Formula 1 Race Prediction

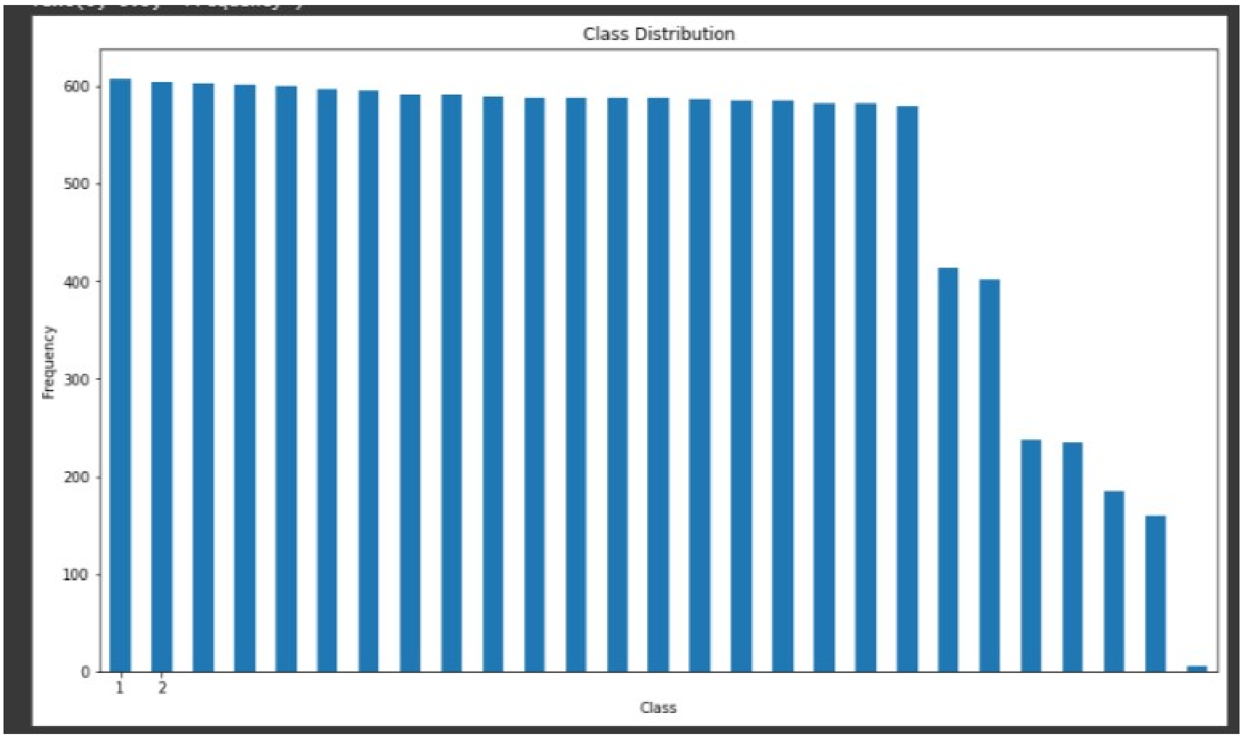
Name:	**Manas Dubey**
Registration No./Roll No.:	19184
Institute/University Name:	IISER Bhopal
Program/Stream:	DSE
Problem Release date:	February 02, 2022
Date of Submission:	

