Vector-Borne Disease Prediction

Rajat Kumar
IIITD
rajat21185@iiitd.ac.in

Rajat Vatwani
IIITD
rajat21186@iiitd.ac.in

Ramit Nag
IIITD
ramit21188@iiitd.ac.in

Yusuf Jamal
IIITD
yusuf21220@iiitd.ac.in

Abstract

This project aims to leverage machine learning for the early diagnosis of vector-borne diseases based on patient symptoms. We will employ a range of algorithms, including Random Forest, Logistic Regression, Decision Trees, Support Vector Machines, and Multi-Layer Perceptron Classifiers, to analyze the data and make predictions.

According to the World Health Organization's 2020 estimates, these diseases contribute to an alarming annual mortality rate of 700,000 individuals, posing a significant global health threat. We believe that machine learning holds the key to addressing this pressing issue, and this belief is what inspired us to pursue this project.
Github Link

1. Motivation and Introduction

Vector-borne diseases represent a critical global health challenge, causing significant morbidity and mortality worldwide. These diseases, transmitted through vectors like mosquitoes, ticks, and fleas, include malaria, dengue, Zika virus, and Lyme disease, among others. They often affect populations in tropical and subtropical regions, where healthcare infrastructure may be limited, making early detection and intervention very challenging.

Traditional diagnostic methods can be time-consuming and costly and often rely on advanced laboratory equipment that is not readily accessible in affected areas. Therefore, there is a pressing need for innovative approaches to improve early detection, enabling quicker response and treatment to prevent severe outcomes.

This project aims to address the challenge of early diagnosis by leveraging machine learning techniques to construct predictive models. Using patient symptomatology as input features, we aim to develop a robust machine-learning model that can assist in the early detection of vector-borne diseases. By identifying patterns in symptoms and correlating them with disease prognosis, we hope to contribute to a more effective and accessible solution to mitigate the global impact of vector-borne diseases.

2. Literature Survey

1. **Disease Prediction using Machine Learning**[1]
The paper presents a system for predicting diseases using machine learning. The authors employed Decision Tree, Random Forest, and Naïve Bayes classifiers to identify diseases based on patient symptoms. They used a dataset of 4,920 patient records, encompassing 41 diseases and 132 symptoms, and selected 95 key symptoms to optimize accuracy and avoid overfitting.

2. **Vector Borne Disease Outbreak Prediction by Machine Learning.** [2]

The paper predicts outbreaks of Chikungunya, Malaria, and Dengue in India using a CNN-based approach, leveraging demographic (positive cases) and meteorological data (temperature, humidity, rainfall) from 2013 to 2017. The CNN-MDOP algorithm classifies regions into low, moderate, or high risk, achieving 88% accuracy by using CNN for feature extraction and a Softmax classifier for prediction.

3. **Application of Machine Learning in Disease Prediction** [3]
The paper applies various machine learning classification algorithms to predict diseases such as heart disease, breast cancer, and diabetes. They utilize the Heart Disease Dataset, Wisconsin Breast Cancer Dataset, and Pima Indians Diabetes Dataset from the UCI repository for their analysis. Some of the key models and techniques used are Logistic Regression, Decision Trees, Random Forest, Support Vector Machine (SVM), and AdaBoost.

3. Dataset

3.1. Description

In this section, we provide an overview of the dataset used in this study and the preprocessing steps applied to prepare the data for modeling. The dataset contains 1,010 records with 66 columns, and it consists of records that capture the symptoms and medical information of patients. It includes the following key attributes:

- **ID**: A unique identifier for each record.
- **Symptoms**: A set of 64 binary features indicating the presence (1) or absence (0) of specific symptoms for each patient. These symptoms serve as input features for the predictive model, and they include various indicators like fever, headache, vomiting, muscle pain, etc.
- **Prognosis**: The target variable that represents the medical outcome or disease diagnosis for each patient. This column has categorical values such as "Lyme disease," "Zika," "Rift Valley fever," and others.
- **Dataset Dimension**: The dataset contains 64 features, leading to a high-dimensional dataset. This highlights the need for dimensionality reduction techniques like PCA to manage and extract essential patterns efficiently.

