Machine Learning Methods For Classification

Victor Prieto

Introduction

Machine Learning is a sub-section of artificial intelligence. We use various algorithms to try and predict new outcomes from a given dataset. A program is fed a lot of data about a specific topic, and the algorithm uses the data to try and make its own predictions. There are many uses for machine learning and we for our purposes, we can divide them into two groups: regression and classification. There is also the importance of supervised and unsupervised learning in which one model knows an output, but the other does not.

This paper will explore various machine-learning models that deal with classification. We will explore and analyze the results of these five classification models: Logistic Regression, K-Nearest Neighbor, Support Vector Machine, Decision Tree Classification, Random Forest Classification. For better understanding, lets define what classification models do. The purpose of classification models is to predict categorical labels like "sunny" or yes and no. The most used case of this type of model is for spam detection, image recognition, and other topics that can be labeled.

For the dataset used in this analysis, we used a weather type dataset. A dataset obtained from the website, Kaggle, it mimics weather data for classification tasks. The main goal of the data is to categorize whether the weather would be rainy, sunny, cloudy, or snowy. The dataset has ten variables which include: temperature, humidity, wind speed, precipitation, cloud cover, atmospheric pressure, UV index, season, visibility, and location.

Models

Continuing forward, lets discuss the various models used for this analysis. We start off with Logistic Regression, a statistical method used for multi-class classification. It estimates probabilities between the input features and the likelihood of a binary outcome. It works by calculating a weighted sum of the input features and applies a sigmoid function to squish the results between 0 and 1 - the probability of belonging to the positive class. It minimizes a loss function to improve classification accuracy. It is most effective for linearly separable data but struggles with non-linear data especially so in feature engineering and transformation.

K-Nearest Neighbor classifies new samples based on the majority class among the k closest neighbors in its training data. For all predictions, the distance from the target point to all training samples is calculated using Euclidean distance - usually selecting the k-nearest points. The majority determines the prediction. It improves on capturing non-linear patterns, but struggles with large datasets.

Support Vector Classification is similar in method to its regression counterpart. It finds the optimal hyperplane to separate classes with maximum distance between the nearest points of each class. As it comprises of vectors and an inherent 3D space, it can handle non-linear data by using kernel functions to map the data to higher dimensional spaces. Its goal is to maximize the margin between classes by adjusting the position of the hyperplane. For higher dimensions, radial basis function and the polynomial kernel are used to map data to higher dimensions for linear separation. It greatly improves for high-dimensional spaces and non-linear boundaries, but is computationally expensive and is highly sensitive to parameter tuning.

Decision Tree Classification wok like flowcharts. Each internal node represents a decision based on any given feature. A branch represents an outcome and the leaves represent a class label. Data is recursively split based on its feature values. The algorithm selects the feature and threshold that results in the best split after utilizing criteria like mean squared error. It goes on splitting until it reaches the stopping criteria or max depth. Decision trees are easy to interpret and require minimal preprocessing but are prone to overfitting and are highly sensitive to small changes in the data.

Finally, Random Forest Classification is an ensemble method that utilizes multiple decision trees using random samples. It averages the predictions to improve accuracy and reduce overfitting. Each tree learns on a random subset of features and data points. Each tree is trained on a sample of the data using a random subset of features for each split. A divers set of weak learners produce averages that are combined for majority voting. Random Forest Classification reduces overfitting and has high accuracy, it also works well with large datasets but struggles in computation.

Results

The following section is the recorded values of the accuracy of the five different models used.

The accuracy score of Logistic Regression is: 0.8612121212121212
The accuracy score of K-Nearest Neighbor is: 0.9042424242424243
The accuracy score of Support Vector Classification is: 0.8824242424242424
The accuracy score of Decision Tree Classifier is: 0.9124242424242425
The accuracy score of Random Forest Classifier is: 0.9227272727272727

Conclusion

We can see from the accuracy score that the Random Forest Classifier produced the best outcome. As stated earlier, it is known that the Random Forest model usually produces results with high accuracy. All models performed well with relatively high accuracy. All models will produce fairly accurate results, there is no model that has notable out-performance. The dataset was fairly large in size and had multiple features, a good fit for the Random Forest model.

