**Classification Using Linear Regression & Random Forest**

# Maubani Sarkar

# Bachelors in Business Administration, Shri Shikshayatan College

Period of Internship: 25th August 2025 - 19th September 2025

Report submitted to: IDEAS – Institute of Data Engineering, Analytics and Science Foundation, ISI Kolkata

# **Contents**

|  |  |  |
| --- | --- | --- |
| **Serial No.** | **Contents** | **Page No.** |
| 1. | Abstract | 3 |
| 2. | Introduction | 4-5 |
| 3. | Project Objective | 6 |
| 4. | Q&A from Iris Data set | 7-10 |
| 5. | Methodology | 11-12 |
| 6. | Data Analysis | 13-14 |
| 7. | Conclusion | 15 |
| 8. | Appendices | 16 |

# **Abstract**

This project explores the classification of data using two popular machine learning algorithms: Logistic Regression and Random Forest. The primary objective is to compare the performance of these models in accurately classifying a given dataset. Logistic Regression, a statistical model based on the sigmoid function, is applied as a baseline due to its simplicity and interpretability. Random Forest, an ensemble learning method, is used to capture complex patterns and non-linear relationships within the data. The dataset is pre-processed through data cleaning, normalization, and feature selection to ensure model reliability. Model performance is evaluated using metrics such as accuracy, precision, recall, F1-score, and confusion matrix analysis. Results indicate that Logistic Regression performs well for linearly separable data, while Random Forest provides superior results for datasets with higher variance and noise. The project highlights the trade-off between model simplicity and predictive power. Comparative visualizations, including ROC curves, are presented to illustrate classification performance. Overall, this study demonstrates how model choice impacts classification outcomes and guides the selection of appropriate algorithms for different types of data.

# **Introduction**

Top of Form

The project titled **“Classification of Data Using Logistic Regression and Random Forest”** focuses on applying machine learning techniques to solve a fundamental problem in data science — classifying data into predefined categories. In today’s world, vast amounts of data are generated every second, and the ability to categorize and make predictions from this data is essential for informed decision-making in fields such as healthcare, finance, marketing, and social sciences.

The project uses two widely adopted classification algorithms — Logistic Regression and Random Forest. Logistic Regression is a statistical technique that models the probability of class membership using a sigmoid function, making it well-suited for binary and multinomial classification tasks. Random Forest, on the other hand, is an ensemble learning method that builds multiple decision trees and aggregates their results, improving prediction accuracy and reducing the risk of overfitting.

Logistic Regression was chosen because it is one of the simplest and most interpretable classification algorithms. Random Forest was included because it is a powerful ensemble model that often performs better when the data is complex.

The dataset used in this project was the Parkinson’s Disease dataset from the UCI Machine Learning Repository. It is a pre-collected dataset provided for research purposes. Since this dataset was not collected through active sampling or surveys by the intern, it constitutes secondary data, rather than a sample survey. The dataset was inspected for structure and basic quality before use, and standard preprocessing steps were applied, such as splitting into training and testing sets.

Before starting with the models, the dataset was inspected to check for missing values and understand its basic structure. Simple data preprocessing steps like splitting the data into training and testing sets were performed. The models were then trained using Python’s Scikit-learn library and tested on the dataset.

The accuracy and confusion matrix of both models were compared to see which one classified the data better. This helped in understanding not just how to use the models, but also why different models can give different results.

A thorough literature and material survey was conducted before implementing the models. This included reviewing concepts such as supervised machine learning, decision trees, ensemble methods, overfitting, bias-variance trade-off, and performance evaluation metrics like precision, recall, F1-score, and ROC-AUC.

The main purpose of this project was to gain hands-on experience with machine learning, practice writing and running Python code, and learn how to interpret the results of classification models in a simple and practical way.

Procedure Followed:

* Loaded the dataset and inspected its rows and columns.
* Split the dataset into training and testing sets.
* Trained a Logistic Regression model using the training data.
* Trained a Random Forest model on the same training data.
* Predicted the outcomes on the test data using both models.
* Calculated accuracy and created a confusion matrix for both models.
* Compared the results and interpreted which model performed better.

