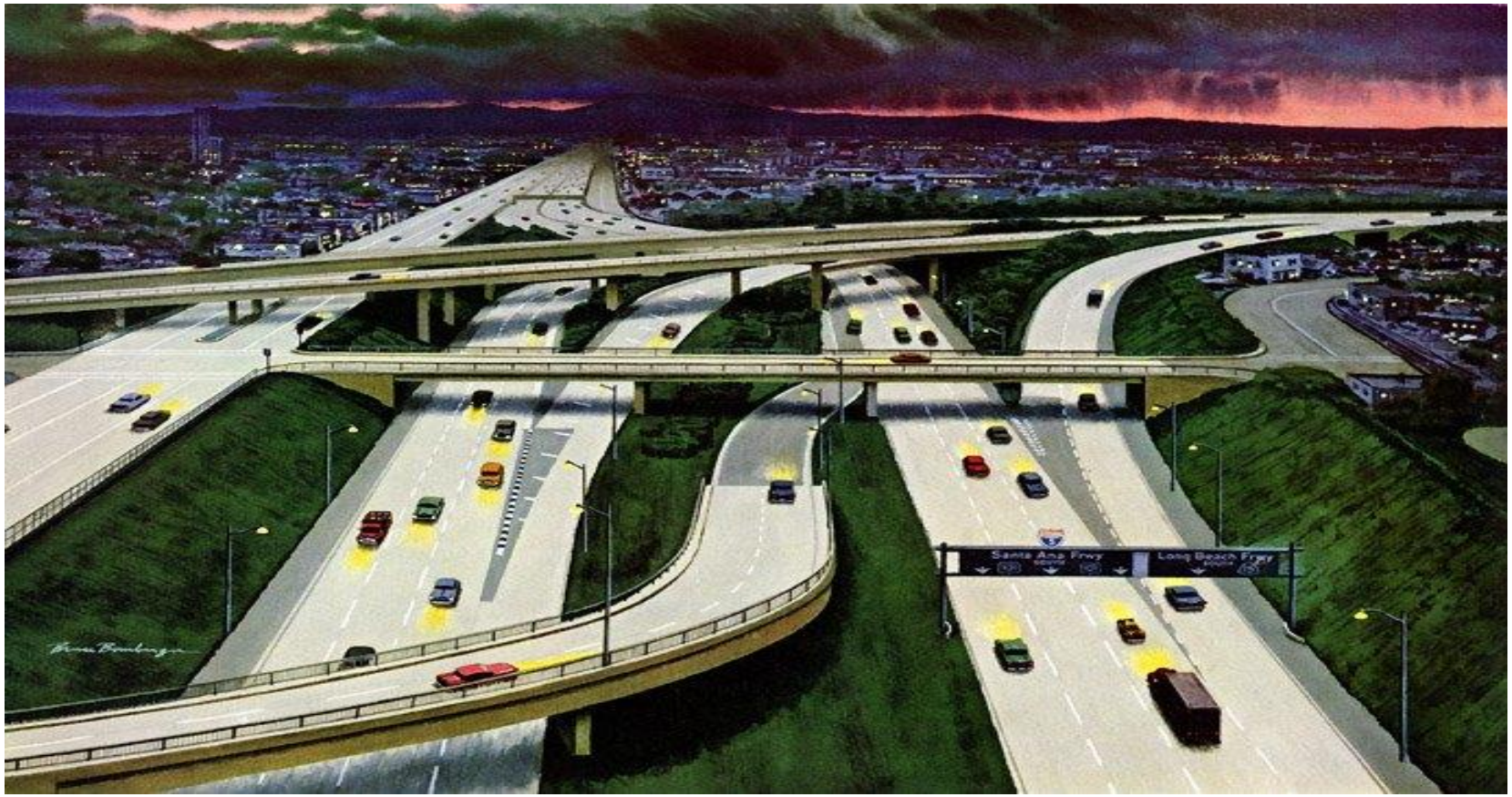
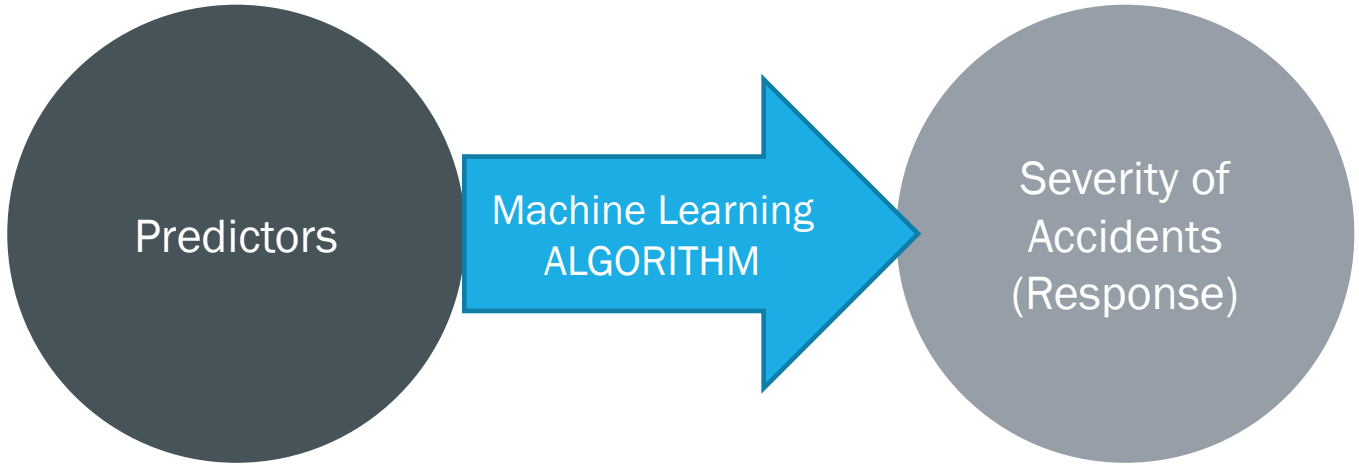
PREDICTING THE SEVERITY OF US CAR ACCIDENTS BASED ON MULTIPLE PARAMETERS