Confusion Matrices

Here we have a visualization of the confusion matrix for each classification model.

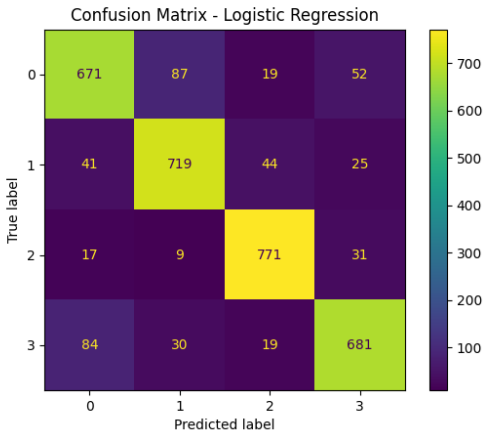

Figure 1: Confusion Matrix for Logistic Regression

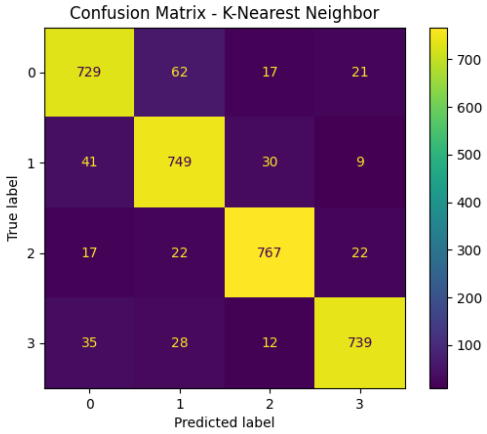

Figure 2: Confusion Matrix for K-Nearest Neighbor

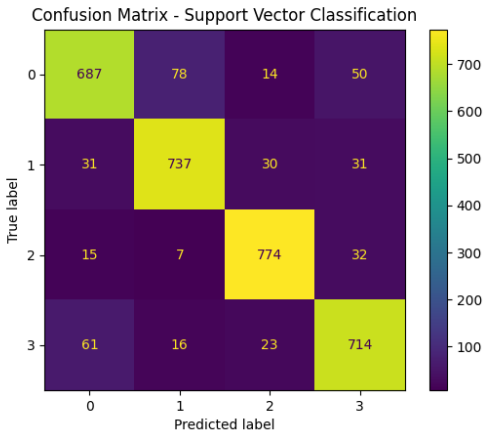

Figure 3: Confusion Matrix for Support Vector Classification

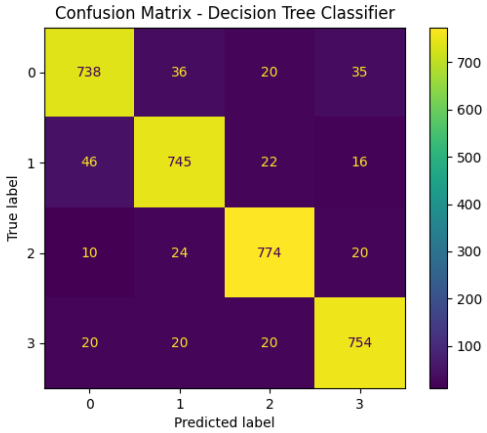

Figure 4: Confusion Matrix for Decision Tree Classification

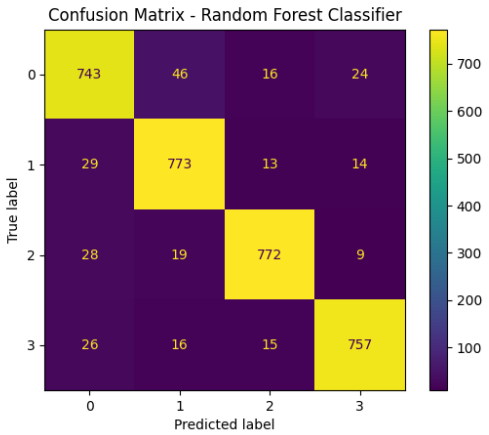

Figure 5: Confusion Matrix for Random Forest Classification