COMP5310 Principles of Data Science

Assignment - Project Stage 2

Student Name: Ruijue Zou
Student ID: 500709979
Unikey: rzou3444

Setup

The problem of this project that needs to be solved was defined in the previous Stage 1, which is, finding a proper machine learning model to help astronomers identifying which candidate they detected is represented as a true pulsar. In Stage 2, the **research question** is to identify whether a certain classification model could have better performance than other models, such as Decision Tree and Naive Bayes.

The hypothesis for this project is set as the following:
• **Null hypothesis (H0):** The performance of classification models are similar, and there is no significant difference between them.
• **Alternative hypothesis (H1):** The performance of classification models are not similar.

In terms of solving the research question and making sure quantify reliability, the accuracy and f1-score will be used to measure the performance of each classification model, and the Kruskal-Wallis H-test will also be applied to test the significance.

The dataset is the same as stage one, which is downloaded from UCI website (https://archive.ics.uci.edu/ml/datasets/HTRU2). The statistics summary of the dataset has shown in Table 1.

	count	mean	std	min	25%	50%	75%	max
Mean_IP	17898.0	111.079968	25.652935	5.812500	100.929688	115.078125	127.085938	192.617188
SD_IP	17898.0	46.549532	6.843189	24.772042	42.376018	46.947479	51.023202	98.778911
EK_IP	17898.0	0.477857	1.064040	-1.876011	0.027098	0.223240	0.473325	8.069522
S_IP	17898.0	1.770279	6.167913	-1.791886	-0.188572	0.198710	0.927783	68.101622
Mean_C	17898.0	12.614400	29.472897	0.213211	1.923077	2.801839	5.464256	223.392140
SD_C	17898.0	26.326515	19.470572	7.370432	14.437332	18.461316	28.428104	110.642211
EK_C	17898.0	8.303556	4.506092	-3.139270	5.781506	8.433515	10.702959	34.539844
S_C	17898.0	104.857709	106.514540	-1.976976	34.960504	83.064556	139.309331	1191.000837
Class	17898.0	0.091574	0.288432	0.000000	0.000000	0.000000	0.000000	1.000000

Table 1: Original Dataset Statistics Summary

Approach

1. Pre-processing

• *Check missing values, outliers and duplicated records*

Before building the classifier models, first of all, it is necessary to make sure there are no missing values in this dataset. After checking, there are no missing values in the dataset. Then, to check the outliers, the Interquartile Range (IQR), the number of outliers and the percentage of outliers are calculated. The results show that the column “Mean_C” (Mean of the DM-SNR curve) has 2927 outliers, 16.4% of value in this column is outliers, meanwhile, the column “SD_C” (Standard deviation of the DM-SNR curve) has 2346 outliers, which stands for 13%. However, considering these data points might have special meanings corresponding to the true pulsars, these outliers will not be removed. Also, we checked there are no duplicated records in the dataset.

• *PCA*

The Principle Components Analysis (PCA) has been applied to this dataset. The results of PCA has been shown in Figure 1. It is clear that the first three components can explain more than 98% of the data. Since this dataset only has 8 attributes, the number of attributes is not too much. Therefore, this project will use all the attributes.

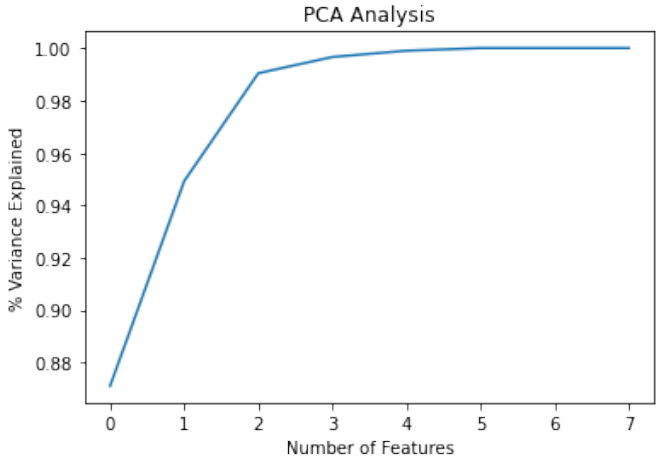


Figure 1: PCA Analysis

• *Normalisation*

According to Table 1, it is obvious that the scale of each attribute is not the same, and there are huge differences. For instance, the range of "EK_IP" is from -1.88 to 8.07 whereas the range of "S_C" is from -1.98 to 1191. Because of some of the classification algorithms, such as KNN, require to calculate the distance between data points, the same scale of data could make the calculation efficient. Therefore, normalisation will be implemented to the dataset to make sure the scale is from 0 to 1.

