Candidate Selection Prediction

Problem Statement and Expectation

Problem : You are a data scientist of a big MNC who usually hires more than 10k candidates every year. To complete the task they conduct more than 1 lakhs interviews every year.

Expectation : Using data analysis, we want to predict who gets hired after an interview to make hiring decisions better and faster.

Data Facts and Preprocessing

Data Gathering

Data includes important details like :-

- candidate id
- interview duration
- late joining
- candidate profile

Data Cleaning

The dataset contains a small number of missing values.

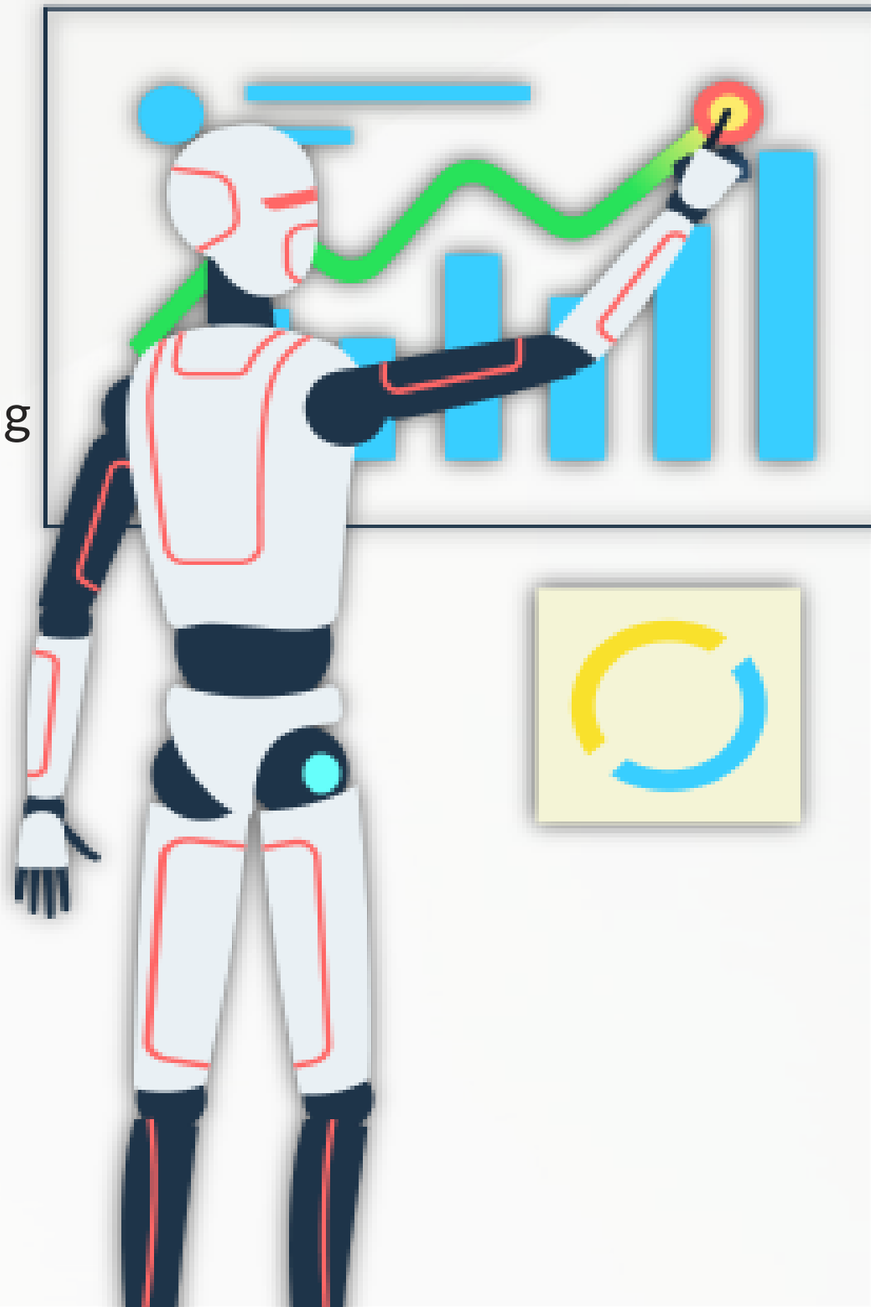
For categorical columns, I used the mode, and for numerical columns, I employed the mean. Additionally, there were no duplicated values present.