3.1.1 Initial Observations

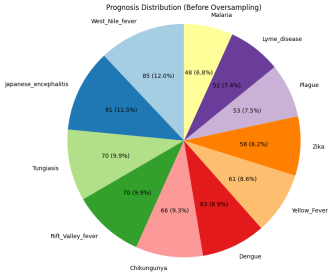

Figure 1. Sample Class Distribution

Upon an initial examination of our dataset, several key observations emerged. The dataset is free of missing values, ensuring data integrity, and it maintains a clean structure without any duplicate records. There are 11 distinct prognosis categories, reflecting the complexity of the conditions we aim to predict. All features are one-hot encoded, indicating the presence or absence of specific symptoms as binary values.

3.1.2 Correlation Matrix

To better understand relationships between features, we computed a correlation matrix for the binary symptom features. It revealed moderate feature collinearity, with some

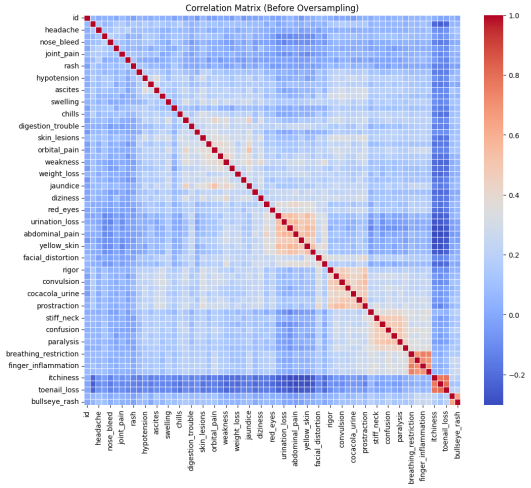

Figure 2. Sample Class Distribution

symptoms frequently co-occurring, indicating relationships between specific symptoms. Strong correlations between certain features suggest interdependencies and potential redundancy.

3.1.3 Correlation Heatmap

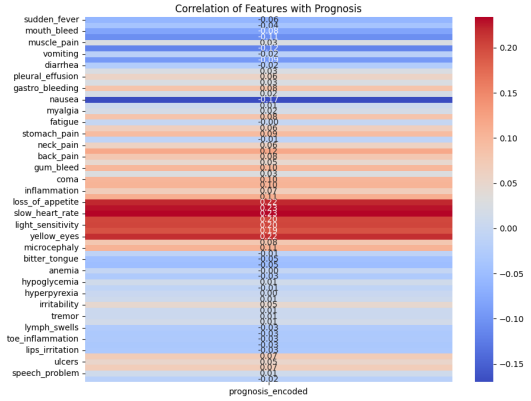

Figure 3. Features vs Prognosis

The heatmap presents the correlation of various features (symptoms) with the prognosis (target variable). Positive correlations are shown in red, while negative correlations are shown in blue. The intensity of the color represents the strength of the correlation. Strongly correlated features like "Slow Heart Rate" and "Light Sensitivity" can be treated as important predictors and retained for model training. Features with weak or insignificant correlations can potentially be dropped to reduce model complexity and noise.

3.2. Data Preprocessing

3.2.1 Label Encoding (Also Handling Missing Values)

The target variable, *prognosis*, which is categorical, was label-encoded into numerical values to facilitate processing by machine learning models. Missing values in the feature columns were handled by imputing the mean of each column, ensuring that all entries in the dataset were complete and ready for training.

3.2.2 Dataset Splitting

The processed dataset was split into *training* and *testing* sets using an 80:20 ratio. This ensures the model is trained on a majority of the data and evaluated on unseen data, providing an accurate measure of performance on future predictions.

3.2.3 Principal Component Analysis (PCA)

PCA was applied for dimensionality reduction while retaining 95% of the dataset's variance. This step helps in speeding up the training process by reducing the number of features, while also minimizing the risk of overfitting by eliminating irrelevant information.