PROBLEM STATEMENT

- Every Year there are thousands of accidents in the United States, which costs the society hundreds of Billions in economic and societal costs
- By analysing the severity of the accidents on various parameters corrective actions could be taken by the State and Judiciary.
- The project divides the severity of accidents in four different parameters, which is our response variable.
- For the purposes of this project we have analysed 47 variables which could affect the severity of the accidents



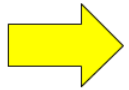
DATA CLEANING

- **Categorical Variables:**
  - There are 14 continuous variables and 33 categorical variables. These variables were given dummy values.
- **Missing Values:**
  - Data such as street number, wind chill, and precipitation had more than 25% of its value missing, therefore we dropped these data sets.
  - For other variables with missing values, mean was taken to fill the data. For categorical missing variables we dropped the row. Our dataset is very large, so we dropped them.
  - We removed Start_Time and End_Time and replaced it with Time duration. Severity depends only on the time duration.
- **Bin Generation:**
  - We had around 13 Boolean variables and they were given binary value of 0 or 1 using labelBinarizer().
  - The other categorical variables were encoded by using one hot encoding. This creates a binary column for each category and returns a sparse matrix or dense array.
- **Standardization and Transformation:**
  - Values with high numbers could cause biases, therefore variables like pressure, humidity, Visibility, Windspeed etc. Was standardized using Min-Max Scaler.
- **Used Random Function to reduce the size:**
  - We used the random function to select the data from our data set randomly, therefore reducing the size and making it easy for running the models.

```
ID
Severity
Start_Time
End_Time
Start_Lat
Start_Lng
End_Lat
End_Lng
Distance(mi)
Description
Number
Street
Side
City
County
State
Zipcode
Country
Timezone
Airport_Code
Weather_Timestamp
Temperature(F)
Wind_Chill(F)
Humidity(%)
Pressure(in)
Visibility(mi)
Wind_Direction
Wind_Speed(mph)
Weather_Condition
Amenity
Bump
Crossing
Give_Way
Junction
No_Exit
Railway
Roundabout
Station
Stop
Traffic_Calming
Traffic_Signal
Sunrise_Sunset
Civil_Twilight
Nautical_Twilight
Astronomical_Twilight
Time_Duration(min)
```

