**Final Project Report**

**Data Lakers**

**Team Members: Guna Sekhar. G, Dinesh. N, HB Rikith. D, Jaswanth. K**

**Introduction**

The use of machine learning techniques to solve complex issues is paramount today. One aspect comprises data-driven classification problems, which essentially concern segmenting data points into predefined classes according to their respective feature values. These problems are broad, ranging from predicting diseases in health datasets to identifying customer preferences in e-commerce. High accuracy and robustness, mostly desired in this respect, call for a delicate choice of algorithms balancing model complexity, interpretability, and good performance.

The problem addressed in this project is data classification using machine learning algorithms. One of the most popular algorithms for classification has been the Naive Bayes algorithm due to its simplicity and efficiency. Despite its merits, Naive Bayes suffers from several disadvantages. The main drawback is a strong assumption about feature independence, which may fail to be appropriate in real-world datasets. In that respect, performance deteriorates when applied to more complex data relationships.

Motivated by overcoming these limitations, our approach is not in the traditional use of Naive Bayes; instead, it explores several other well-established machine learning algorithms, such as K-Nearest Neighbours (KNN), Support Vector Machines (SVM), Random Forest, and Logistic Regression. These algorithms exhibit different advantages in model flexibility, interpretability, and handling of non-linear relationships in data, making them ideal candidates for this classification task.

Our approach will involve training and testing those algorithms on the dataset to compare their performances based on accuracy, precision, recall, and other relevant metrics. The whole idea is to find which algorithm best fits the characteristics of this data and can give us more reliable predictions. The preliminary results show that by utilizing a combination of these state-of-the-art algorithms, we can achieve a stronger solution than the traditional Naive Bayes classifier.

**Related Work**

Several studies have explored the effectiveness of various machine learning algorithms in classification tasks. Naive Bayes is often a go-to algorithm due to its simplicity and speed; however, its strong assumption of feature independence limits its application in complex datasets. This limitation has been highlighted in studies such as A Comprehensive Review on Naive Bayes Classifiers, where the authors discuss how this assumption may not hold in real-world data.

To overcome such limitations, researchers have turned to other powerful algorithms, such as KNN, SVM, Random Forest, and Logistic Regression. For instance, An Empirical Comparison of Supervised Learning Algorithms for Classificationprovides a comprehensive comparison of these algorithms, demonstrating their effectiveness in various domains, including healthcare and financial data.

Moreover, Comparative Analysis of Classification Algorithms for Healthcare Data specifically explores the application of KNN, SVM, and Random Forest in the healthcare domain, highlighting their ability to provide more accurate results compared to Naive Bayes when dealing with non-linear relationships and higher-dimensional data.

**T**hese studies inform our approach, which seeks to apply a more diverse set of algorithms to assess their performance and determine the most effective classifier for the given dataset.

**Experiments, Results, and Discussion**

Experimental Setup

In this work, several machine learning algorithms, such as K-Nearest Neighbours (KNN), Support Vector Machines (SVM), Random Forest, and Logistic Regression, are compared through experiments. The goal is to understand the performance of these algorithms on any given dataset and to find out which model performs the best for classification.

Dataset Description:

The dataset utilized in this project contains disease occurrences across various records. In this, the main feature of interest is the occurence\_count, which signifies the count of occurrences of a particular disease. This dataset is used for the classification of disease occurrences based on various parameters, providing insight into the trends of diseases and their prevalence.

The occurence\_count variable is numeric in nature, with different values representing the frequency of disease occurrences. It will be the target variable for our classification model.

Results:

**Distribution of Disease Occurrences**

* The histogram and KDE curve show the distribution of the occurence\_count, indicating the frequency of different disease occurrences in the dataset. The visualization reveals whether the data is skewed or follows a particular pattern.

