Classification Algorithm for Planets discovered by Kepler

Aim

Given a list of planets discovered by KEPLER. Create an ML algorithm to classify the planets as Candidate/False positive/Confirmed etc based on the column “koi_disposition”.

Data Preprocessing Steps

For the given dataset three data pre-processing steps were implemented before feeding it to the model:

1. Data Formatting- First of all the unnecessary rows and columns were removed from the dataset like the rows containing details about the parameters, irrelevant columns, etc.
2. Data Cleaning- All the rows having missing values for any of the given parameters were deleted.
3. Data Normalization- Since the range of value for the given varied by a large margin so data normalization was applied. In this case, Min-Max Scaling was used.

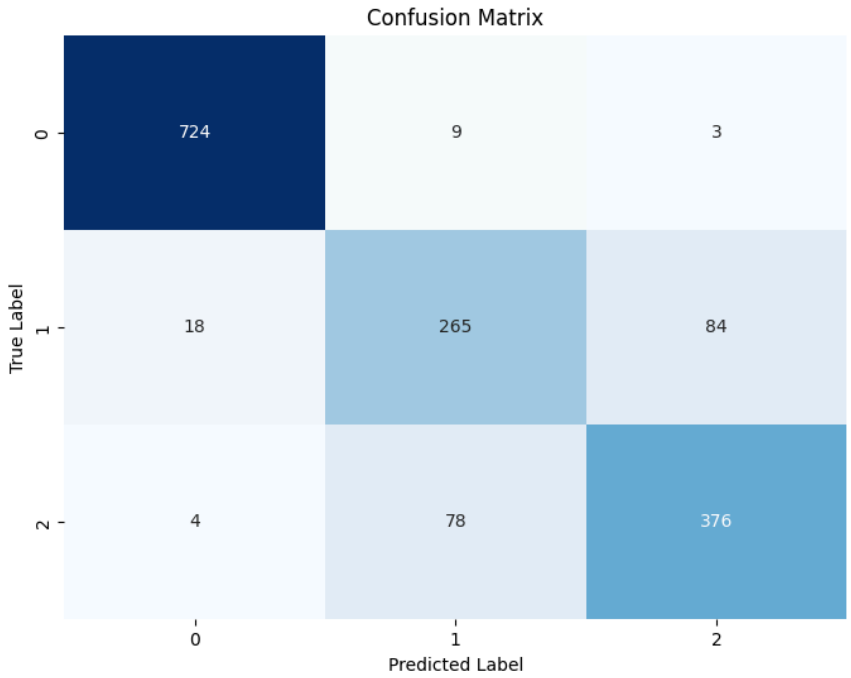
Models Used

For this task three different models were selected-

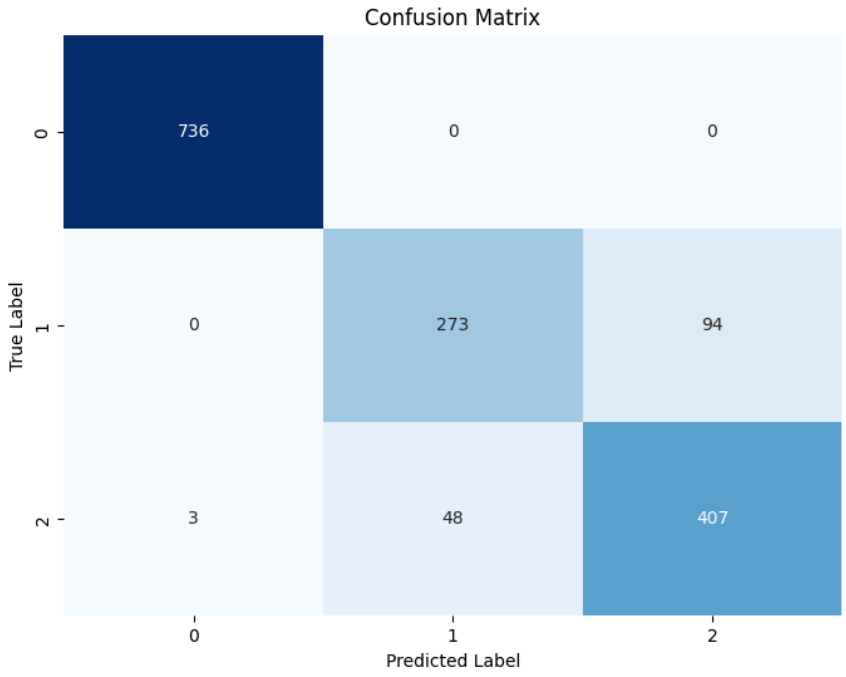
1. ANN(Artificial Neural Network)- An ANN model with a total of 7 dense layers(5 hidden layers) was used. For the input layer, 44 neurons are used, for hidden layers, 50 neurons are used, and since there are three classes to classify, the output layer has 3 neurons with softmax function as activation function. For other layers, a hyperbolic tangent activation function is used. For the optimizer, the Adam optimization algorithm is used with a learning rate of 0.005 and loss function as Sparse Categorical Crossentropy function. The model was trained for 50 epochs and with a batch size of 200 data points with a test-train split of 20%.
2. Random Forest Model- For this case, a model with the number of estimators is 100, the maximum number of features considered is 25 and the criterion for selecting the root node is set to be entropy.
3. XG Boost- We used an XG Boost model with the number of estimators as 1000 and the learning rate as 0.01.

