Analysis and Report on Model Performance

Beltrán Hilda, A01251916, Tecnológico de Monterrey Campus Querétaro

Abstract – This report presents the implementation and analysis of the performance of a Random Forest Classifier. As well as modifications to improve the model's performance with some hyperparameters.

I. INTRODUCTION

Heart attacks have been one of the biggest causes of death worldwide, so it is vitally important to predict who is most likely to suffer from this condition. This condition occurs when blood flow to the heart is affected, that is, when a clot is blocking the passage of blood to the heart.

Technology has been constantly growing and there are more and more tools to facilitate the early detection of certain diseases. As is the case with machine learning models, which can help us predict future events based on previous data to train models.

These machine learning algorithms, in conjunction with historical data on patients who have suffered, and patients who have not suffered from this condition, can achieve a prediction algorithm. Being a very useful tool to prevent being in this situation, reduce the mortality rate due to this condition and improve people's lifestyle.

II. DATASET

The dataset used for this analysis is from the Kaggle platform, and is made up of around 300 documented cases, where it is classified, depending on the entries, as slightly or highly likely to suffer a heart attack. The dataset contains the following data to generate predictions:

1. **Age**: Age of the patient.
2. **Sex**: Sex of the patient.
3. **Exang**: Exercise induced angina, 1 for yes, 0 for no.
4. **Ca**: Number of major vessels, 0 to 3.
5. **Cp**: Chest pain.
   - 1: typical angina
   - 2: atypical angina
   - 3: non-anginal pain
   - 4: asymptomatic
6. **Trtbps**: Resting blood pressure, in mm/Hg.
7. **Chol**: Cholesterol in mg/dl.
8. **Fbs**: If fasting blood sugar is greater than 120 mg/dl.
9. **Rest_ecg**: Resting electrocardiographic result.
   - 0: Normal
   - 1: Having ST-T wave abnormality
   - 2: Showing probable or definite left ventricular hypertrophy
10. **Thalach**: Maximum heart rate achieved.

11. **Target**: Chance of heart attack.
    - 0: Less chance
    - 1: More chance

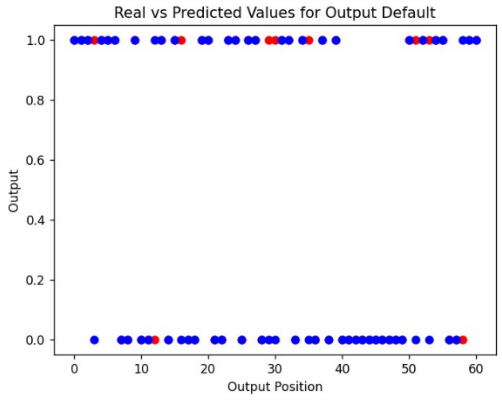
III. MODEL PROPOSAL

The model consists of a Random Forest Classifier, which is made up of a set of decision trees. Each one oversees certain features to obtain decision trees with different knowledge. The algorithm uses this combination of decision trees to obtain a better prediction, in addition to helping reduce overfitting in the model.

This model uses certain hyperparameters to obtain a better learning adjustment, for the first implementation the default parameters are used. Only the number of trees to be used is specified, although it is defined as the default value, which is 100 trees.

Before testing the model, the data is separated between train and test with the train_test_split function of Scikit-Learn. In this way we obtain training data of 80% of the whole dataset, and testing data of 20%.

This way we ensure we have enough data to train the model, but also enough data to verify that the model has learned correctly. When training and testing the model, we observed the following behavior when learning.



The metrics used to analyze the behavior of the model were the Accuracy Score and the Mean Squared Error. In this case, for the model with the default values for all the hyperparameters, we got the next scores and errors.

The Accuracy Score for the training data is 1.0 and the Mean Squared Error is 0.0. Even though this would be wonderful, we need to evaluate the scores and errors from the testing data. For the testing data in the model, the Accuracy Score is 0.8525 and the Mean Squared Error is 0.1475. The difference between the training and testing Accuracy Score means there's overfitting in the model's learning, this is caused by the machine learning algorithm memorizing the data set and not finding patterns to predict new data. We can say the bias in the model is medium, since the Mean Squared Error is not that big in comparison with our data. The variance among the algorithm is low since our predicted and real data stay close.