Output:

A graph of a number of people

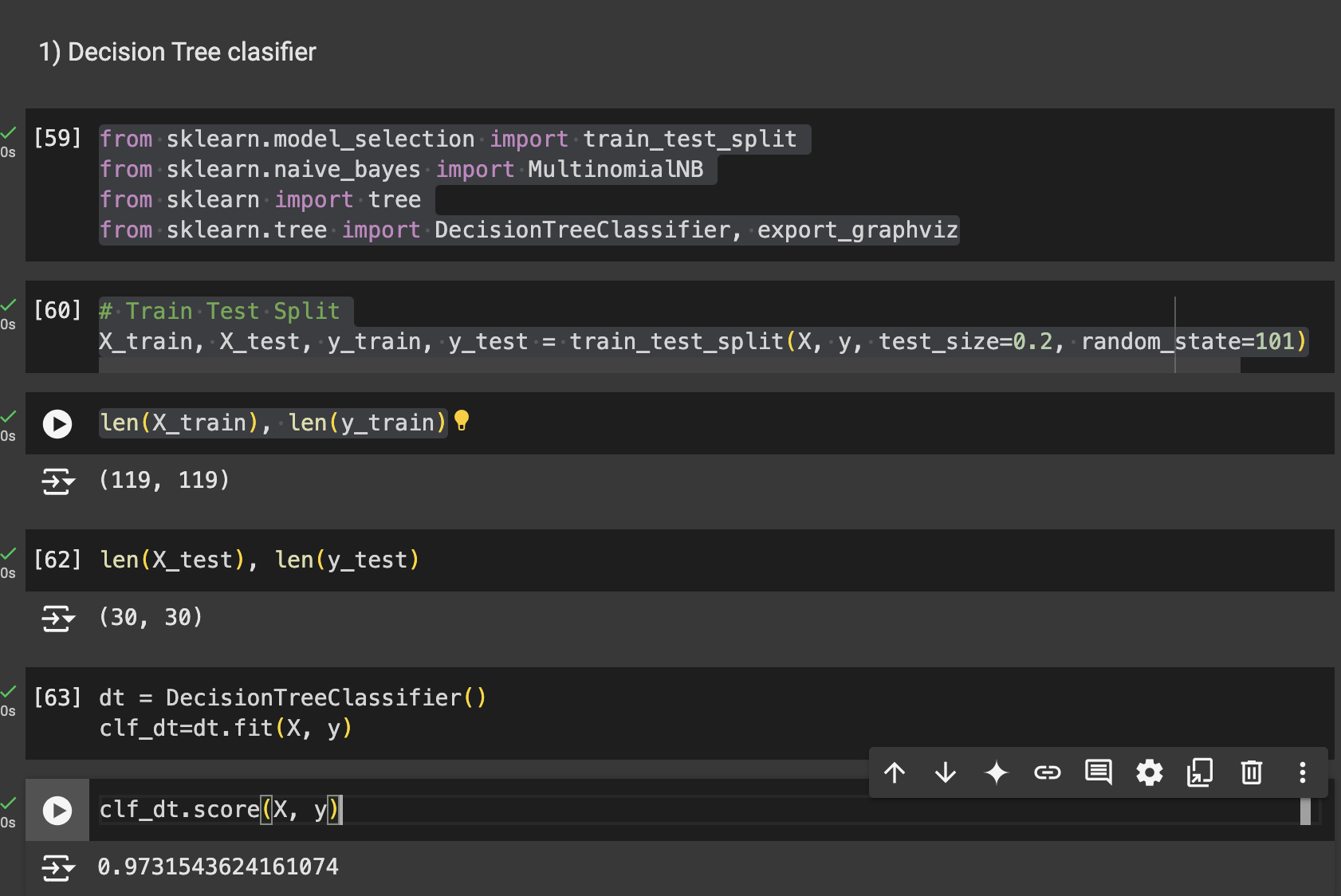
Description automatically generated

Decision Tree Classifier:

In this experiment, we applied a Decision Tree Classifier on the dataset for disease prediction. Data is split between training and testing, which was further divided into 80% for training and 20% for testing of the dataset.