2. Model selection

Since this project is a classification problem, and there are several popular classification algorithms, such as Super Vector Machine (SVM), K-nearest Neighbour (KNN), etc. For this project, five classification algorithms have been chosen to implement, which are SVM, KNN, Logistic Regression, Random Forest and NaiveBayes. And the Random Forest has treated as the benchmark model for comparison.

3. Classification models implementation & Parameters tuning

For the classification modelling and parameters tuning, the data has been split into three subsets for training, validation and test respectively. Then the training set has 10022 samples, validation set has 4296 samples and the test set has 3580 samples. Because of the number of true and false labels are highly imbalanced, we have also checked the percentage of true labels in each subset to make sure there are around 10% true pulsars in each subset.

To train the classifiers, the training set and validation set has been introduced to each classifier. Also, the GridSearchCV has been applied to find the best parameters for each classifier. The best parameters, the accuracy and F1-score of the validation set are shown in table 2. Because the dataset has an uneven class distribution, the F1-score would be more reliable than accuracy. The complete classification report of each classifier can be found in the Appendix.

Classifier	**Best Parameters**	**Accuracy** (Validation set)	**F1-score** (Validation set)
Random Forest (Benchmark)	criterion: gini, max_depth: 30, max_features: log2, n_estimators: 30	0.98	0.87
Logistic Regression	max_iter: 25, multi_class: multinomial, penalty: l1, solver: saga	0.97	0.83
KNN	algorithm: auto, n_neighbors: 7, weights: distance	0.98	0.87
SVM	C: 100, loss: squared_hinge, penalty: l2	0.98	0.88
Naive Bayes	-	0.95	0.73

Table 2: Classifiers with the best parameters and results

4. Apply Kruskal-Wallis H-test

To determine whether the performances are the same or have significant differences, we applied classifiers to the test set and do cross-validation to get 10 F1-scores and accuracy respectively. And the Kruskal-Wallis H-test has been applied on different pairs of classifiers with the benchmark. In this case, the alpha has been set to 0.05. The H-test results have shown as the following two tables.

Classifier	**Average F1-score** (10 folds)	**P-value** (Compared to Benchmark)	**Reject H0?** (α = 0.05)
Random Forest (Benchmark)	0.9785	-	-
Logistic Regression	0.9690	0.01087	Yes
KNN	0.9765	0.42278	No
SVM	0.9385	0.51639	No
Naive Bayes	0.9469	0.00014	Yes

Table 3: H-test Result (F1-score)

Classifier	**Average Accuracy** (10 folds)	**P-value** (Compared to Benchmark)	**Reject H0?** (α = 0.05)
Random Forest (Benchmark)	0.9796	-	-
Logistic Regression	0.9648	0.00109	Yes
KNN	0.9765	0.32172	No
SVM	0.9726	0.15896	No
Naive Bayes	0.9469	0.00015	Yes

Table 4: H-test Result (Accuracy)

Results

According to Table 2, which contains the results of the validation set, the Random Forest, KNN and SVM classifiers got the same accuracy 98%, and the accuracy of Logistic Regression and Naive Bayes model is slightly lower. However, the F1-score shows that the SVM got the best performance. As discussed before, because of the uneven class distribution, F1-score is more reliable than accuracy. Therefore, for the validation set, SVM is the best classifier. Nevertheless, if the classifiers applied to the test set, the results could be different.

As the results are shown in Table 3, which is based on the average F1-score of the test set, the result of H-test suggested that, compared to the benchmark Random Forest model, the Logistic Regression and Naive Bayes have significant differences with benchmark Random Forest, whereas the KNN and SVM are not different from the benchmark model. And Table 4 got the same result. According to the average F1-score and average accuracy, although the Random Forest, KNN and SVM got similar results, it is easy to identify that the performance of the Random Forest classifier is the best among all classifiers.

