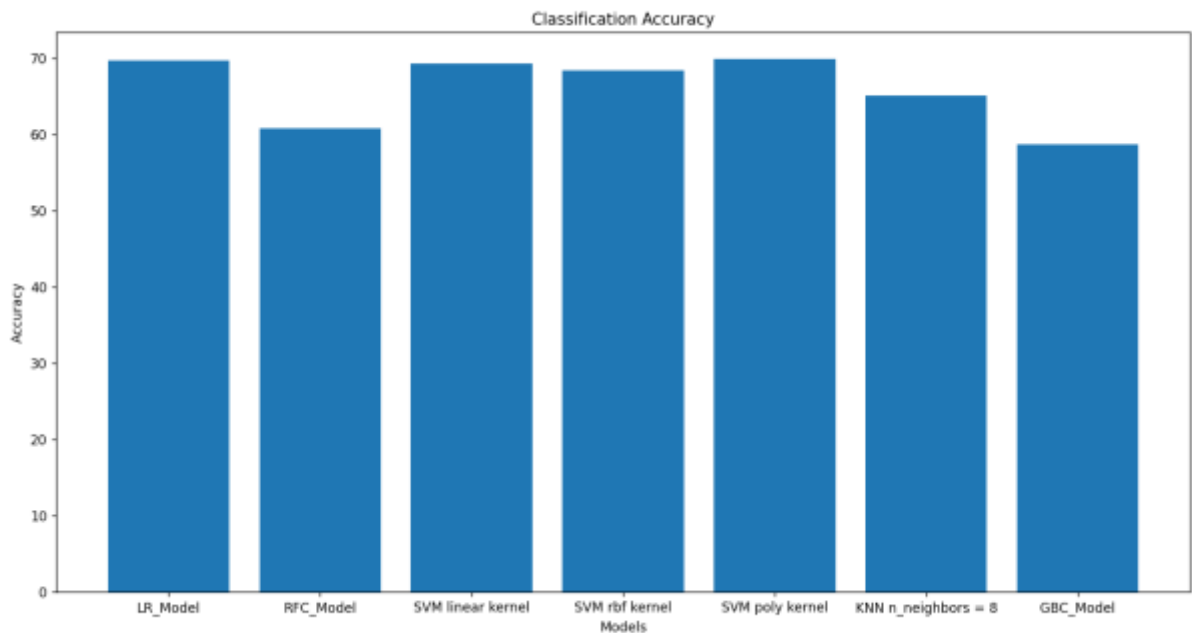
We used preprocessing techniques on this data and the data were coming bad values before the preprocessing but after the preprocessing the data became better.

We used correlation after the preprocessing techniques, and it extracted the best columns.

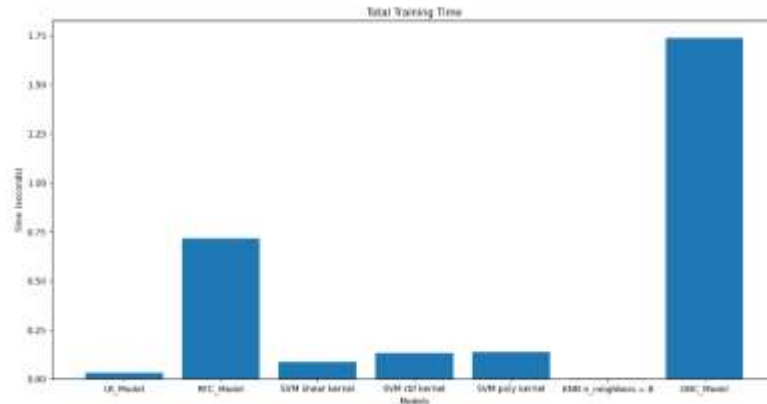
We used linear regression and polynomial regression and the mean square error for the polynomial became better than the mean square for the linear regression.

Milestone 2 Report

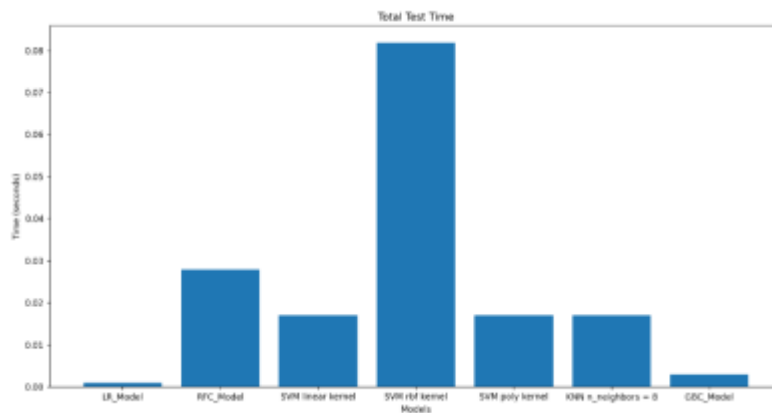
1. The classification accuracy is under range from 60% to 70 % whenever we change the type of classification or we change the hyperparameter of any classification and this is evident in his bar graph.



