A Case of Study for Classifications Algorithm to a Tabular Vector Borne Disease Dataset

Alberto Arath Figueroa Salomon

December 12, 2024

1 Introduction

Vector-borne diseases have become relevant because human activity creates imbalance in ecosystems, and this family of diseases has become a recurring problem in warm climates. These diseases have a common set features, in addition to the fact that they are transmitted bites of a blood-sucking arthropod.
Such common features are the prognosis, where the symptoms between each other could be almost indistinguishable
The purpose of the study is to build a machine learning model that can discern classification patterns for each set of shared symptoms.

2 Data

2.1 Instance composition

Data set is composed by labeled data, 707 rows and 64 features. Training data has a synthetic origin so there is really no need for data preparation which is good as we can focus solely in the application of Machine Learning techniques

Muscle Pain	**Fatigue**	**Weakness**	**Prognosis**
Present	Not Present	Present	Present
Not Present	Not Present	Present	Present

Table 1: Sample training set with a subset of features.

2.2 Predictor Set

Is composed of binary variables specifying if given symptom is present in this prognosis instance it represents the presence or absence of a specific condition, The variable X takes values from the set:

$$X \in \{0, 1\},$$

where:

$$X = \begin{cases} 1 & \text{if the condition is } \textbf{present}, \\ 0 & \text{if the condition is } \textbf{not present}. \end{cases}$$

2.3 Target Variable

The target variable is called Prognosis containing the Vector-borne dissease classfier
The target variable, denoted as Y, is a categorical variable representing the prognosis outcome it can take values from the following finite set of classes:

$$Y \in \{C_1, C_2, \ldots, C_k\},$$

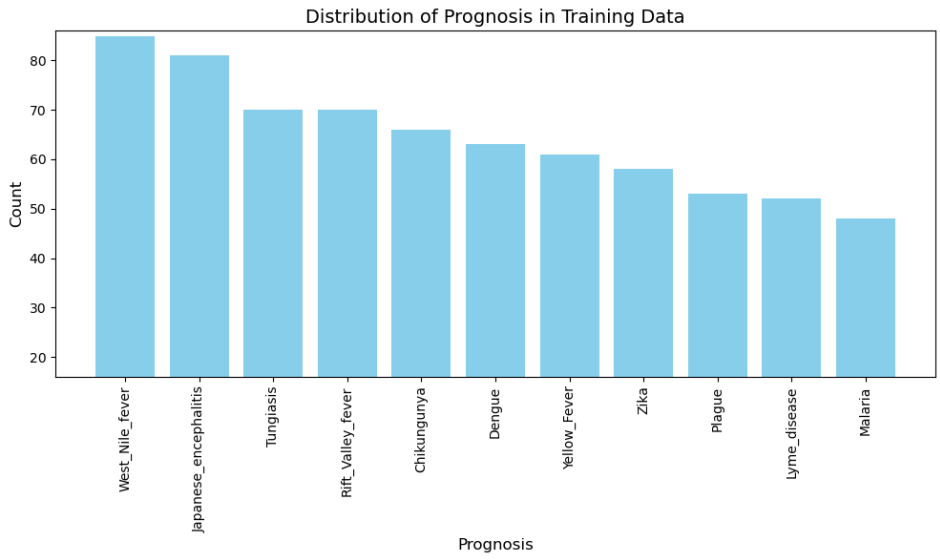

Figure 1: Distribution of Prognosis in Training Data.

where C_i represents the i^{th} class label, for $i = 1, 2, \ldots, k$.
In this study, $k = 11$, and the classes are defined as follows:

- C_1: Dengue,
- C_2: Zika,
- C_3: Malaria.

To allow the application of machine learning algorithms label encoder variables as in following example

$$Y = \begin{cases} 0 & \text{if the class is } C_1 \text{ (Malaria)}, \\ 1 & \text{if the class is } C_2 \text{ (Dengue)}, \\ 2 & \text{if the class is } C_3 \text{ (Zika)}. \end{cases}$$

This encoding To allows the application of machine learning algorithms such as softmax regression, decision trees, or neural networks to predict the likelihood of each class based on the input features.