One Hot Encoding

Color
Red
Red
Yellow
Green
Yellow



Red	Yellow	Green
1	0	0
1	0	0
0	1	0
0	0	1

PRELIMINARY ANALYSIS

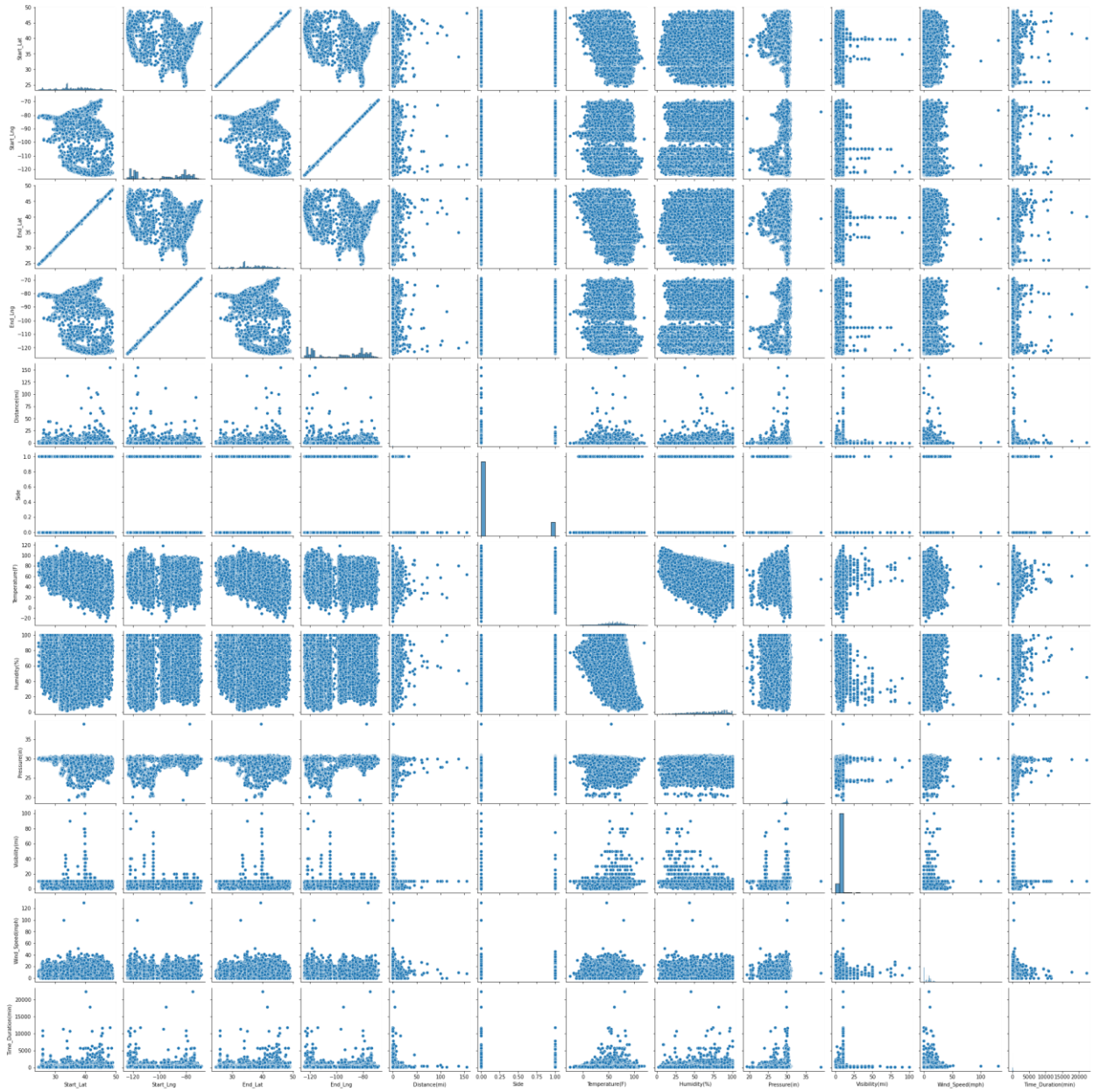
- <u>Predictor-predictor</u>

1) High positive correlations between two pairs of predictor variables can be observed. **Start_Lat and End_Lat** are highly correlated. **Start_Lng and End_Lng** also have collinearity between each other, as the value is higher than 0.7.

2) There is a negative relation between **Temperature** and **Humidity** (-0.4). This is not a strong relation.

- <u>Response-predictor</u>

- Majority of the variables are categorical and hence there is very less presence of multicollinearity in the data.



PRELIMINARY ANALYSIS

- ANOVA

By observing the P values, 32 variables (95% confidence) failed to reject the Null Hypothesis and these predictors were not Statistically Significant.