1 Introduction

Before we proceed ,it is essential for us to know a little bit about the game.Essentially, the driver who crosses the finish line first after completing a predetermined number of laps is declared the winner. The first 27 finishers are assigned podium values from 1 to 27.Many races happen in each season.Having said that,My pursuit is to develop a machine learning model to determine the features that contribute the most to victory of a driver in a Formula 1 race.These selected features would contribute the most in estimation the likelihood of a certain driver to win a Grand Prix. The driver with the most likelihood according to an appropriate classification model, utilising these particular selected features, would be mapped as the winner. After which, i can test the performance of my feature selection on a labelled data set which in my case, is the training data.The data given to me is in the form of training data, testing data and training labels in three separate files. The training data file contains the F1 data of races from 1983 to 2019 except for the data of 1998 season and the 2012 season. The excluded data (season 1998,2012) forms the test data which is unlabelled, for which i will reproduce the labels.The features are a mix of Categorical, Boolean ,integer and continuous variables which were further encoded through one hot encoding and label encoding. After encoding and further pre-processing by checking for missing data (none), dealing with negative values, I ended up with 98 columns which will serve as my features and 13,475 rows which are my training instances. The class labels (y train) are integers from 1 to 27 which represent the position on the podium for a particular driver after a particular race. Training is being done on 13,475 such races. After evaluating the distribution of individual classes I notice that they are almost evenly distributed ,except for labels 24,25,26,27.Ideally, since all races should have all the 27 labels, but it is not the case here. This might be so, because in some races, certain drivers are not able to finish due to car failures, crashes and faulty machinery. This unpredictability is one of the major hindrances of my prediction process.

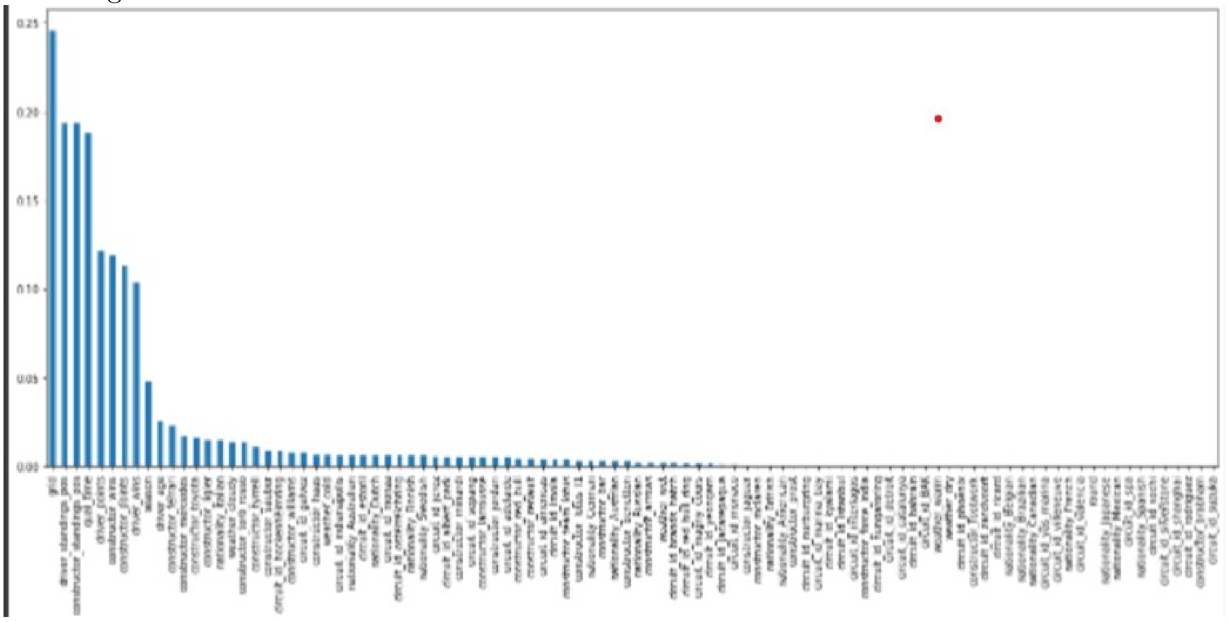
Tasks at hand :-

(i) An assortment of highly important features for prediction and an analysis of the same.
(ii)A classification model for prediction and an analysis of the same.
(iii) Generation of labels
(iv) The prediction of the Winner(1st) for each round in each season.