Exploratory Data Analysis (EDA)

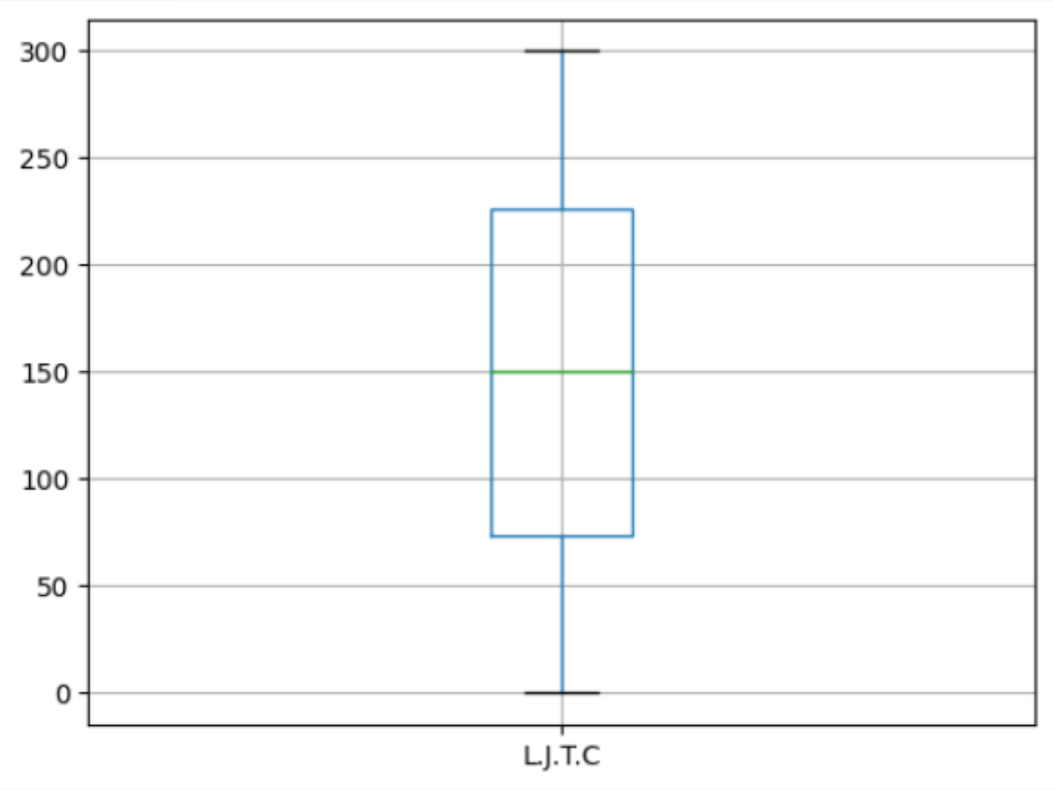
- The dataset contains With 5800 rows and 26 column including with target variable.

- So with the Describe Function we saw that Interview duration
  - mean : 37 minutes
  - standard deviation : 13 minutes
  - shortest interview : 15 minutes
  - longest Interview : 60 minutes

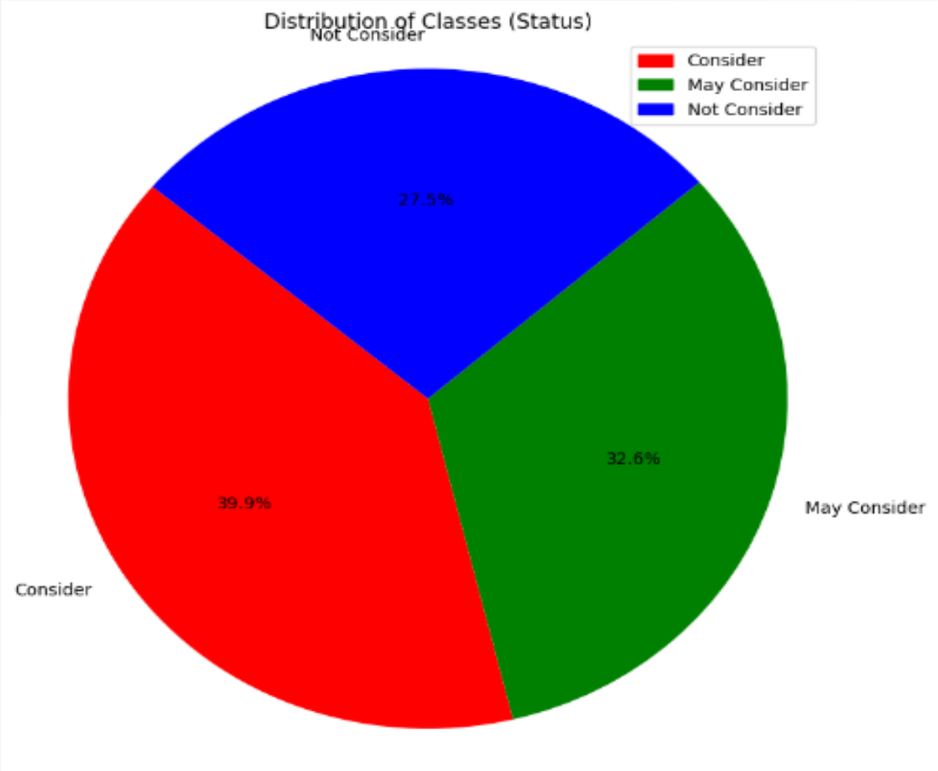
- This summary provides an overview of the numerical characteristics and range of values present in the dataset columns.