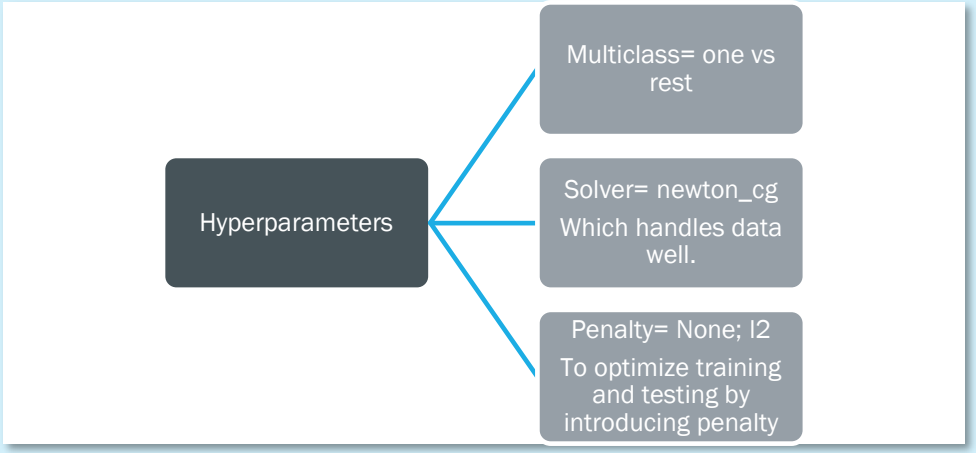
We ignored this table because the R square value was very low. Hence, we considered all the Predictors for our analysis.

OLS Regression Results

Dep. Variable:	Severity	**R-squared:**	0.120
Model:	OLS	**Adj. R-squared:**	0.119
Method:	Least Squares	**F-statistic:**	124.1
Date:	Sun, 28 Nov 2021	**Prob (F-statistic):**	0.00
Time:	00:32:31	**Log-Likelihood:**	-65609.
No. Observations:	88724	**AIC:**	1.314e+05
Df Residuals:	88626	**BIC:**	1.323e+05
Df Model:	97		
Covariance Type:	nonrobust		

MODELS USED FOR PREDICTION

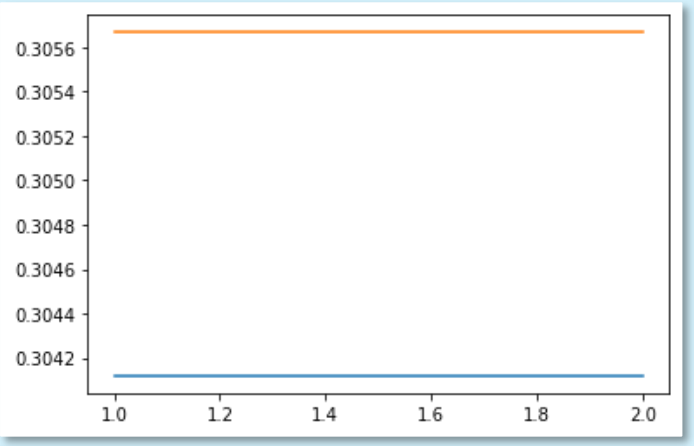
1. LOGISTIC REGRESSION



1. bias-variance trade-off for hyperparameter optimization of logistic regression

Solver	Training Result	Testing Result
newton_cg	0.304	0.305
lbfgs	0.304	0.305
sag	0.304	0.305
Saga	0.304	0.305

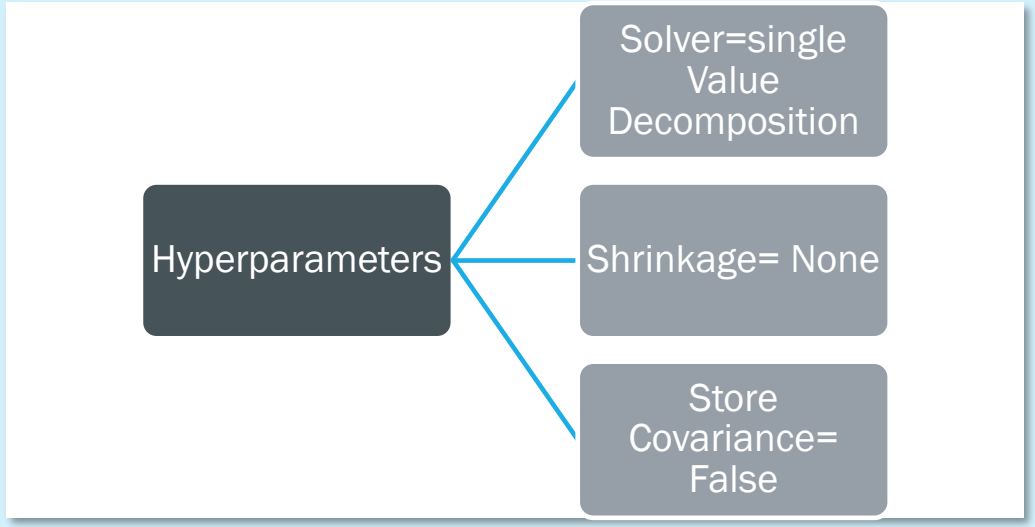
The Model has:
- Low Variance
- Low Bias