Results

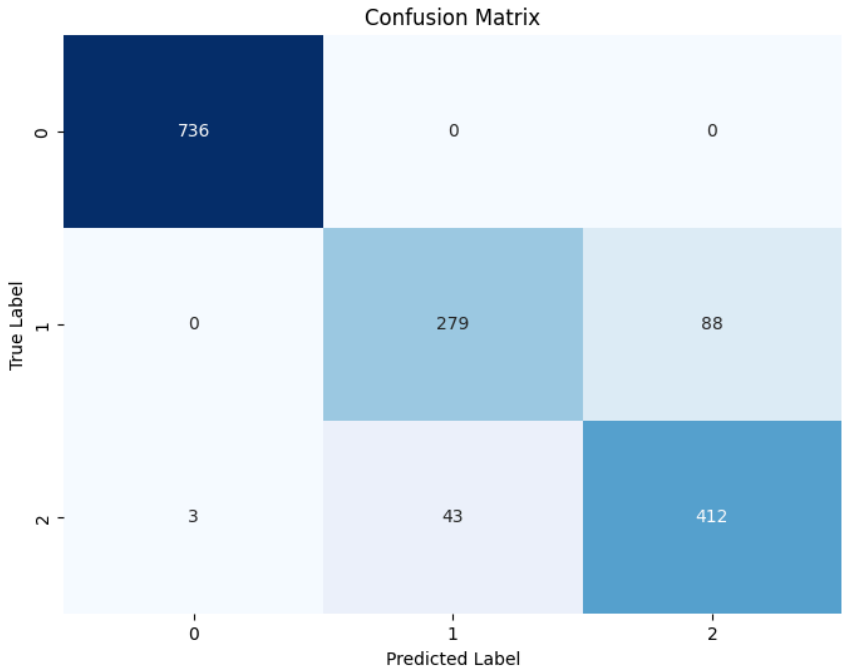
1. ANN Classifier-
   Accuracy: 0.874; Precision: 0.873; Recall: 0.874; F1-Score:0.873



2. Random Forest Classifier-
   Accuracy: 0.907; Precision: 0.908; Recall: 0.907; F1-Score:0.906



3. XG Boost Model-
   Accuracy: 0.914; Precision: 0.915; Recall: 0.914; F1-Score:0.913



Questions related to problem 2.

Q1. Why did you choose the particular algorithm?

Ans- For the kind of dataset that was given, i.e. the dataset with a lot of features involved in the classification and the task being a three-class classification problem was expected to have a very complex relation between the parameters and the classes. So for these kinds of problem statements, we can't use curve-fitting algorithms like logistic regression or binary classifiers like SVC. For these kinds of datasets generally, deep learning algorithms and decision tree-based algorithms are more effective. Moreover among the deep learning algorithms, CNN was not considered because the number of input parameters was not extremely large, so even both CNN and ANN will perform similarly in this case. The RNN-based models were not considered because the results for each classification are independent of previous results.

Q2. What are the different tuning methods used for the algorithm?

Ans- Some of the commonly used Tuning methods used for the algorithms are-

1. Grid Search- Here we specify a certain number of values of each hyperparameter to be tuned and the algorithm tries different combinations of these values and uses the one that gives the best results.
2. Bayesian Optimization- Bayesian optimization is an iterative optimization technique that models the validation loss as a probabilistic surrogate function.

3. Model-based Optimization- Builds a surrogate model of the objective function and iteratively updates the model to explore the hyperparameter.
4. Manual Search- The user tunes the hyperparameters manually according to his/her understanding of the dataset and the model.

In this case, grid search and manual search methods are used.

Q3. Did you consider any other choice of algorithm? Why or why not?

Ans- Explained in Q1.

Q4. What is accuracy?

Ans- Accuracy for different models is mentioned in the results section. The best results were given by the XG Boost model i.e. 91.4% accuracy.

Q5. What are the different types of metrics that can be used to evaluate the model?

Ans- For classification tasks like this generally evaluation metrics like accuracy, precision, recall value, f1-score, etc are used as in this case. It is always a better practice to use the confusion matrix because it gives a pretty good visualization of how the model is performing for different distinctions.