**Data Science Internship | Jai Kisan Case Study**

**Report of Approach**

|  |  |
| --- | --- |
| Name | Uma T V |
| Email | uma.tv1699@gmail.com |
| Institute | Indian Institute of Technology Madras |

**Data Loading and Visualization**

* All required libraries were imported
* The given datasets were loaded
* Visualize the data: We noticed that:
  + Some values in lab\_and\_vitals were missing
  + There were some subjects in lab\_and\_vitals which did not exist in the mrn of baselines.
  + Most of the features in baselines were categorical.

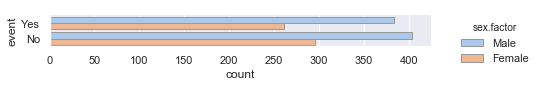
**Handling Missing Values**

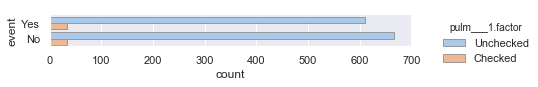
* The number of missing values in every column of the datasets was calculated. Only the column "value" had missing information in lab\_and\_vitals dataframe. Baselines dataframe did not have any null values.
  + Without value the entire row of the lab\_and\_vitals dataframe is not of any use. Hence, all the rows of the lab\_and\_vitals dataframe with missing value of the column "values" were removed.
* We will only need the lab\_and\_vitals values of the medical record numbers (mrn) in the baselines dataframe. Hence, the subjects from lab\_and\_vitals that aren't present in baselines were removed.

**Merging Dataframes**

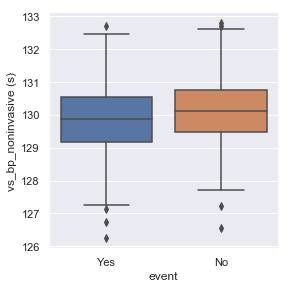
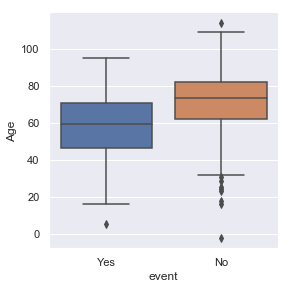
* In lab\_and\_vitals, to deal multiple values of the same test for the same person at different timestamps, the means of the values were taken.
* The tests (lab\_and\_vitals[“name”]) , whose values are given in the dataframe lab\_and\_vitals, were added as columns in baselines. Then, the two dataframes were merged by feeding the information of the lab\_and\_vitals tests for individuals in baselines dataframe, matching the subject column of lab\_and\_vitals with the mrn column of baselines

**Feature Visualization**

* The Features in Baselines were visualized individually. Few visualized graphs are given: 



Categorical Features



Numerical Features

* It was concluded that the individual features didn’t have a lot of influence in predicting the outcome.
* Hence, insignificant features couldn’t be visually removed just on observation as none of the features have evident higher influence that all others.
* Baselines dataframe was split into baselines\_X (features) and baselines\_y (outcome).

**One-hot Encoding**

* The categorical features in baselines\_X were one-hot encoded.

**Normalization**

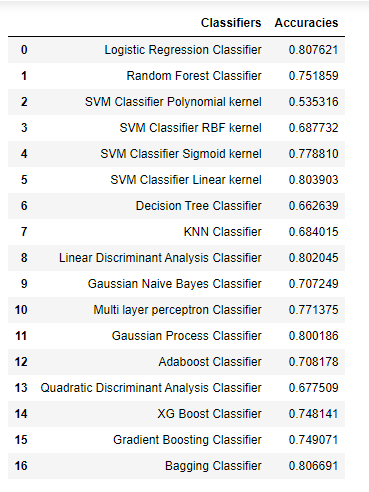
* The data in baselines\_X was normalized.

**Principal Component Analysis**

* PCA was used to extract the most important features (Principal Components) , which captured the maximum cumulative explained variance ratio.
  + The cumulative explained variance ratio of different number of principal components was calculated
  + The cumulative explained variance ratio of the 52 principal components was visualized through a graph of number of principal components vs cumulative variance ratio
  + It was inferred that the first 30 principal components captured the entire cumulative variance ratio.
  + Hence, the number of principal components was taken as 30.

**Machine Learning Models Analysis**

* Baselines\_X and baselines\_y were split into train and test sets in the ratio of 0.8:0.2. Then, various ML classifiers were implemented on the datasets and their accuracy score for the test data were recorded.



* Based on the accuracies information, it was concluded that Logistic Regression performed the best on our data.

**Hyperparameter Tuning**

* The parameters (C and solver) of Logistic Regression Classifier were tuned.

**Bagging**

Bagging classifier was used to further improve the accuracy, using Logistic Regression with tuned parameters as the base estimator.

**Conclusion**

Tuned and bagged Logistic Regression best predicted our data with the model fitting **81.19%** of the entire baselines dataset provided correctly.