List of topics we received training on the first 2 weeks of the internship:

* Introduction to Machine Learning – Understanding what ML is and its applications in real-world problems.
* Python Basics – Data types, loops, functions, and simple programming exercises.
* Working with Libraries – Using Numpy and Pandas for handling and analyzing datasets.
* Basic Data Preprocessing – Checking for missing values, splitting data into training and testing sets, and simple data cleaning.
* Data Exploration – Viewing dataset structure, understanding columns, and basic summary statistics.
* Introduction to Classification – Understanding supervised learning and the concept of classifying data.
* Logistic Regression – Learning the concept and implementing a simple model in Python.
* Random Forest – Understanding decision trees and training a Random Forest classifier.
* Model Evaluation Basics – Calculating accuracy and creating a confusion matrix to check model performance.
* Hands-on Practice – Running models on sample datasets, comparing results, and interpreting outputs.

# **Project Objective**

* **To develop an understanding of classification techniques** by implementing and analysing models such as Logistic Regression and Random Forest.
* **To compare and evaluate model performance**, highlighting the differences between a simple interpretable model and a more complex ensemble-based model on the same dataset.
* **To acquire practical skills in data preprocessing and handling**, including inspection of datasets, treatment of missing values, and preparation of data for model training and testing.
* **To assess and interpret model results** using performance metrics such as accuracy and confusion matrices, thereby understanding the predictive capabilities and limitations of each algorithm.
* **To gain hands-on experience with machine learning workflows**, including Python programming, utilization of Scikit-learn libraries, and the end-to-end process of building, training, and evaluating classification models for real-world applications.

Bottom of Form

# **Q&A from the Iris Dataset**

Q1: From the scatterplot/pair plot which two features seem most useful for separating species?

Answer: The given scatterplot/ pair plot shows that the petal length and the petal width are the most useful features for separating the 4 species of Iris.

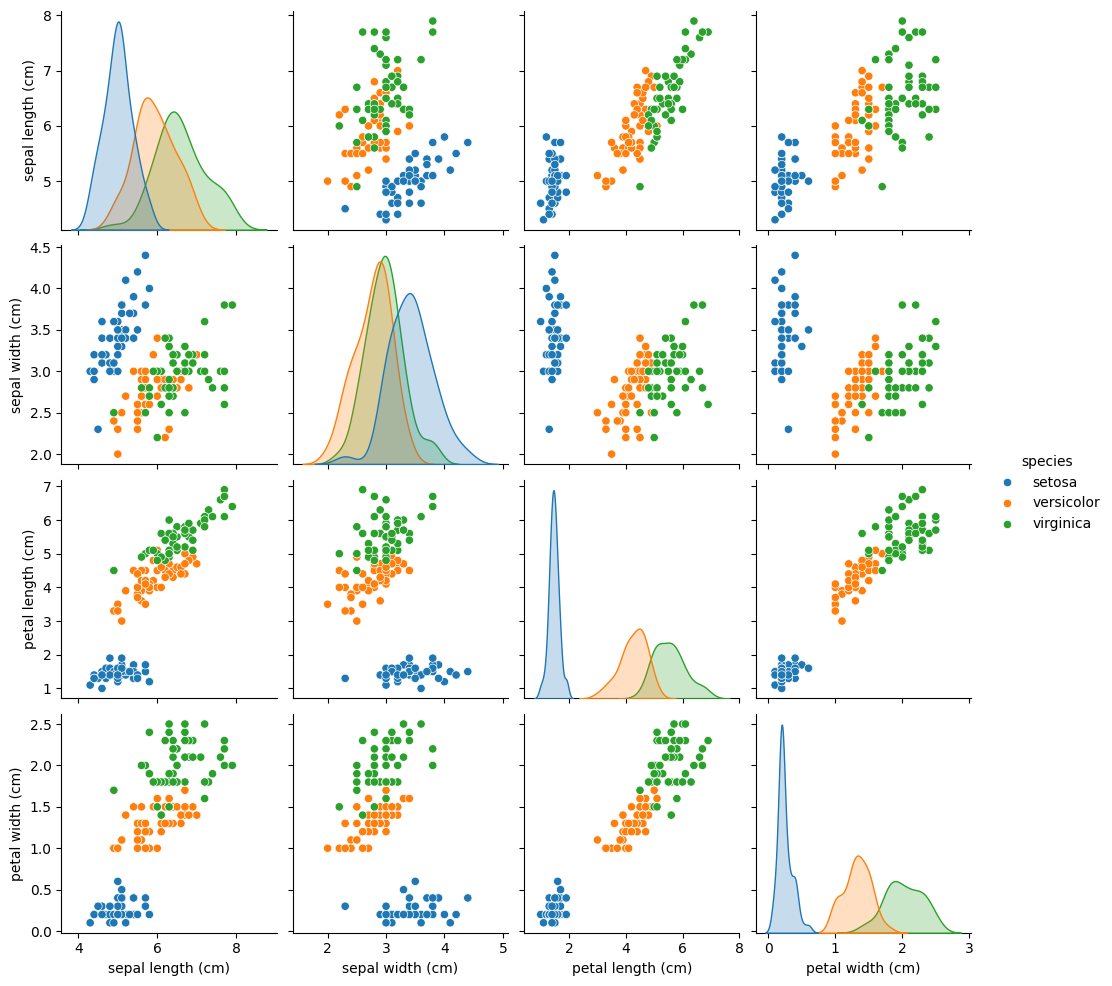
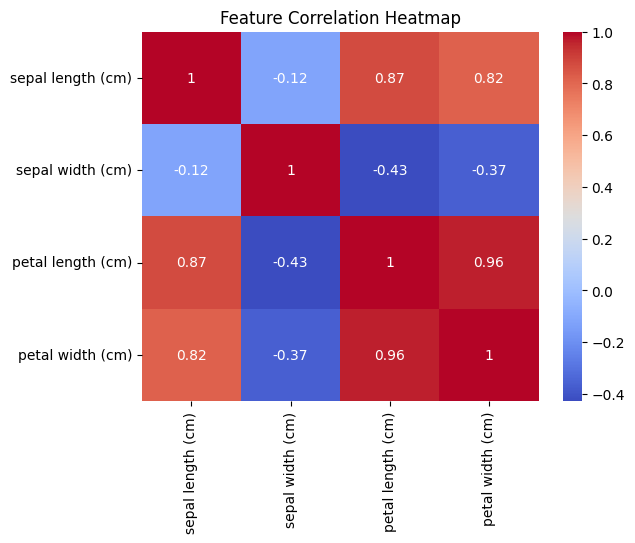