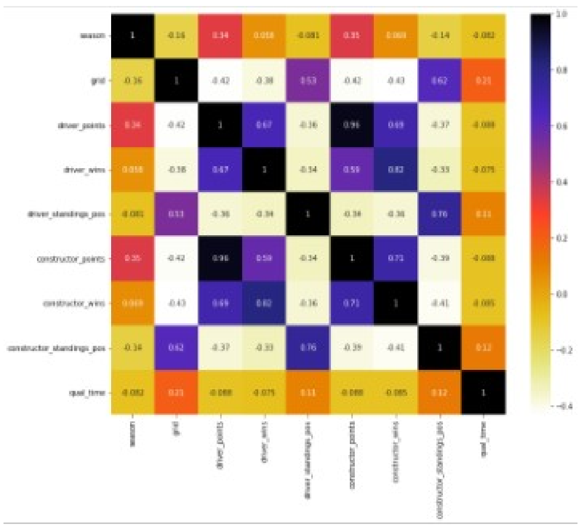


2 Methods

Going ahead with feature selection, i chose Mutual Information gain and chi2 statistics as my feature selection framework.[1] High mutual information indicates a large reduction in uncertainty; low mutual information indicates a small reduction; and zero mutual information between two random variables means the variables are independent.After running the Mutual Information selection framework i got the following.



I got grid, driver standings position, constructor standings position,qualifying time, driver points, constructor wins, driver wins and season as our best 9 features which were ranked the top 9 on both of the feature selection frameworks. This tells us the importance of these features.Moving ahead, one needs to account for multi-collinearity, which can make the model suffer .

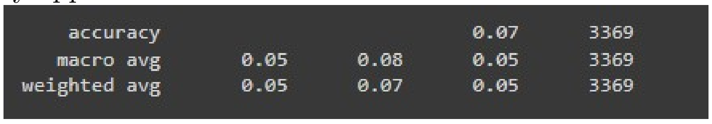


There through Pearson correlation method, i created my own function through Seaborn which would not only plot a heatmap as given above but also sifted out features which had more than 85 percent correlation with each other.After deleting the relevant feature i went ahead with classification.

Now in classification, a pipeline was created which had an assortment of some of the run of the mill classifiers such as k-means,SVM,Logistic Regression, Decision Trees, Random Forest and AdaBoost Classifier. This pipeline would first do a Stratified k-Fold train-test split ,followed by a grid search with relevant hyper-parameter tuning of the above classifiers. The pipeline would also print the classification report and the confusion matrix.

3 Experimental Analysis

After observing the results performed on this data set (27 class labels) I was getting a pretty bad results.Even the best classifiers weren't performing well enough, I was getting pretty low accuracies and macro average f1 scores.For e.g. for Multinomial Naive Bayes my f1 macro avg. score was around 9 percent. Same went for SVM, Random Forest (11 percent) etc. Therefore , I had to devise a better strategy if i was to tackle this problem. Since, I had to just give the winner, i went with a one vs. many approach.



```
    accuracy                           0.07      3369
   macro avg       0.05      0.08      0.05      3369
weighted avg       0.05      0.07      0.05      3369
```

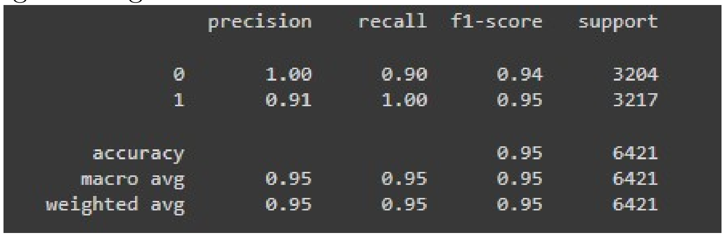
3.1 One vs. Many

I mapped all the labels which corresponded to first on the podium to 1 and the rest of the positions on the podium were mapped to 0. Now i have a binary classification task at hand.I performed feature selection (mutual classification) on these new class labels and i got the same best 9 features as before. After which i ran the classification pipeline again. To my surprise, I got much better results than before in terms of accuracy. But on closer look, i found out that this high accuracy was due to an obvious imbalance of data. After the mapping of the y train, i had at my disposal around 600 1's and 13,000 0's. Therefore, any model would learn to predict the majority class and those predictions would be mostly true, but the problematic concern is when a 1 is predicted as 0. That means, when a true winner is predicted as loser. Therefore, i had to minimize the number of False Negatives. The representative of that in the classification report would be the recall of the minority class.