2. We used the time before and after each fit to calculate The total training time , We notice that it changes significantly with each classification and It changes from classifier to classifier and also changes with the change of the hyperparameters of each classifier.



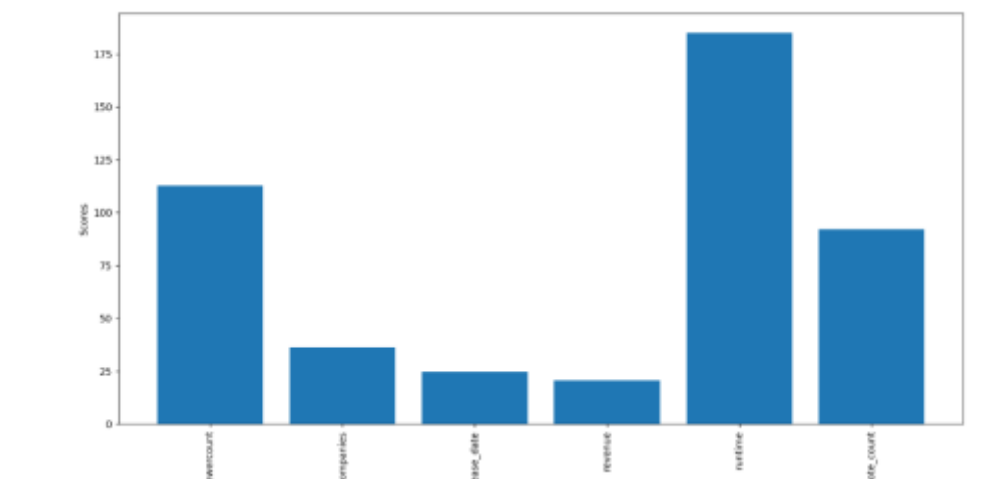
3. We used the time before and after each predict to calculate The total testing time , We notice that it changes significantly with each classification and It changes from classifier to classifier and also changes with the change of the hyperparameters of each classifier.



4. In the classification phase, the feature selection process may differ from the previous regression phase. In regression, the feature selection process involved using correlation analysis to identify relevant features. However, in classification, a different approach was adapted.

For classification, the feature selection process utilized the K-best feature selection method. This technique selects the K best features based on statistical significance and ability to contribute to the classification task. The specific criteria used to determine the best features may vary, but common approaches include chi-squared test, mutual information, or ANOVA F-value.

By adopting the K-best feature selection approach in the classification phase, the feature set was refined, focusing on the most informative and discriminative features. This process aimed to enhance the model's ability to accurately classify instances and improve overall classification performance.



5. Hyperparameter tuning plays a crucial role in optimizing the performance of machine learning models. In the case of classification models like Support Vector Machines (SVM) and K-Nearest Neighbors (KNN), tuning specific

hyperparameters such as 'C' (regularization parameter) and 'kernel' for SVM, and 'n_neighbors' for KNN can have a significant impact on the models' performance.

a. SVM:

- 1-C Hyperparameter: The C hyperparameter controls the trade-off between maximizing the margin and minimizing the classification error. Higher values of C result in a narrower margin and can lead to overfitting, while lower values encourage a wider margin but may result in underfitting.
- 2-Kernel Hyperparameter: The kernel determines the type of decision boundary created by the SVM model. Common kernel functions include linear, polynomial, and radial basis function (RBF). Choosing the appropriate kernel can greatly affect the model's ability to capture complex patterns in the data.

b. KNN:

- 1-n_neighbors Hyperparameter: The n_neighbors hyperparameter defines the number of neighbors considered for classifying a new data point. A higher value of n_neighbors can smooth out the decision boundary, reducing the model's sensitivity to noisy data but potentially oversimplifying the classification. On the other hand, a lower value of n_neighbors can lead to a

more complex decision boundary but may be prone to overfitting.

Overall, by tuning the hyperparameters of classification models such as SVM and KNN, we aim to find the optimal configuration that maximizes the models' performance on the given task. The tuning process involves selecting appropriate ranges for the hyperparameters, exploring different combinations, and evaluating the models' performance using suitable evaluation metrics.

Hyperparameter tuning can significantly improve the models' performance by finding the best trade-off between underfitting and overfitting and adapting the models to the specific characteristics of the dataset.

6. Conclusion

- In this phase of the project, the same preprocessing techniques used in the previous phase were applied. The K-best feature selection method was employed and proved to be effective in selecting the most relevant features for the classification task.
- Multiple classification models, namely Logistic Regression, SVM, KNN, Random Forest, and Gradient Boosting, were evaluated, and their performances were compared. The intuition behind this approach was to explore different models with varying characteristics and capabilities,

expecting that some models would outperform others in terms of accuracy.

- The hyperparameter tuning process was also conducted, aiming to optimize the performance of each classification model. By adjusting the hyperparameters, such as regularization parameters for SVM, the number of neighbors for KNN, respectively, the models' performance could be further improved.
- Through experimentation and evaluation, conclusions can be drawn about the effectiveness of each classification model. The accuracy of each model was monitored and compared, and it was observed how the hyperparameter tuning influenced the models' performance.