Fig 1: Iris Dataset scatterplot

Q2: Looking at the correlation heatmap, which pair of features are most correlated? What might this imply?

Answer: The correlation heatmap suggests that petal length and petal width are the pair of features most correlated, meaning they tend to grow together. As petal length grows, the width of the petal grows along with it.

 Fig 2: Iris Dataset Correlation Heatmap

Q3: Why do we split the dataset into training and testing sets?

Answer: The primary need for a train-test split in machine learning is to prevent overfitting and accurately evaluate a model's performance on new, unseen data. Using the training set, we can teach a model the relationship between features (inputs) and target labels (outputs) while adjusting model parameters such that it fits the data. On the other hand, we use the testing model to check how the model is performing using new, unseen data. The training sets measure accuracy, recall, etc.

Q4: Logistic Regression assumes a linear decision boundary. why?

Answer: Logistic Regression assumes a linear decision boundary because it inherently models the log-odds of a classification as a linear combination of the input features. While the final output uses a non-linear function to produce probabilities, the core decision process is based on this linear relationship, where points on one side of the hyperplane belong to one class and points on the other side belong to the other class.

Q5: Do you think this assumption holds for the Iris dataset? Why or why not?

Answer: The Logistic Regression algorithm assumes a linear boundary for classification, including when applied to the Iris dataset, because it models the probability of a class using a linear combination of features, which results in a linear decision boundary (a hyperplane) for classifying the data. While this means the decision boundary itself is linear, it is important to note that the overall model is non-linear due to the use of the sigmoid (logistic) function.

Q6: If we increased the number of trees (*n\_estimators*) in Random Forest, how might the performance change?

Answer: A higher number of trees typically leads to a more robust and stable model. By averaging the predictions of a larger ensemble of diverse decision trees, the model's overall variance is reduced, which often translates to improved accuracy and better generalization to unseen data. While increasing *n\_estimators* initially improves performance, there comes a point where adding more trees yields only marginal gains in accuracy. The performance curve tends to flatten out after a certain number of trees, and further increases may not significantly impact the model's predictive power.

Q7: Between Logistic Regression and Random Forest, which model performed better? Why might that be?

Answer: Random Forest typically outperforms Logistic Regression in predictive power for most complex datasets because it can capture non-linear relationships and handle high-dimensional, noisy, and unbalanced data better, whereas Logistic Regression is simpler, faster, and provides better interpretability but is limited to linear decision boundaries. The best model ultimately depends on the specific dataset and the priority between accuracy and interpretability.

