Group Summary – Assignment 3

Omer Avisror ,Tal Vazana ,Gleb Tcivie :Authors

Table of Contents

Summary and Comparison of Classification Algorithms

Introduction

In our project, we explored various classification algorithms to predict diseases based on blood sample parameters. The data, derived from the Multiple Disease Prediction dataset on Kaggle, includes parameters such as Cholesterol, Hemoglobin, White Blood Cells, and several others, which are scaled between 0 and 1 for normalization purposes.

Decision Trees

Conclusion:

The Decision Tree classifier provided a good balance between accuracy and interpretability. It performed not so well with discrete output, but we got around 40% accuracy on the testing dataset.

Ease of Use:

Setting up and training the Decision Tree was straightforward. Scikit-learn's **DecisionTreeClassifier** is well-documented, making it easy to implement and tweak parameters like **max_depth** and **min_samples_split**.

Overall Impression:

Decision Trees are very handy for quick prototyping and when a clear explanation of the decision process is required. However, they can be prone to overfitting, especially with a dataset having many features.

K-Nearest Neighbors (K-NN)

Conclusion:

K-NN performed really badly, with only 18.5% accuracy. Its performance heavily depended on the choice of 'K' and the distance metric used.

Ease of Use:

K-NN is easy to implement and understand. The primary challenge is in choosing the right value of K and the distance metric, which can significantly affect the model's performance. Also, in the dataset, we had trouble converting the parameters to points in a 2D plane to classify them using the K-NN algorithm.

Overall Impression:

K-NN is effective for datasets where similar cases tend to cluster together. However, its requirement for distance computation for each query instance makes it computationally expensive for large datasets.

Logistic Regression

Conclusion:

Logistic Regression provided a robust baseline model. It was particularly effective for binary classification problems and performed well with a logistic loss function.

Ease of Use:

Implementing Logistic Regression using libraries like scikit-learn is straightforward. The model is highly interpretable, with coefficients providing insights into the relevance of each feature.

Overall Impression:

It's a powerful algorithm for binary classification problems, though it assumes linearity between the dependent and independent variables, which can be a limitation for complex relationships.

Support Vector Machine (SVM)

Conclusion:

SVM seemed effective enough to predict some targets, but also missing others.
The most logical conclusion in that case is that the training overfitted for certain cases.
After trying out different combinations of options, the best results yielded a prediction accuracy of over 50%.

Ease of Use:

The setup for SVM is more complex compared to other models, especially in choosing and tuning the kernel type (linear, RBF, polynomial).

Overall Impression:

From our impression, SVM is susceptible to overfitting, and therefore might only fit certain targets or certain nature of data, but not what we used.

Comparison

All models provided unique strengths and weaknesses:

- **Decision Trees** were the easiest to interpret.
- **K-NN** was simple but computationally expensive with larger datasets.
- **Logistic Regression** offered a strong baseline with good interpretability but limited to linear relationships.
- **SVM** excelled in specific scenarios where classes are well separable but require careful tuning and computational resources.

In conclusion, our exploration of these models provided valuable insights into their applicability depending on the nature of the dataset and the specific requirements of the disease prediction task. Depending on the priority of accuracy versus interpretability or computational efficiency, one might choose differently. Logistic regression and decision trees might be preferable for large datasets or where computational resources are a constraint. For higher accuracy and complexity, SVM could be the best choice. Finally, we can clearly see that without any preprocessing made the results are quite bad and lack accuracy, like we saw in the K-NN classification and others.