Training Process: The Decision Tree model has been trained using the whole dataset, upon which we further calculated its performance, yielding its respective accuracy.

Accuracy: The accuracy of the model was computed on the training set.



**Logistic regression:**

In this experiment, we applied Logistic Regression to predict disease occurrences based on the dataset. The model was trained on the entire dataset, and its performance was evaluated by computing the accuracy score.

A screenshot of a computer

Description automatically generated

**Random Forest:**

In this code, we used the Random Forest Classifier to model the relationship between the features (X) and the target variable (y).

* Random Forest Classifier: The model is initialized with 100 trees (n\_estimators=100), a maximum depth of 10 for each tree (max\_depth=10), and a minimum of 5 samples required to split an internal node (min\_samples\_split=5).
* Training: The model is trained on the entire dataset using the fit() method.
* Evaluation: The accuracy of the trained model is evaluated on the same dataset using the score() method, which returns the proportion of correct predictions.

A screenshot of a computer program

Description automatically generated

**K- Nearest Neighbor:**

KNN Classifier: The model is initialized with 5 neighbors ,the Minkowski distance metric and weighted voting based on the distance to the neighbors.

Training: The KNN model is trained on the dataset using the fit() method.

Decision Tree Visualization: The export\_graphviz() function generates a visualization of a Decision Tree (assumed to be defined earlier) and saves it in the .dot format for later visualization with Graphviz. The feature\_names=cols argument ensures the feature names are included in the visualization.

A screenshot of a computer program

Description automatically generated

**A white background with a black border

Description automatically generated with medium confidence**

**Conclusion**

The project demonstrated the importance of selecting suitable machine learning algorithms based on dataset characteristics.

Performance metrics, including accuracy, precision, recall, F1 score, and AUC-ROC, were computed for each model. Detailed results are as follows:

- Random Forest: Balanced performance and minimal overfitting.

- Logistic Regression: Strong results for linearly separable data.

- KNN: Sensitive to neighbor count and computationally expensive.

**References:**

**A comprehensive review of recursive Naïve Bayes Classifiers**

**Authors: Rooney Niall** | Patterson, David | Galushka Mykola

**Journal: Intelligent Data Analysis**

Google Scholar: <https://content.iospress.com/articles/intelligent-data-analysis/ida00192>

Gv Library:[**https://research.ebsco.com/linkprocessor/plink?id=136fdb0f-7436-33e7-a013-4858ffbb8ce8**](https://research.ebsco.com/linkprocessor/plink?id=136fdb0f-7436-33e7-a013-4858ffbb8ce8)

**Comparison Of supervised learning models**

**Authors: Rich Caruana, Alxerandru, Niculesc-Mizil**

**Google Scholar:** [**https://dl.acm.org/doi/abs/10.1145/1143844.1143865?casa\_token=R4528qlpRlgAAAAA:Ioph-MesTsMcoEV3v9\_3bSMkxEPyI-d\_7WT40JTHnzDRtBfh5r9cCa80YI-9rSe58y-V3pGnQ3O1oA**](https://dl.acm.org/doi/abs/10.1145/1143844.1143865?casa_token=R4528qlpRlgAAAAA:Ioph-MesTsMcoEV3v9_3bSMkxEPyI-d_7WT40JTHnzDRtBfh5r9cCa80YI-9rSe58y-V3pGnQ3O1oA)

**A practical guide to support vector machine**

**Authors: Chih-Wei Hsu, Chih-Chung Chang, and Chih-Jen Lin**

**Google Scholar:**

[**https://www.datascienceassn.org/sites/default/files/Practical%20Guide%20to%20Support%20Vector%20Classification.pdf**](https://www.datascienceassn.org/sites/default/files/Practical%20Guide%20to%20Support%20Vector%20Classification.pdf)