Boxplot

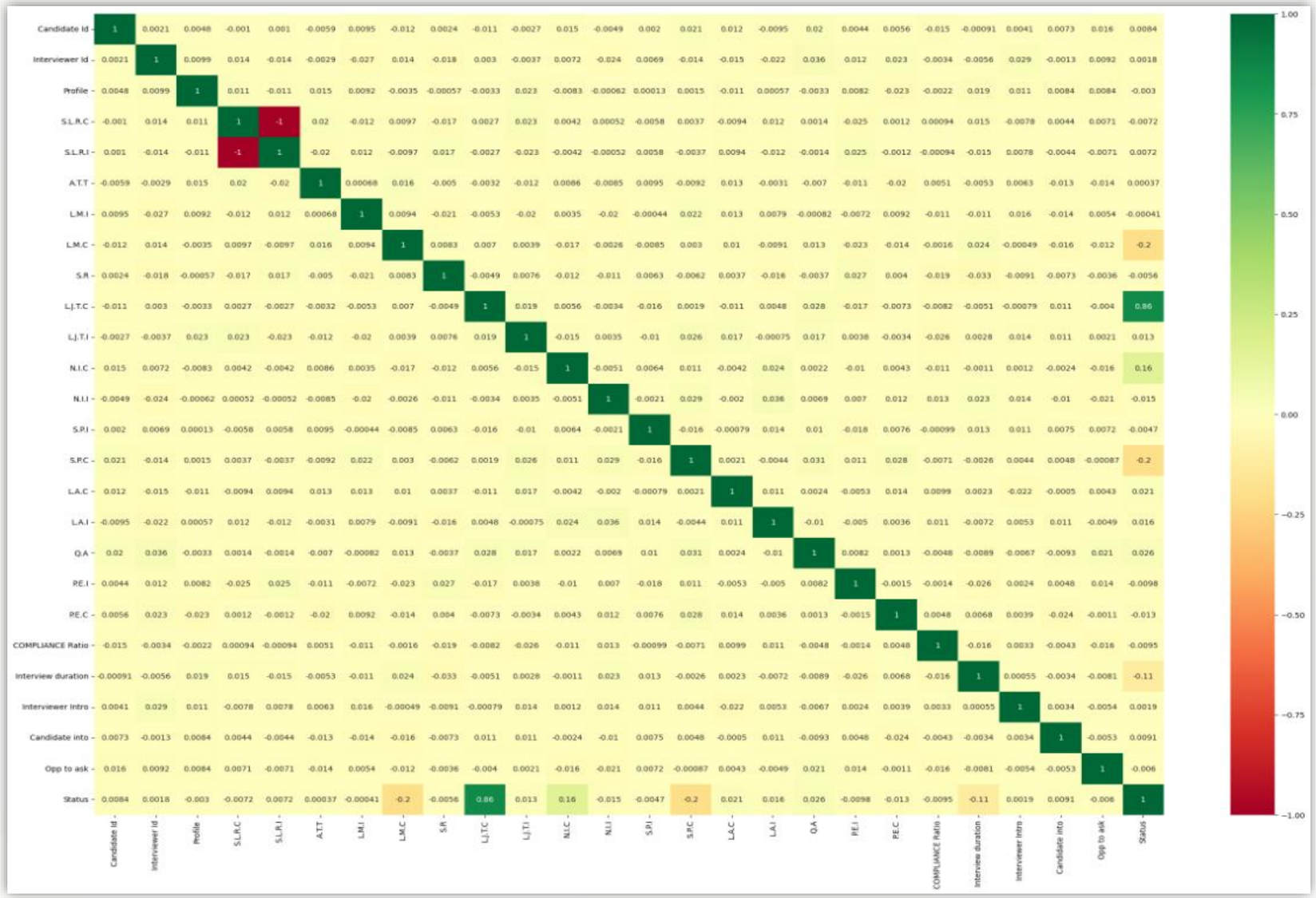


Pie chart



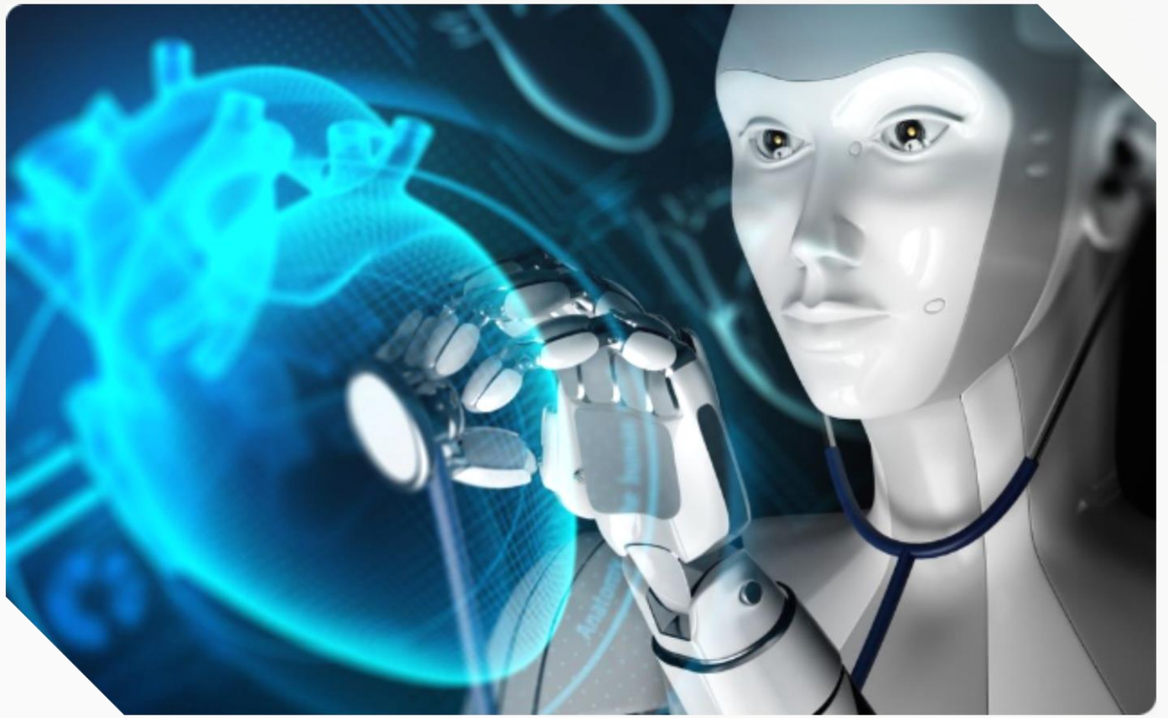
I carefully examined all 26 columns of the data and didn't find any outliers in it. The classes of our target variables also have equal distribution, similar sizes, which is beneficial for our analysis.

Heatmap Correlation



- I can see that five columns in my data really matter for the target column status.