MSE for Training and Testing data of Logistic Regression

The Logistic Regression Accuracy is 0.8414 after introducing a penalty ‘l2’

2. LINEAR DISCRIMINANT ANALYSIS

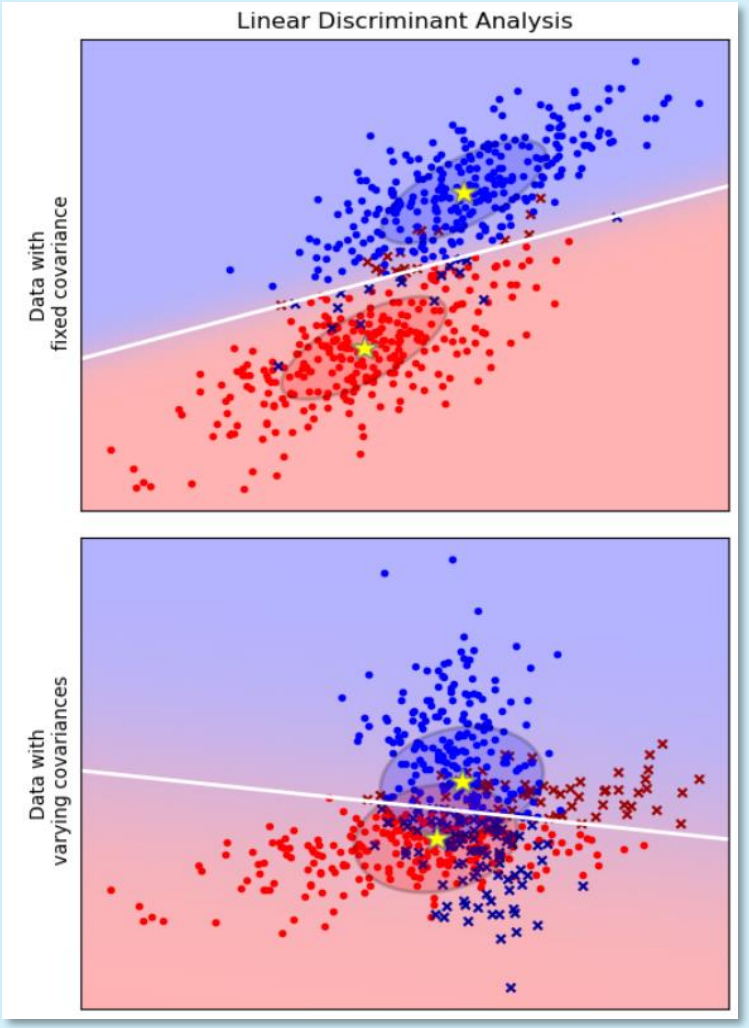


Solver ‘svd’ gave the best result (CV = 10)

The Model has:

- Low Variance
- Low Bias

	Training Data	Testing Data
Mean Prediction Accuracy	0.83	0.82
Mean Squared Error	0.33	0.34



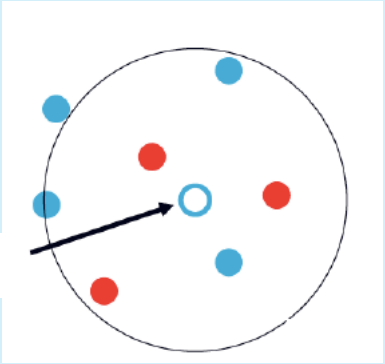
The Linear Discriminant Analysis the precision accuracy is 0.787.

