**1.Why did you choose the particular algorithm?**

I chose the RandomForestClassifier algorithm because it's known for its ability to handle both classification tasks and datasets with a large number of features. Random forests are robust against overfitting and work well with both numerical and categorical data, making them suitable for the Kepler dataset, which includes a mix of numeric and categorical features.

**2.What are the different tuning methods used for the algorithm?**[**¶**](http://localhost:8888/notebooks/Desktop/now/Web%20scrapping%20and%20ML/Kepler_dataset%20classification.ipynb#2.What-are-the-different-tuning-methods-used-for-the-algorithm?)

For tuning the RandomForestClassifier, I mainly focused on adjusting the hyperparameters that control the behavior of the model. Some common tuning methods include:

Grid Search: Searching exhaustively over a specified hyperparameter grid to find the combination that yields the best performance.

Random Search: Sampling hyperparameter combinations randomly from specified distributions, which can be more efficient than grid search.

Cross-Validation: Using techniques like k-fold cross-validation to evaluate the model's performance across different subsets of the training data, helping to assess generalization performance and avoid overfitting.

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

Yes, I did consider other algorithms before choosing RandomForestClassifier. Some alternatives include:

Support Vector Machines (SVM): Suitable for binary classification tasks and effective in high-dimensional spaces.

Gradient Boosting: Ensemble learning technique that builds multiple weak learners sequentially, often achieving high accuracy.

Neural Networks: Deep learning models capable of learning complex patterns in data, although they require more computational resources and data preprocessing.

However, I opted for RandomForestClassifier due to its simplicity, versatility, and generally good performance across a wide range of datasets.

I also tried it with decsion tree and it was able to train and test the data sets well as i achieved a higher percent score which only took place due to dropping of a few columns and replacemnet with NULL.But working on a Rando forest would be way accurate for unseen data.

**4.What is the accuracy?**

Random forest- 98.43%

Decison Tree- 99.90%

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

Precision, Recall, and F1 Score: Useful for evaluating the balance between false positives and false negatives.

Confusion Matrix: Provides a more detailed breakdown of the model's predictions across different classes.

ROC Curve and AUC Score

All these help make informed decisions about its deployment and further optimization.