Feature Engineering



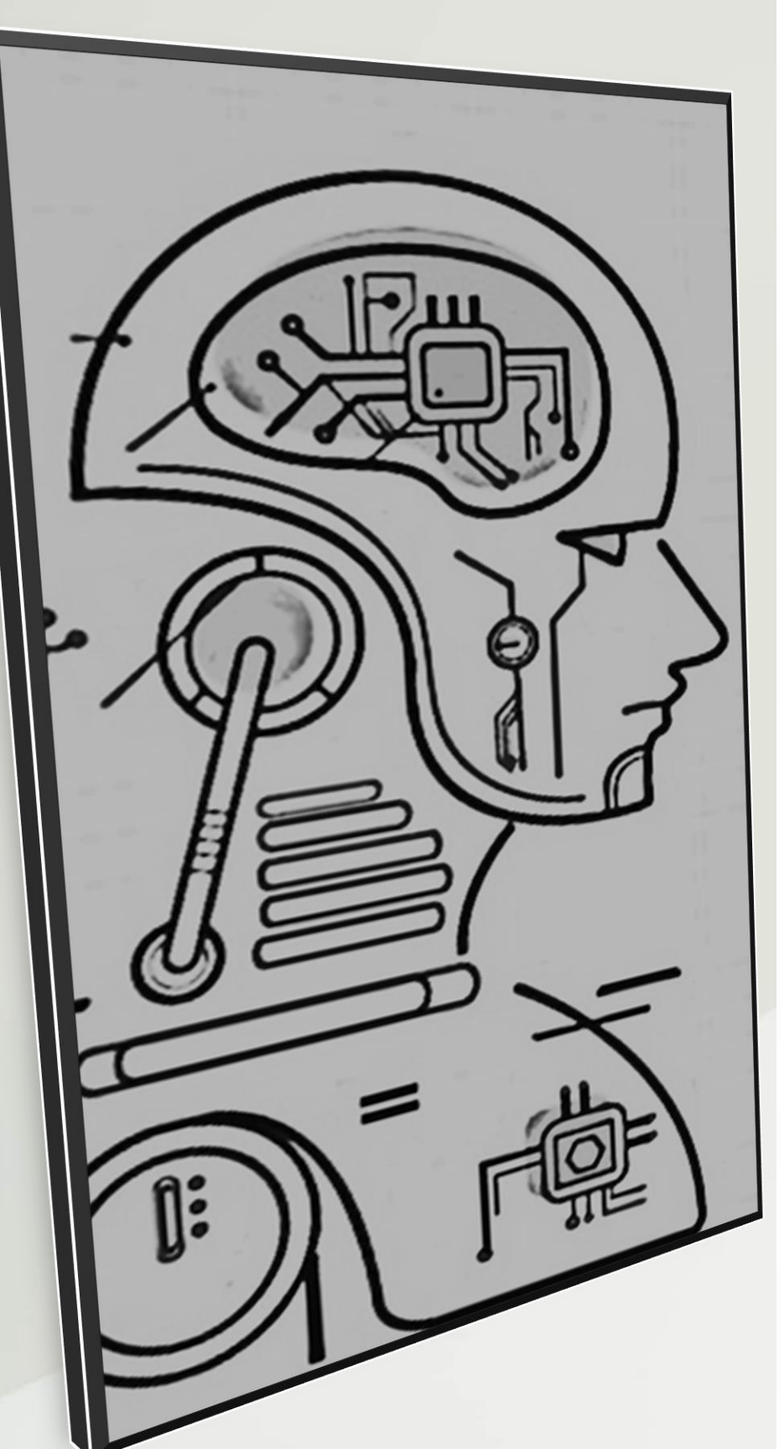
1 Feature Transformation

I applied the Standard Scaler technique to transform the data.

2 Feature Selection

The columns (L.M.C, L.J.T.C , N.I.C, S.P.C, L.A.C, Interview duration and Status) based on a heatmap correlation analysis.

This approach has helped me for accurate modeling and insightful decision-making in the context of the interview process.



Model Selection and Hyperparameter Tuning

- I split the data into training and testing sets with an 80 : 20 ratio.
- I evaluate different machine learning algorithms like Logistic regression, Decision Tree, Random Forest, K-Nearest Neighbors and SVC.
- Hyperparameter tuning — grid search cv