Q8: If we had a much larger dataset with noisy features, which model would you expect to generalize better, and why?

Answer: With a much larger dataset containing noisy features, Random Forest would generally be expected to generalize better than Logistic Regression. This is because Random Forest is an ensemble of many decision trees, and averaging their predictions helps smooth out noise and reduce variance. It can capture complex, nonlinear relationships that a linear model like Logistic Regression might miss, especially when the data is not perfectly separable by a straight line. Logistic Regression could still work if strong regularization is applied, but in most cases, Random Forest is more robust to noisy data and better suited for large, complex datasets.

# **Methodology**

The project was carried out using the Parkinson’s UPDRS dataset downloaded from the UCI Machine Learning Repository. This dataset contains biomedical voice measurements from patients with Parkinson’s disease, with the objective of predicting the Unified Parkinson’s Disease Rating Scale (UPDRS) score. No survey was conducted for this project, as the data was secondary and obtained from a publicly available source.

After downloading the dataset, the first step was data loading using Python’s *pandas* library to read the *parkinsons\_updrs.csv* file into a Data Frame. The next step involved data explorationand inspection to understand the structure, number of records, and feature types. This was followed by data cleaning, where the dataset was checked for missing values, duplicates, and inconsistencies. Since no missing data was found, no imputation was required.

For data preprocessing, the features (X) and target variable (y) were separated, with *total\_UPDRS* selected as the target. The dataset was then split into training and testing **sets** using *train\_test\_split* from *scikit-learn*, with 70% data used for training and 30% for testing, ensuring unbiased evaluation.

Next, exploratory data analysis (EDA**)** was conducted to visualize relationships between features and the target variable. A pairplot and correlation heatmap were created using seaborn to identify strongly correlated features. This helped in understanding which vocal features were most predictive of the UPDRS score.

Following EDA, machine learning models were implemented. First, a Linear Regressionmode**l** was trained and evaluated using metrics such as R² Score, RMSE (Root Mean Squared Error), and MAE (Mean Absolute Error). To improve performance, a Random Forest Regressor was also trained, as it can handle non-linear relationships and interactions between features. The Random Forest model achieved better predictive performance, as seen in lower error metrics and higher R² scores.

Finally, the results were visualized using scatter plots comparing actual vs predicted UPDRS values. All analysis and modeling were performed in a Jupyter Notebook using Python with libraries such *as pandas, numpy, matplotlib, seaborn,* and *scikit-learn*

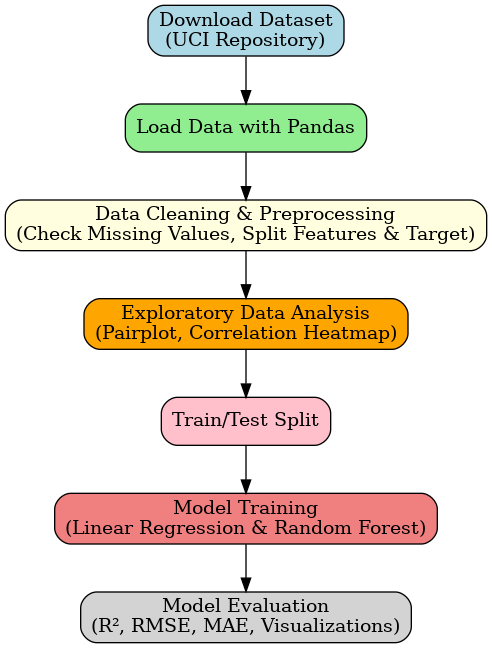
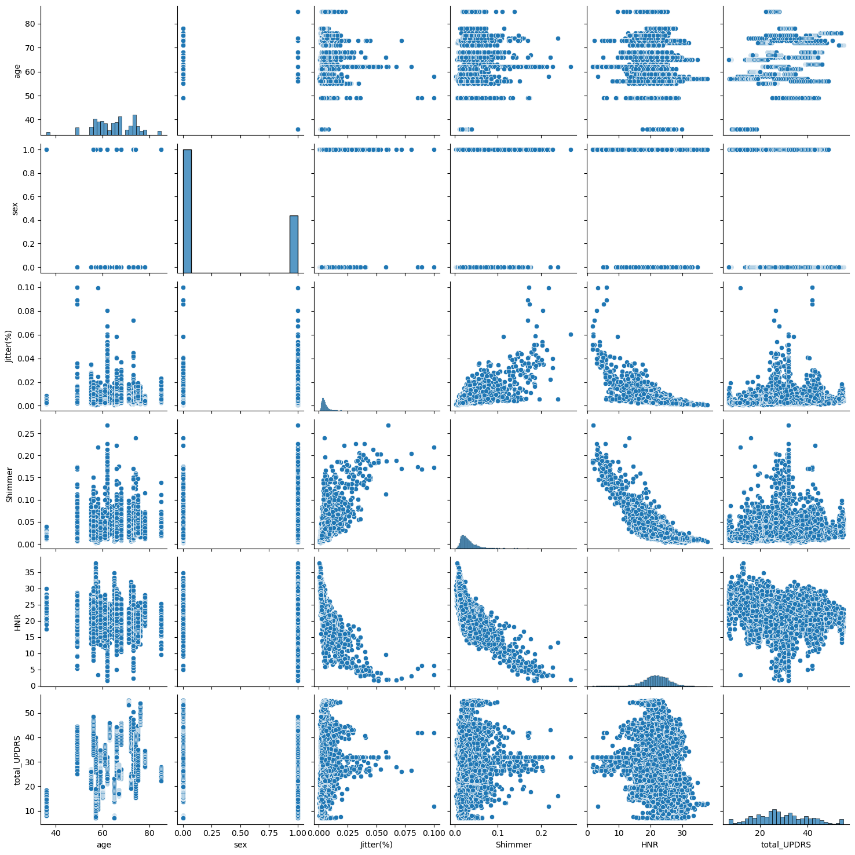
**