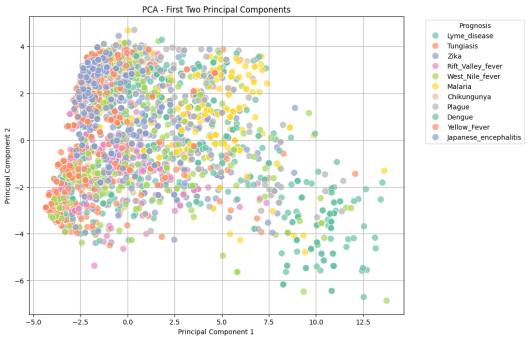

Figure 4. PCA for Dimensionality Reduction

3.2.4 Data Balancing and Using Oversampling (SMOTE)

To address the minor class imbalance in the dataset, *SMOTE* (Synthetic Minority Over-sampling Technique) was used. This technique generates synthetic samples for minority classes, balancing the dataset and ensuring that models don't bias toward the majority class.

3.2.5 Recursive Feature Extraction (RFE)

We performed Recursive Feature Extraction to select the top 30 features. Doing this helped us streamline our dataset by retaining only the most impactful features. This not only boosted the accuracy of all models but also made them more efficient and interpretable. By focusing on the features that matter most, we achieved better performance and gained clearer insights into the factors influencing our target variable.

3.2.6 Data Standardization

All feature columns were standardized using *StandardScaler* to ensure they are on a similar scale. This is particularly important for models that rely on distance calculations, as standardization improves overall performance by preventing features with larger scales from dominating the learning process.

4. Methodology

We developed a disease prediction model using a machine-learning approach. After preprocessing and imputing missing values with means, the target variable was label-encoded. SMOTE addressed class imbalance, and PCA reduced dimensionality while retaining 95% variance. The data was split (80:20) into training and testing sets. Models, including Naive Bayes, Decision Tree, Random Forest, Logistic Regression, and MLP, were trained using a custom bootstrapping function with majority voting for prediction aggregation.

4.1. Model Details

- **Naive Bayes, Decision Trees, Random Forest, Logistic Regression):** Naive Bayes, a probabilistic model based on Bayes' theorem assuming feature independence, is inherently stable, with bootstrapping showing only a 1% accuracy difference. Decision Trees, prone to overfitting, benefited from bootstrapping with an 8% accuracy improvement. Random Forest, leveraging bootstrapped samples and aggregating predictions, improved generalization significantly. Logistic Regression, a linear model, performed well with regularization: L1 improved accuracy by 2%, while L2 reduced overfitting. The best performance was achieved with a regularization strength of $C = 1$. These results were obtained before our midsem deadline.

- **XGBoost Classifier:** XGBoost (Extreme Gradient Boosting) is a powerful machine learning algorithm based on the gradient boosting framework. It is widely used for classification, regression, and ranking tasks due to its high performance and speed. XGBoost builds an ensemble of decision trees in a sequential manner, where each tree tries to correct the errors of the previous trees.

- **Support Vector Machine (SVM):** Support Vector Machine (SVM) is a powerful linear classifier that

finds the optimal hyperplane to separate classes in the feature space. It is robust to overfitting, especially in high-dimensional spaces, and performed well with an accuracy of 0.86 in our experiments.

- **MultiLayer Perceptron (MLP):** Multilayer Perceptron (MLP), a neural network-based model, uses backpropagation for training and can model complex relationships in data. It achieved the highest performance in our experiments, with an accuracy of 0.94 and a ROC-AUC score of 0.99.

5. Results and Analysis

Model	Precision	Recall	F1-score	ROC-AUC
Decision Tree	0.71	0.71	0.70	0.76
Random Forest	0.78	0.78	0.77	0.96
Logistic Reg	0.49	0.50	0.48	0.85
XGBoost	0.78	0.78	0.78	0.97
Naive Bayes	0.42	0.43	0.40	0.81
Multilayer Perceptron	0.94	0.95	0.94	0.99
Support Vector Machine	0.88	0.86	0.86	0.97

Table 1. Performance Metrics of Machine Learning Models

- **Precision**: Support Vector Machine (0.88) and Multilayer Perceptron (0.94) have the highest precision, showing better reliability in predicting true positives. Naive Bayes has the lowest precision (0.42), indicating more false positives.
- **Recall**: Multilayer Perceptron (0.95) excels in recall, identifying the most true positives. Naive Bayes performs the worst (0.43), missing many true positives.
- **F1-score**: Multilayer Perceptron (0.94) achieves the best F1-score, maintaining an excellent balance between precision and recall. Naive Bayes has the lowest F1-score (0.42), reflecting poor precision-recall balance.
- **ROC-AUC**: Multilayer Perceptron (0.99) and Support Vector Machine (0.97) have the highest ROC-AUC, indicating strong class separation. Naive Bayes has the lowest ROC-AUC (0.81), showing weaker class separation.