3. K- Nearest Neighbour

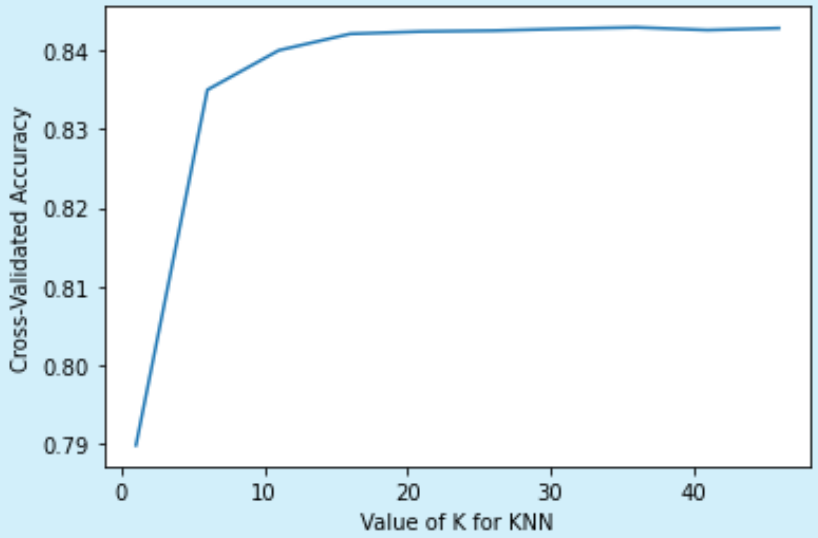
Hyperparameters

- n_neighbours = 31
- Weights= uniform,
  It gives neighbours weighted average

The Model has:
- Low Variance
- Low Bias



Value of K for KNN vs Cross Validated Accuracy

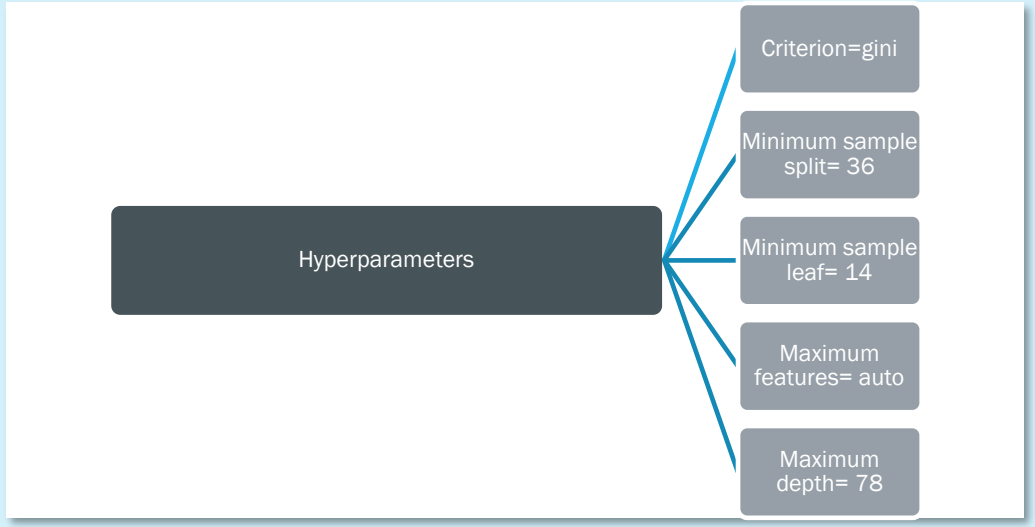


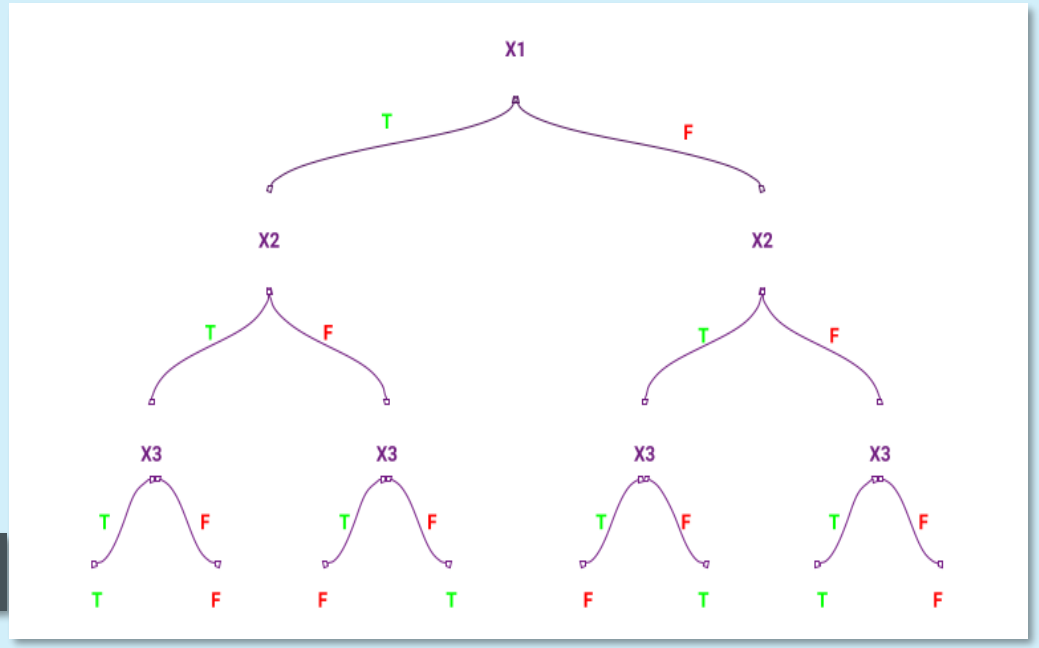
Testing error after optimization of Hyperparameters = 0.84