After making this observation, I got my best classifiers as Random Forest, Multinomial Naive Bayes and SVM as best performers.The best parameters for SVM were the radial basis function, c parameter is 0.1. The f1 macro avg score was 0.60 and recall of 1 was 0.24.To solve the problem of disbalance i went re-sampling of the data, oversampling.

3.2 Resampled Output

I used a combination of undersampling and oversampling using SMOTTomek feature. In the literature, Tomek's link and edited nearest neighbours are the two methods which have been used and are available in imbalanced-learn library. After this, my class distribution evened out to 12000 per class. Over this K Means classification showed brilliant results. The hyper parameters being n neighbours =5, distance being the weight metric.



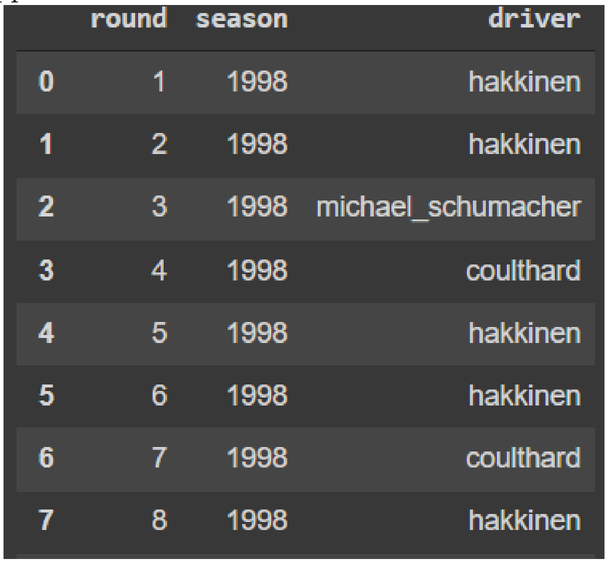
```
              precision    recall  f1-score   support

           0       1.00      0.90      0.94      3204
           1       0.91      1.00      0.95      3217

    accuracy                           0.95      6421
   macro avg       0.95      0.95      0.95      6421
weighted avg       0.95      0.95      0.95      6421
```

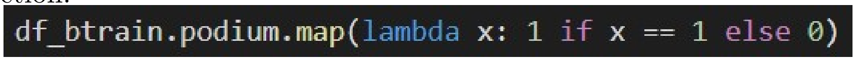
4 Discussions

I created a function called Winner Extractor which would take the trained classifier and predict the probability of a data point in test data belonging to a certain class and for each season for each race based on the prediction model. After that I would sort the probabilities of all players of belonging to class 1, i.e. being a winner.[2] The driver bagging the highest probability of winning would be marked as winner. The function would return a pandas data frame of the above. The diagram below is a snippet.



	round	season	driver
0	1	1998	hakkinen
1	2	1998	hakkinen
2	3	1998	michael_schumacher
3	4	1998	coulthard
4	5	1998	hakkinen
5	6	1998	hakkinen
6	7	1998	coulthard
7	8	1998	hakkinen

After comparing the winner predictions of multi-class vs. sampled binary class labelled model we found that the later predicted the winning drivers more accurately after comparing the results of true winners from the internet. KNN giving the most accurate results in the latter. In order to test the accuracy of our model ,you will have to map your test labels to 0 and 1 also through the below function.



```
df_btrain.podium.map(lambda x: 1 if x == 1 else 0)
```

References

[1] Jerome H. Friedman, Robert Tibshirani, and Trevor Hastie. *The Elements of Statistical Learning.* Springer, second edition, 2008.

[2] Veronica Nigro. Formula 1 race predictor, Jun 2020.