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

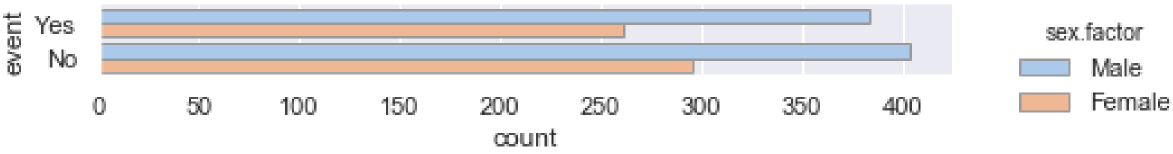
- The number of missing values in every column of the datasets was calculated. Only the column "value" had missing information in lab_and_vitals dataframe. Baselines dataframe did not have any null values.
  - Without value the entire row of the lab_and_vitals dataframe is not of any use. Hence, all 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.

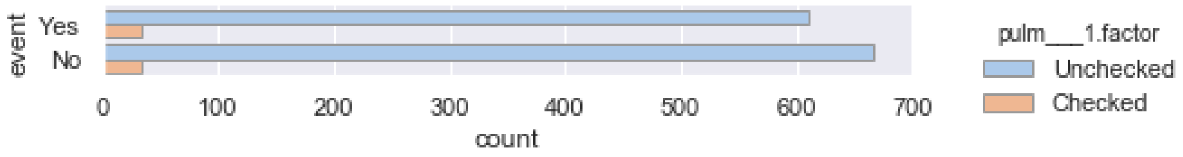
Merging Dataframes

- 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

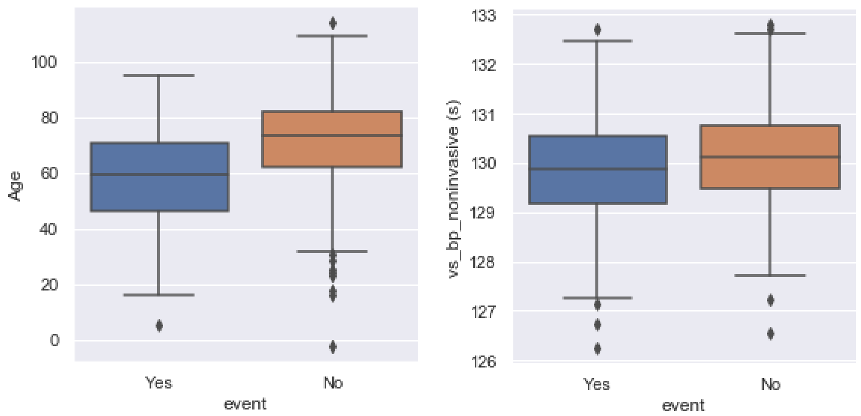
Feature Visualization

- 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

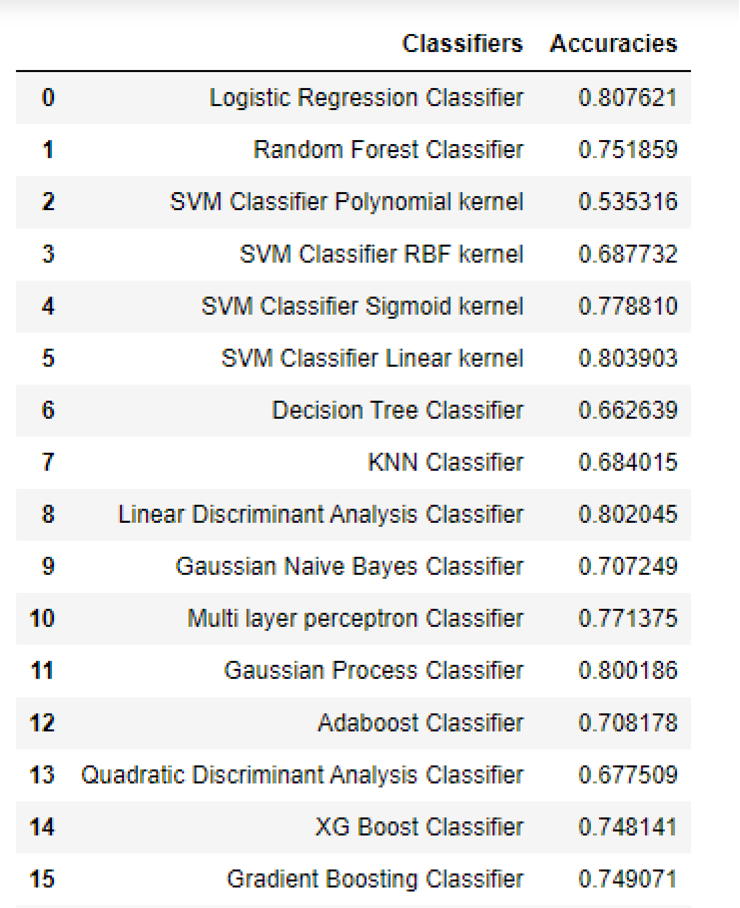
- The data in baselines_X was normalized.

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.



	Classifiers	Accuracies
0	Logistic Regression Classifier	0.807621
1	Random Forest Classifier	0.751859
2	SVM Classifier Polynomial kernel	0.535316
3	SVM Classifier RBF kernel	0.687732
4	SVM Classifier Sigmoid kernel	0.778810
5	SVM Classifier Linear kernel	0.803903
6	Decision Tree Classifier	0.662639
7	KNN Classifier	0.684015
8	Linear Discriminant Analysis Classifier	0.802045
9	Gaussian Naive Bayes Classifier	0.707249
10	Multi layer perceptron Classifier	0.771375
11	Gaussian Process Classifier	0.800186
12	Adaboost Classifier	0.708178
13	Quadratic Discriminant Analysis Classifier	0.677509
14	XG Boost Classifier	0.748141
15	Gradient Boosting Classifier	0.749071

- 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.