**Stellar Classification**

**Setup**

***Research question***

There are still a lot of endless mysteries the boundless space. As astronomy is a traditional science, it has significant contributions to human beings, and it has strong inspirations in the future development of the science world. The deep exploration of the universe has always been the dream of human race.

The foundation star classes such as stars, galaxies, planets, and quasars are the fundamental in astronomy. At the beginning stage, the challenge is to identify the types of stars. The methods how to classify the stars are various, such as the size, color, life cycle. One way is that stars can be classified based on their spectral characteristics, which is called stellar classification. The stellar classification plays an important role in early stage.

In this pathway, it is expected that stellar classification method can reinforce the identification of the stars. In reverse, with the provided spectral characteristics, the astronomers will be able to conclude the types of the stars.

***Data***

The data was downloaded from Kaggle website: <https://www.kaggle.com/datasets/fedesoriano/stellar-classification-dataset-sdss17> and named stellar classification dataset – SDSS17. This data consists of 100000 observations from space taken by Sloan Digital Sky Survey (SDSS). Data is cleaned and ready for next step. As I had mentioned the process of data cleaning in stage 1. Details about cleaned data, please see the table 2 in appendix.

***Proposal***

The goal simply would be to build classification models for the star based on the eight attributes. Since we wanted to classify the class based on their eight attributes, the classification model would be suitable. However, the challenge was how to achieve a high-accurate classification model. The core classification algorithm would be random forest classification. And the benchmarks of the classification algorithms would be multiple logistics classification, decision tree classification and K nearest neighbor classification. The tool for this report was python with a few libraries such as ‘SKlearn’ and ‘matplotlib’. The dataset would be separated to two parts, the training set, and the testing set. The training set was used to train the models, and the testing set was used to check how the models were. Performance measures would be executed with confusion matrix to receive accuracy, precision, recall, f1 scores, and the results among the algorithms would be compared. Accuracy, precision, recall, f1 scores were better when they were close to 1. Tuning with ‘GridSearchCV’ would be used to find the best parameters of model to acquire the best model. Then, The McNemar’s test would be used to do the hypothesis tests between the random forest classification and benchmark models. The research question for the McNemar’s test would be if there is a difference of performance between two models. Detail null and alternative hypotheses were in approach section. At last, the graph of training error versus the generalization error would be plotted to verify whether the model was overfitted for the core model random forest classification. Finally, the random forest classification model can classify a star, galaxy, or quasar depending on their properties.

**Approach**

***Random forest classification***

A random forest classification was powerful ensemble classifier that fitted many decision trees on the sub-set of the training data and took the averaging to improve the performance of the model. The ‘RandomForestClassifier’ and ‘GridSearchCV’ in ‘SKlearn’ would be utilized for various parameters of tuning. Many parameters for random forest can be modified. The major parameters were the number of estimators, criterions, and max depth. The number of estimators was the number of the tress in the forest. The criterions were the different function to measure the quality of a split. The max depth was the maximum depth of every decision tree. The best tuning result would be used into the best model for the research question.

***Decision tree classification***

As mentioned on random forest classification, the decision tree classification was also a classification algorithm. The tree was constructed in a top-down recursive manner based on the attributes. The ‘DecisionTreeClassifier’ and ‘GridSearchCV’ in ‘SKlearn’ would be utilized for various parameters of tuning. The parameters were partial the same as random forest classification. The max depth and criterions were tuned. Also, splitter is in the ‘GridSearchCV’. Splitter was the strategy that chose the split at each node.

***K nearest neighbor classification***

K nearest neighbor algorithm was also a common algorithm for both classification and regression. The ‘KNeighborsClassifier’ and ‘GridSearchCV’ in ‘SKlearn’ would be utilized for various parameters of tuning. The parameters such as ‘algorithm’ and ‘p’ were well defaulted by the classifier itself. ‘algorithm’ was on ‘auto’ and ‘p’ was on ‘2’, which indicates the Euclidean distance. The only major parameters that needed to be tuned was the k value, which is the number of neighbors to use.

***Multiple logistics classification***

Logistics classification was a popular binary classifier. The ‘multiple’ version of it allowed the logistics regression worked on more than two classes. The ‘LogisticRegression’ and ‘GridSearchCV’ in ‘SKlearn’ would be utilized for various parameters of tuning. The only parameters required to be tuned were penalty and max iteration. The ‘solver’ was ‘sage’ and ‘multi\_class’ was 'multinomial' because they were the settings for the multiple logistics regression.