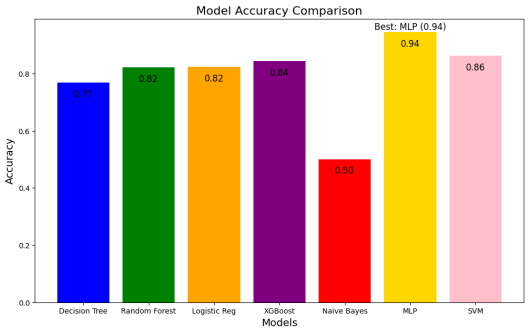

Figure 5. Accuracy for different models

The bar chart compares the accuracy of various models, including Decision Tree, Random Forest, Logistic Regression, XGBoost, Naive Bayes, Multilayer Perceptron (MLP), and Support Vector Machine (SVM). Among these, MLP achieves the highest accuracy at 0.94, making it the best-performing model. SVM follows with an accuracy of 0.86, while XGBoost performs slightly lower at 0.84. Random Forest and Logistic Regression both achieve an accuracy of 0.82, indicating reliable performance but not as strong as the top models. The Decision Tree model has a moderate accuracy of 0.77, and Naive Bayes performs the worst with an accuracy of 0.50, suggesting it is not well-suited for the dataset. This analysis highlights the effectiveness of ensemble methods like XGBoost and advanced models like MLP compared to simpler approaches.

6. Conclusion

- **Best Precision**: The Support Vector Machine (SVM) model achieved high precision (0.88), making it the best choice when minimizing false positives is critical (e.g., avoiding misclassification of non-disease cases as vector-borne diseases).
- **Best Recall**: The Multilayer Perceptron (MLP) model demonstrated the highest recall (0.95), excelling at identifying true positives. This makes it ideal for scenarios where capturing all disease cases is essential.
- **Best F1-Score**: MLP achieved the highest F1-score (0.94), showing its effectiveness in maintaining a balance between precision and recall.
- **Best Overall Performance**: MLP also stood out with the highest ROC-AUC score (0.99), indicating its superior ability to distinguish between disease and non-disease cases. This makes MLP the most reliable model for vector-borne disease prediction.

In summary, while SVM excels in precision, MLP outperforms in recall, F1-score, and ROC-AUC, making it the optimal choice for achieving accurate, balanced, and reliable predictions in vector-borne disease cases. Ensemble models like Random Forest and XGBoost also performed well, highlighting their robustness for similar tasks.

7. Contributions

- Yusuf Jamal: Literature review, Data preprocessing, and visualization, XGboost, Report, SVM, PPT
- Rajat Kumar: Decision Trees, Random Forest, Naive Bayes, Code, MLP, Data preprocessing
- Rajat Vatwani: Logistic Regression, XGBoost, MLP, SVM, Literature Review, Report, PPT
- Ramit Nag: Naive Bayes, Literature review, Logistic Regression, Data preprocessing, PPT

References

[1] S. Grampurohit and C. Sagarnal, "Disease prediction using machine learning algorithms," in *2020 International Conference for Emerging Technology (INCET)*, 2020, pp. 1–7. [Online]. Available: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9154130

[2] S. Raizada, S. Mala, and A. Shankar, "Vector borne disease outbreak prediction by machine learning," in *2020 International Conference on Smart Technologies in Computing, Electrical and Electronics (ICSTCEE)*, 2020, pp. 213–218. [Online]. Available: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9277286

[3] P. S. Kohli and S. Arora, "Application of machine learning in disease prediction," in *2018 4th International Conference on Computing Communication and Automation (ICCCA)*, 2018, pp. 1–4. [Online]. Available: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8777449