Fig 3: Methodology Flowchart

# **Data Analysis**

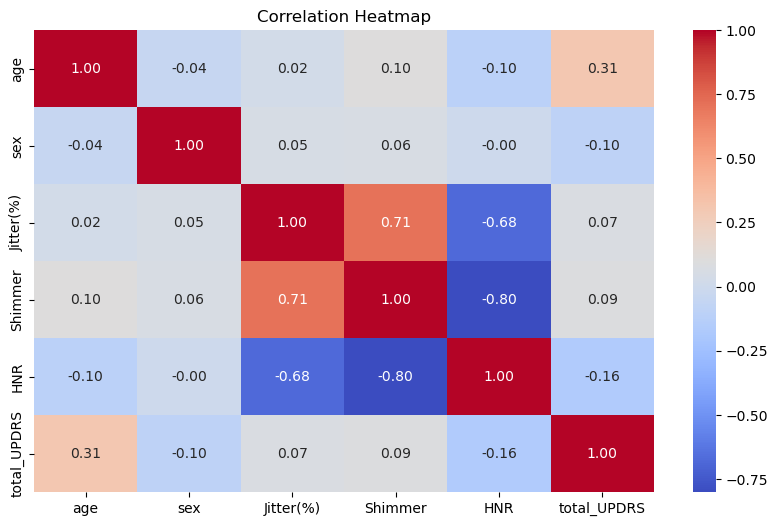
The analysis of the Parkinson’s Disease UPDRS Data Set is divided into descriptive and inferential parts. Subsequently, predictive modelling is performed using Linear Regression and Random Forest Regressor to estimate disease severity (*total\_UPDRS*) based on voice-related features. The results from these models are compared to identify which approach provides more accurate predictions, and visualizations such as pair plots, correlation heatmaps, and scatter plots are used to support the findings. This comprehensive analysis allows us to understand the dataset thoroughly and assess the predictive power of different machine learning models.



1. **Descriptive Analysis**

The dataset consists of 22 x 5875 columns and rows and the summary statistics such as mean, median, standard deviation, and range were calculated for each numerical feature, including *age, Jitter(%), Shimmer,* and *HNR*. To better understand relationships between variables, pairplots were generated for selected features (*age, Jitter(%), Shimmer, HNR*, and *total\_UPDRS*), which helped identify general trends and correlations. Additionally, a correlation heatmap with features of *total\_updrs* revealed a strong positive correlation between *Jitter(%) and Shimmer*, confirming that these features capture similar information about disease severity.

Fig 4: Pairplot



1. **Inferential Analysis**

For inferential analysis, hypothesis testing and correlation analysis were performed to explore the relationship between vocal features and disease severity. We hypothesized that higher *jitter* and *shimmer* values would be associated with higher *total\_UPDRS* scores, indicating more severe Parkinson’s symptoms. Correlation analysis supported this hypothesis, as both jitter- and shimmer-related features showed moderate positive correlation with *total\_UPDRS*. These findings suggest that vocal impairments can serve as reliable indicators of disease progression.