3 Exploratory Data Analysis (EDA)

3.1 Class distribution

Class distribution shows there is no significan class imbalance, Figure 1 show West Nile Fever with 80 instances and Malaria falls short with 50 as part of study, class imbalance could have and impact but we expect not be signicant. Just for the sake of the study we will apply oversampling techniques to improve model.

3.2 Features correaltion matrix

Correlation matrix shows there is little information between features. So dimensionality reduction techniques are not likely to be effective, in Model section we will place this hypothesis to test.

4 Methodology

In this study, we performed hyperparameter tuning using Grid Search, testing multiple machine learning algorithms to identify the best-performing model. The following algorithms were evaluated:

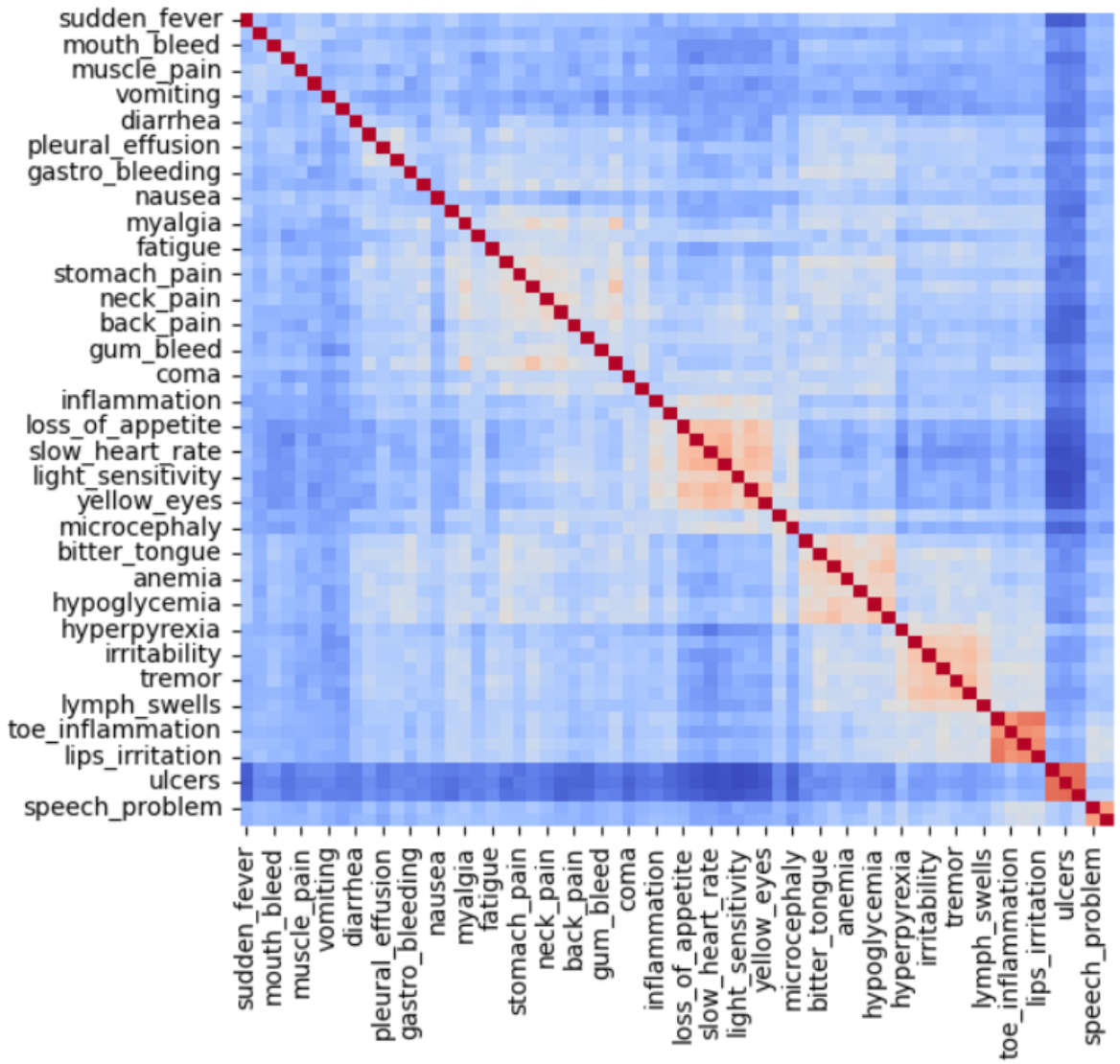

Figure 2: Features Correlation Heat Map. Blue areas indicate low correlation.