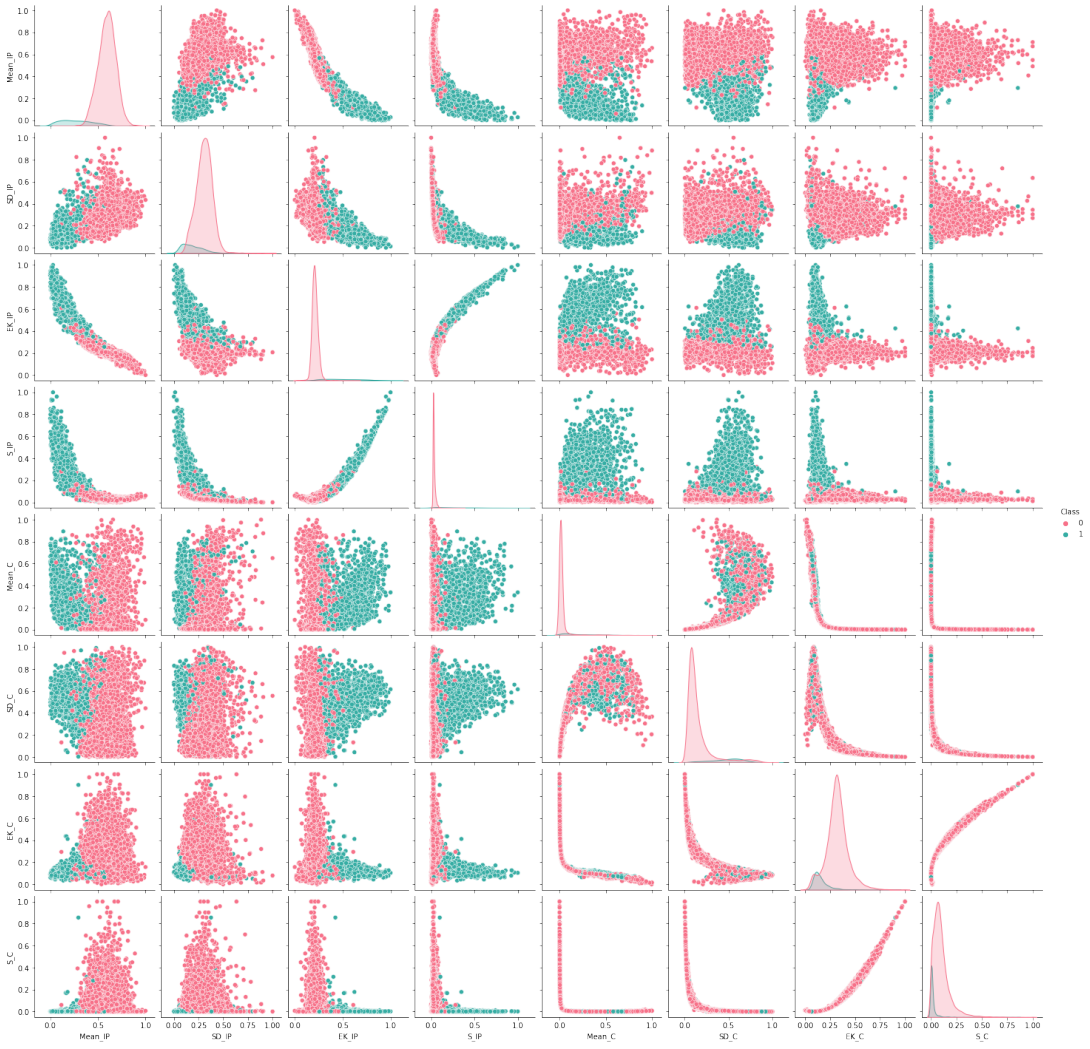
Therefore, to answer the research question set before, the Random Forest classifier has the best performance out of all models for this project, with parameter: criterion: gini, max_depth: 30, max_features: log2, n_estimators: 30. The astronomers could use this Random Forest classification model to identify whether they detected is a true pulsar very quickly with high accuracy.