It can be seen in the graph that the predicted and the real outputs are not so similar. We can explain the Mean Squared Error obtained before, since we can see the

difference for some points, even though the majority are close.

This is seen when the algorithm has excellent accuracy in the training data, but a significantly lower accuracy for the testing data. The overfitted model is not useful for predictions in this context, since we'll be getting predicted outputs that can be false positives and false negatives that could be harmful for a patient diagnostic.

IV. MODEL IMPROVEMENT

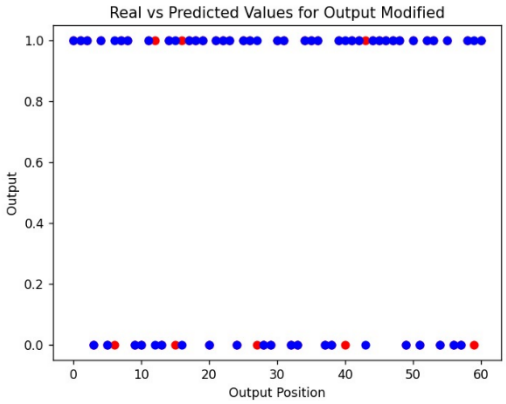
Because the previous model is not optimal for making predictions about the possibility of suffering from a heart attack, we must modify certain hyperparameters. For this, the most common hyperparameters were used to seek to improve the performance of the model. Some of the parameters that were adjusted were the following:

1. n_estimators: The number of trees generated for the model, the default, 100 trees, were used.
2. max_features: How the features are splitted for the trees, the logarithm base 2 is used for this model.
3. max_depth: The longest distance from the root node and the last leaf node, 4 is being used.
4. random_state: Used to reduce overfitting and control randomness among runs of the algorithm.
5. max_leaf_nodes: Controls the number of leaves in each tree, it's set to 10.
6. n_jobs: Set to -1 means it's using all the processors to run jobs in parallel.

After these modifications in the hyperparameters, the model's bias was significantly reduced, and the variance remained low. Being a best fit for the problematic looking to be resolved.

The following behavior can be observed in the graph after having applied the changes to the hyperparameters. You can notice a greater coincidence in the points, which means a better fit of the model to the data.

Same as in last model, the distribution between train and test data was 80% and 20% respectively. The only difference between this model and the last one is that the features were scaled, using the Standard Scaler from Scikit-Learn, before training the model.



As well as in the last model, the metrics used to analyze the behavior of the model were the Accuracy Score and the Mean Squared Error. For the model with the modified

hyperparameters we obtained the next scores and errors.

The Accuracy Score for the training data is 0.8926 with a Mean Squared Error of 0.1074, while the testing data Accuracy Score is 0.8688 with a Mean Squared Error of 0.1311. Meaning we got rid of the overfitted model, since the difference between the training and testing Accuracy Score is not that big. The model can still be improved to achieve a greater Accuracy Score for the data, but 89.26% of Accuracy in training and 86.88% for the testing data is considered good. We can say the bias in this model is low, since there's no great difference between Accuracy's, from the training and testing data. And the variance of the model remained low since the predictions for the output variable are not far from each other.

After these modifications, we can say our model is fitted, based on the Accuracy Score, the Mean Squared Error, the bias, and the variance found in the model predictions.

V. CONCLUSIONS

After modifying the hyperparameters, the performance of the model was significantly better, since the model is not memorizing data, noise and it's generalizing. Concluding that the second model is a better implementation for a classification model, working best for the kind of classification problem we want to solve in this case. Even though the results were good, it's not enough to predict the probability of suffering from a heart attack or not. That's why the model must be complemented with other hyperparameters, data cleaning and transformation to use the features more relevant to the situation.

VI. DATASET RETRIEVAL

The dataset was taken from Kaggle's Heart Attack Analysis & Prediction Dataset, and it can be found in this link:

https://www.kaggle.com/datasets/rashikrahmanpritom/heart-attack-analysis-prediction-dataset

VII. REFERENCES

[1] Heart attack Analysis & Prediction Dataset. (2021, 22 marzo). Kaggle. https://www.kaggle.com/datasets/rashikrahmanpritom/heart-attack-analysis-prediction-dataset

[2] Sklearn.ensemble.RandomForestClassifier. (s. f.). scikit-learn. https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html