4.1 K-Nearest Neighbors (KNN)

The **K-Nearest Neighbors (KNN)** algorithm is a non-parametric, instance-based learning method. It classifies a data point based on the majority class of its k nearest neighbors in the feature space. Hyperparameters tuned include:

- **n_neighbors**: Number of neighbors considered.
- **weights**: Weighting function for neighbors (*uniform* or *distance*).

4.2 Random Forest Classifier

The **Random Forest Classifier** is an ensemble learning method that constructs a collection of decision trees and combines their predictions. It reduces overfitting compared to individual trees. Hyperparameters tuned include:

- **n_estimators**: Number of trees in the forest.
- **max_depth**: Maximum depth of each tree.
- **min_samples_split**: Minimum number of samples required to split an internal node.

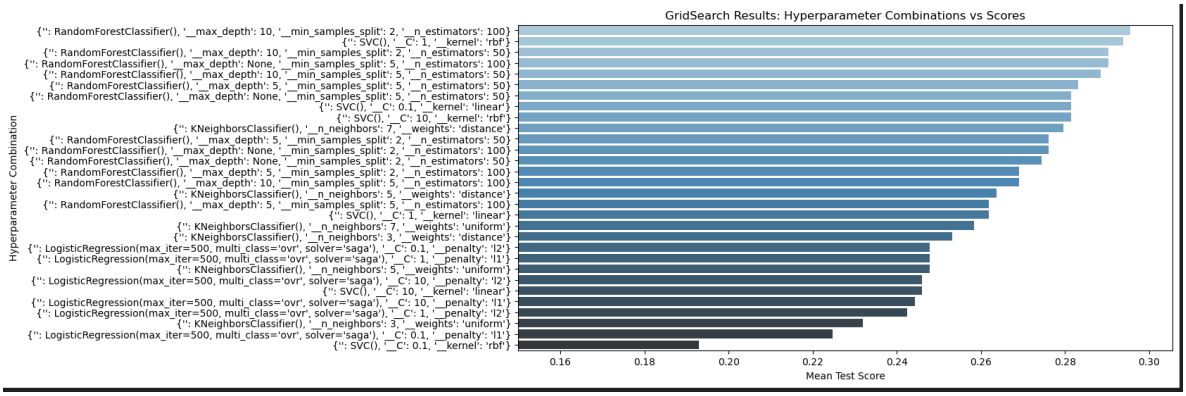

Figure 3: Distribution of Prognosis in Training Data.

4.3 Support Vector Classifier (SVC)

The **Support Vector Classifier (SVC)** is a kernel-based method that finds an optimal hyperplane to separate classes in a high-dimensional space. Hyperparameters tuned include:

- **C**: Regularization parameter.
- **kernel**: Kernel function (*linear*, *rbf*, or *poly*) *rbf* was just included just for benchmark as it is widely used kernel

4.4 Logistic Regression

The **Logistic Regression** model is a linear classifier used for binary and multi-class classification problems. It models the probability of a class using the logistic function. Hyperparameters tuned include:

- **penalty**: Type of regularization (*l1*, *l2*, or *elasticnet*).
- **C**: Inverse of regularization strength.
- **solver**: Optimization algorithm for model fitting.

4.5 Hyperparameter Tuning

The hyperparameter tuning process involved evaluating combinations of the above hyperparameters using cross-validation. The model performance was assessed using the *mean test score*, which is the average score across all cross-validation folds. Models were ranked based on their performance, with the best model achieving the highest mean score.

4.6 Best cross validation score training set

Test set best cross validation Score

Table 2: Classifier Results with Rank and Scores

Classifier	**Mean Test Score**	**Rank**
KNeighborsClassifier	0.295575	1
RandomForestClassifier	0.293805	2
RandomForestClassifier	0.290265	3
KNeighborsClassifier	0.290265	4
RandomForestClassifier	0.288496	5