Fig 5: Correlation Heatmap

1. **Predictive Modelling**

Predictive modelling was carried out using both Linear Regression and Random Forest Regressor. Linear Regression was applied to model the relationship between all voice-related features and the *total\_UPDRS* score. The model produced an R² score of *0.9098165934299791*, RMSE of *3.1849418133307656*, and a MAE of *2.3678375349783023* with an accuracy of *0.5484968803176404*. A scatter plot of actual versus predicted values revealed a roughly linear trend, indicating that Linear Regression provided a reasonable baseline prediction, though with some variance. Random Forest Regressor was then trained on the same dataset to capture potential non-linear relationships among features. The Random Forest model achieved an R² score of *0.9990093124991511*, an RMSE of *0.33381552142816545*, and an MAE of *0.15803388655700554*, with an accuracy of *0.9971639251276234*, where RMSE & MAE is slightly lower than those of Linear Regression while R² & Accuracy is higher than the Linear Regression model.

Feature importance analysis showed that Jitter(%), and Shimmer were the most influential predictors of disease severity.

1. **Comparative Analysis**

Comparative analysis indicates that Random Forest Regressor outperformed Linear Regression in predicting *total\_UPDRS* scores. The Random Forest model achieved a higher R² and lower error metrics (RMSE and MAE), demonstrating that it captures the non-linear relationships between vocal features and disease severity more effectively. Linear Regression, which assumes a linear relationship among features, performed worse on this dataset due to the underlying non-linear interactions between variables. Overall, vocal features such as Jitter(%) and Shimmer are strong predictors of Parkinson’s severity, and Random Forest Regressor provides a more accurate predictive model for this continuous target.

|  |  |  |
| --- | --- | --- |
| Basis of Difference | Linear Regression | Random Forest |
| R² Score | 0.9098165934299791 | 0.9990093124991511 |
| RMSE | 3.1849418133307656 | 0.33381552142816545 |
| MAE | 2.3678375349783023 | 0.15803388655700554 |
| Approximate Accuracy (within ±2 points) | 0.5484968803176404 | 0.9971639251276234 |
| Actual vs Predicted Total UPDRS |  |  |

Fig 6: Summary Table of both models

# **Conclusion**

Based on the analysis of the Parkinson’s UPDRS dataset, several key conclusions can be drawn. First, vocal features such as Jitter(%), Shimmer, and HNR are strongly associated with the severity of Parkinson’s disease, as evidenced by their moderate to strong correlations with the *total\_UPDRS* score. This indicates that voice measurements can serve as reliable indicators for assessing disease progression.

Second, in predictive modelling, Random Forest Regressor outperformed Linear Regression, achieving a higher R² and lower error metrics (RMSE and MAE). This demonstrates that the relationships between the voice features and disease severity are non-linear, which Random Forest can capture more effectively than Linear Regression. The superior performance of Random Forest is further supported by feature importance analysis, which highlighted motor\_UPDRS, Jitter(%), and Shimmer as the most influential predictors.

Overall, the project confirms that vocal characteristics provide valuable information for predicting Parkinson’s disease severity and that ensemble-based machine learning models like Random Forest are well-suited for capturing complex patterns in such biomedical datasets. These conclusions are justified by the correlation analysis, pairplots, and the comparative predictive performance of the models used.

# **Appendices**

1. References:
   * <https://share.google/IxhkQzdKdjdfncuwf>
   * <https://share.google/tSnG2O2TKQh4GzSlQ>
   * <https://sebastianraschka.com/faq/docs/logistic_regression_linear.html>
   * <https://www.geeksforgeeks.org/machine-learning/the-effects-of-the-depth-and-number-of-trees-in-a-random-forest/>
   * <https://www.geeksforgeeks.org/machine-learning/logistic-regression-vs-random-forest-classifier/>
   * <https://datascience.stackexchange.com/questions/87038/what-is-the-best-machine-learning-algorithm-for-large-noisy-datasets-with-inter>
2. [GitHub](https://github.com/maubanisarkar-cloud/Autumn-Internship-Project-Section-1)
3. [Google Drive](https://drive.google.com/drive/folders/1LYpmAhebcAd8SHby20XRoSJgrzNL11y4?usp=drive_link)