Accuracy on Training and Test Data

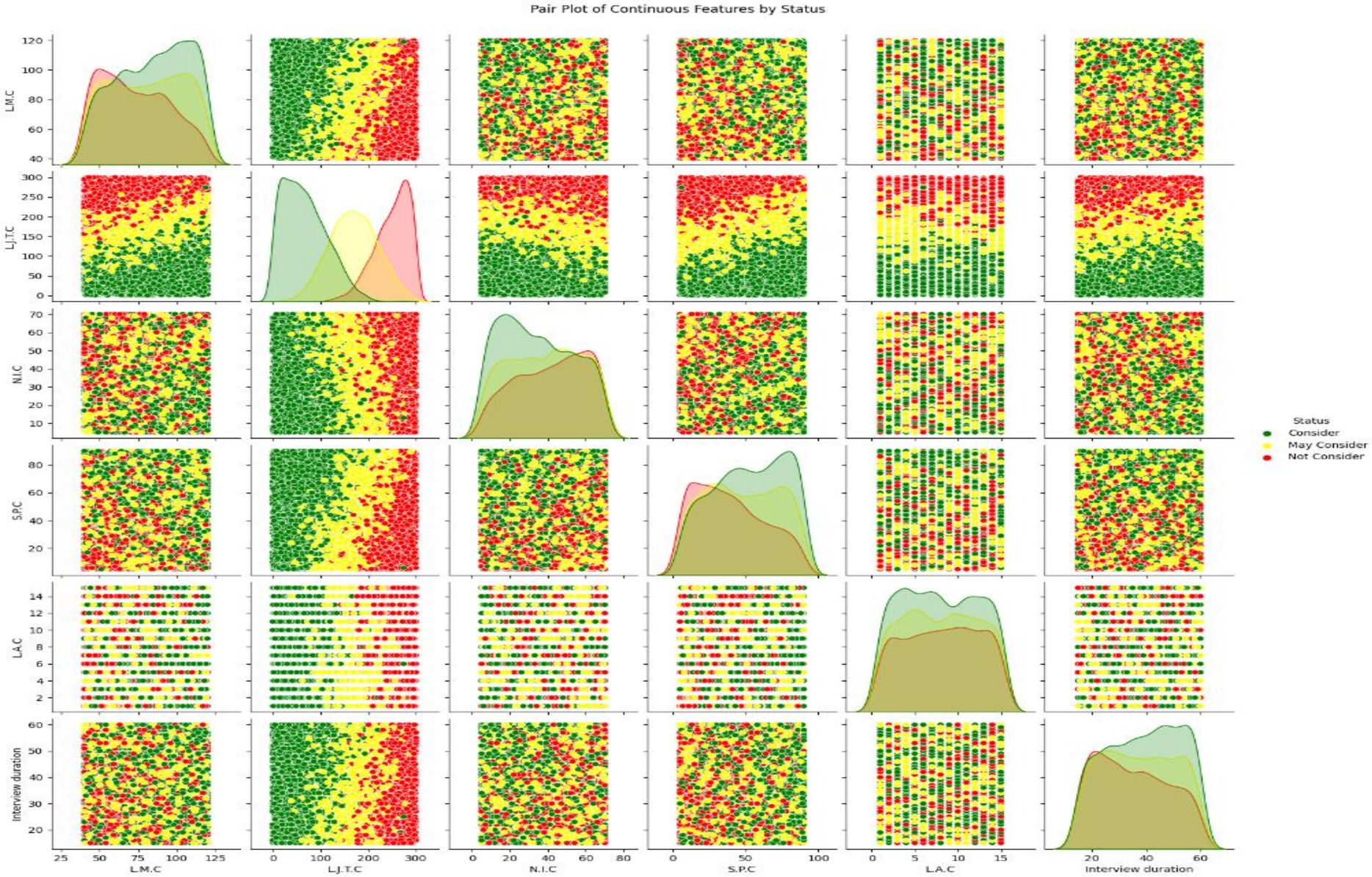
Model	Training	Test
• Logistic regression	0.9968	0.7091
• Decision Tree	0.9293	0.6483
• Random Forest,	0.9241	0.6708
• K-Nearest Neigxhbors	0.9422	0.6475
• SVC	0.9974	0.7075

- Logistic regression and SVC is performing better almost 71% accuracy.

Challenging Part :

- The most challenging part of the project is to pick the right features for my model.
- Tuning logistic regression takes a lot of time to find the exact value for the best parameter.
- But I haven't reached the best accuracy yet. Maybe I can do better with deep learning or other methods, but I’m not familiar with that, because also of less domain knowledge.

Pairplot - Visualizing Relationships



Conclusions

- We achieved a 70.91% accuracy by removing unnecessary features and cleverly handling missing data. Logistic Regression performed well in predicting Status.
- I found that (Interview Duration) How long a candidate takes in the interview affects the model's performance.
- Late joining candidates (LJTC) also impact the likelihood of not getting selected.
- Longest amount of time spoken (LMC) column is also impacting for selection, as speaking for a long time might lead to their selection.
- This method helps avoid biases, saves time for recruiters, and ensures fair hiring.