	Training Data	Testing Data
Prediction Accuracy (CV = 10)	0.850	0.837
Mean Squared Error (CV = 10)	0.303	0.309

The K-Nearest Neighbour has precision accuracy of 0.788.

4. Decision Tree Classifier





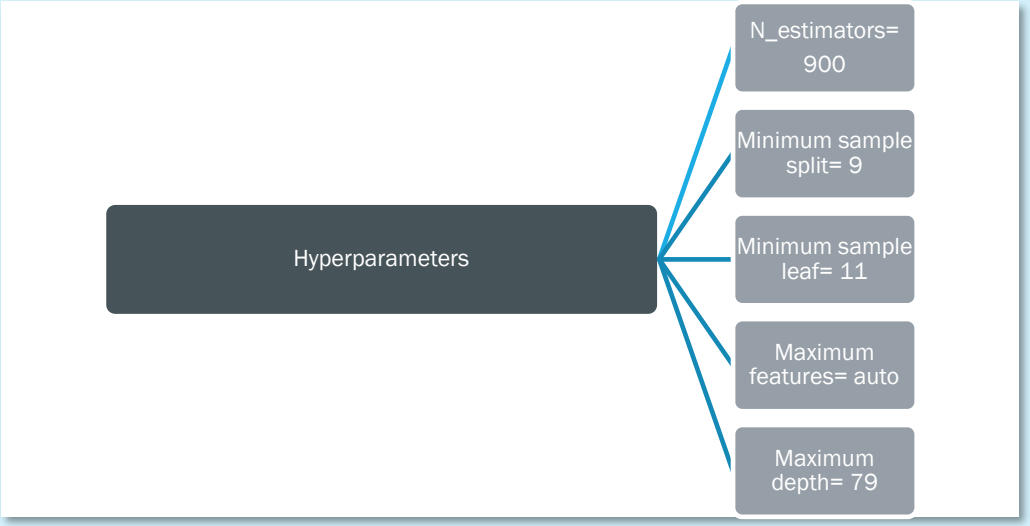
Testing error after optimization of Hyperparameters = 0.845