Conclusion

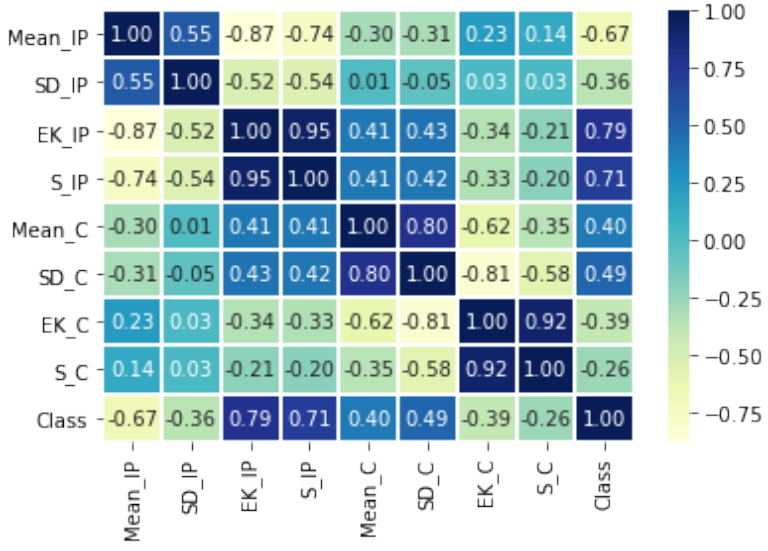
By doing this project, the most important thing I have learnt is how to do the rigorous analysis of the comparison among prediction models. The T-test or H-test could identify the significant difference among different models. Besides, normally, the most important score among the result of the classification report is the accuracy, but due to the different situation of datasets or topics, the other score in the classification report should be noticed as well. For instance, the F1-score of this classification problem is more important than the accuracy, due to the imbalance data.

Also, the pre-processing of the original data might have an impact on the performance of classifiers as well. From the visualisation of features, it can be seen that the distribution of some features is highly skewed. Apply log transformation on those features could reduce this skewed distribution and affect the model performance. The feature importance has also been checked during this project and the result can be found in the Appendix. According to the feature importance, not all features have the same importance on the model. Some features are much more important than the others. Therefore, in further research, introduce part of features instead of using all features to classification models might get different performance.

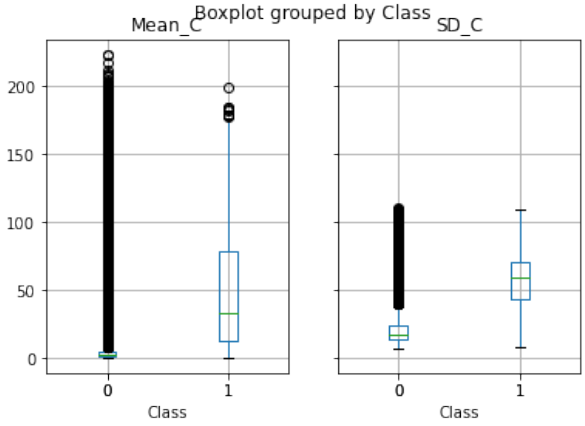
Appendix



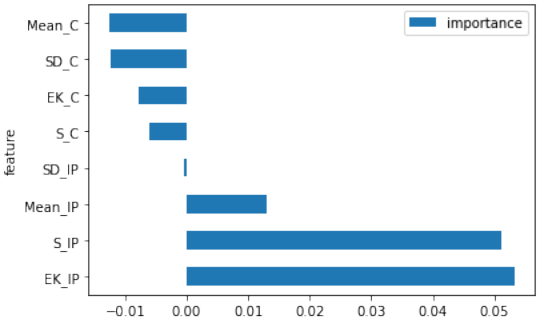
Visualisation of Features



Feature Correlation Heat-map



Check outliers in "Mean_C" and "SD_C"



Feature Importance

Classification report of validation set:

	precision	recall	f1-score	support
0	0.98	0.99	0.99	3916
1	0.94	0.82	0.87	380
accuracy			0.98	4296
macro avg	0.96	0.91	0.93	4296
weighted avg	0.98	0.98	0.98	4296

Random Forest Classification Report

Classification report of validation set:

	precision	recall	f1-score	support
0	0.99	0.98	0.98	3916
1	0.79	0.88	0.83	380
accuracy			0.97	4296
macro avg	0.89	0.93	0.91	4296
weighted avg	0.97	0.97	0.97	4296

Logistic Regression Classification Report

Classification report of validation data:

	precision	recall	f1-score	support
0	0.98	1.00	0.99	3916
1	0.94	0.80	0.87	380
accuracy			0.98	4296
macro avg	0.96	0.90	0.93	4296
weighted avg	0.98	0.98	0.98	4296

KNN Classification Report

Classification report of validation data:

	precision	recall	f1-score	support
0	0.99	0.99	0.99	3916
1	0.89	0.87	0.88	380
accuracy			0.98	4296
macro avg	0.94	0.93	0.93	4296
weighted avg	0.98	0.98	0.98	4296

SVM Classification Report

Classification report of validation data:

	precision	recall	f1-score	support
0	0.98	0.96	0.97	3916
1	0.65	0.84	0.73	380
accuracy			0.95	4296
macro avg	0.82	0.90	0.85	4296
weighted avg	0.95	0.95	0.95	4296

Naive Bayes Classification Report