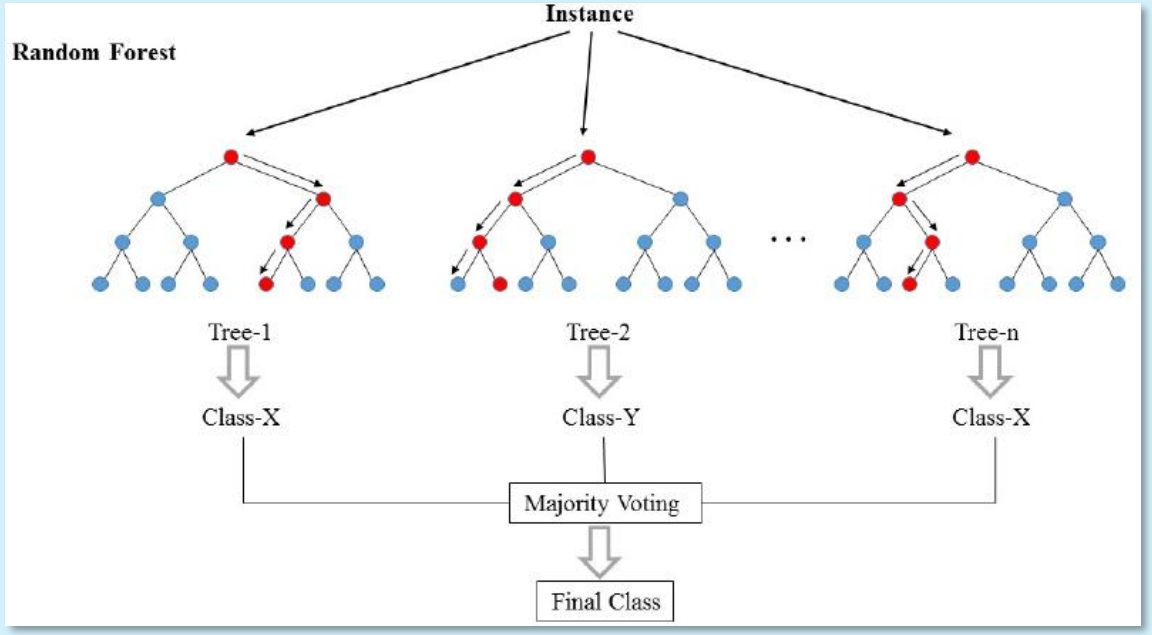
	Training Data	Testing Data
Prediction Accuracy (CV = 10)	0.999	0.833
Mean Squared Error (CV = 10)	0.287	0.311

The Model has:
- High Variance
- Low Bias

The Decision Tree model has precision accuracy of 0.811.

5. Random Forest Classifier





	Training Data	Testing Data
Prediction Accuracy	0.876	0.862
Mean Squared Error	0.264	0.284

The Model has:
- Low Variance
- Low Bias

The Random Forest Classifier has precision accuracy of 0.857.

COMPARING PERFORMANCE OF THE MODELS

Model Name	Training error	Testing Error	Precision Accuracy Score
Logistic Regression	0.307	0.309	0.80
Linear Discriminant Analysis	0.337	0.340	0.787
K- Nearest Neighbour	0.303	0.304	0.788
Decision Tree Classifier	0.287	0.311	0.811
Random Forest Classifier	0.264	0.284	0.857

- K-Nearest Neighbour has the minimum difference between the training and testing errors but the precision score is not so good. The next model with least difference is Random Forest Classifier.
- Random Forest has a training accuracy of 0.876 and testing accuracy is 0.862 which is considered a good fit for a low bias model.
- Also, it has the highest precision accuracy score, so the model explains the variation well, suggesting a low variance model.
- Therefore, Random Forest would be out model choice.

MODEL SELECTION

- Performing the Principal Component Analysis dimension reduction, the dataset was narrowed to 30 principal components, that explained the most useful information for the dataset.

- When we applied PCA we observed from the Cumulative variance Plot that almost 19.82% of our explained variance of our feature data was explained by the PC1 component, 17.86% of the explained variance was explained by the PC2 component, 5.6% of the explained variance was explained by the PC3 component, 4.46% by PC4,….PC30. So, to fit our final model, we consider only 30 principal components to be significant and hence, the reduced feature set is concatenated into 30 principle components. These 30 principal components together explain about 91% of the explained variance in our feature dataset.
- As the difference between the training and testing accuracies is higher in the reduced dataset, it means that the model has not improved in performance in predicting the testing data. This model has high bias and variance when compared to the old model

Full Dataset	Reduced Dataset using PCA
Mean Accuracy (Training): 0.876	Mean Accuracy (Training): 0.888
Mean Accuracy (Testing): 0.862	Mean Accuracy (Testing): 0.842
Precision Avg. Score: 0.857	Precision Avg. Score: 0.809

CONCLUSION

Our dataset is was very large with just 1 million observations. With large data, the model will not be able to accommodate all the predictors but the model performed well on the chosen model.
For our analysis, we fitted the model on Logistic Regression, LDA, K-Nearest Neighbor, Tree Classifier and Forest classifier
From our findings and analysis, we found that Random Forest Classifier outperforms the other models as a good fit to our model.

For the Forest Classifier, the model has been fitted considering all optimized hyperparameters between the input and output function. The training accuracy of 87% and testing accuracy of 86%, cross validated with a score of 86%, which can been expected to perform well on unseen data.

After performing the Principal Component analysis, by decreasing the model to 30 features that explains the best variance in the model. The Random Forest classifier improves as the best model with accuracy score on training as 89%, and 84% when validated on testing data. The original model performed better, hence dimension reduction is not required.