***McNemar’s test***

McNemar’s test was a statistical test for 2 × 2 contingency tables. It was a chi-squared test statistic. In this report, it was applied on the confusion matrix from the performances between the models. The null hypothesis was that there is no difference between two models. The alterative hypothesis was that there was a difference between two models. If the p-values of the tests were smaller than 0.05. Then reject the null hypothesis and vice versa.

***Overfitting***

In case of overfitting, the complexity analysis would be performed. The dash line of generalization error should show the switch from underfitting to overfitting. However, both dash lines may not either underfitting or overfitting in this case because the larger samples were, the better the models were, and this dataset contains 99999 observations. The relatively large data sample cause low possibility of underfit and overfit.

**Result and Analysis**

***Random forest classification***

In the beginning, random forest classification with only 2 max depth was established. It received accuracy of 0.89, see table 3 for more details. The ‘GridSearchCV’ set max depth from 7 to 14. The criterions were among ‘entropy’,’gini’,’log\_loss’. The number of estimators were among 20,40,60. The best combination was criterion of ‘entropy’, max depth of 13 and the number of estimators of 40. After applied ‘GridSearchCV’ in python, the random forest classification represented a high accuracy of 0.98, which is relatively high accuracy, see table 4 for more details. The accuracy increased from 0.89 to 0.98 after ‘GridSearchCV’. The tuning was very effective.

***Decision tree classification***

As the same as random forest, the max depth was set at 2 before the tuning. Accuracy was 0.95, see table 5 for more details. The ‘GridSearchCV’ set max depth from 2 to 12. The criterions were between ‘entropy’ and ’gini’. The splitters were between ‘best’ and ‘random’. The best combination was criterion of ‘gini’, max depth of 9 and splitter of ‘best’.

After applied GridSearchCV in python, the decision tree classification represented a high accuracy of 0.98, which is relatively high accuracy, see table 6 for more details, and it was similar to the result of random forest classification. The accuracy increased from 0. 95 to 0.98 after ‘GridSearchCV’. The tuning was effective.

***K nearest neighbor classification***

A random number 5 was chosen in the beginning to be the k value. The accuracy of 0.83 had received, see table 7 for more details. Since the main tuning value of KNN classification was the k value. The GridSearchCV was only changing k values and established different model. The k value ranges were from 3 to 20.

After applied the GridSearchCV in python, the best value of k was 3. The accuracy of the model was 0.84, which was still high but not as high as decision tree and random forest classifications, see table 8 for more details. The accuracy increased from 0.83 to 0.84 after ‘GridSearchCV’. The tuning was not very effective.

***multiple logistics classification***

In the beginning, the multiclass argument was set as ‘multinomial’ since there are 3 classes of stars to classify. The solver of this classification was ‘saga’ and the max iteration was 2000. The accuracy was 0.95, which was great, see table 9 for more details. However, an error was called that the max iteration reached and not converged. Therefore, the max iteration was starting from 3000 to 6000 during the GridRearchCV parts. Also, the penalty of ‘l1’, ‘l2’ and ‘elasticnet’ were selected.

After applied GridRearchCV in python, the accuracy did not changed, which was still 0.95, see table 9 for more details. There was slightly increasing inside precision, recall and f1-score however.

***McNemar’s test***

The major hypothesis tests performed were McNemar’s test between the random forest model and other benchmark models. From the performance of the four classification models, this is not a surprise to receive the result of ‘fail to reject null hypotheses’ to all the McNemar’s test, see the output 1 in appendix. It indicted that there was no evidence proved that the performance of random forest classification model differed to other models.

***Overfitting***

To make sure the model of random forest classification was not overfitting, a complexity analysis by the plot of generalization error and training error was performed. As the graph shown, the tails of the line became flat very quickly. Even at the beginning of the line, both errors were low. Therefore, I can conclude that there was no overfitting. Recalling in the approach, this flat line appeared due to the large sample data, so the model had little error.

图形用户界面

描述已自动生成

Figure 1 plot generalization error and training error for random forest classification.

In table1 below, accuracy, precision, recall and f1 score represented the performance of the models. The higher values were, the better the models were. The table summarized the accuracy and the average precision, recall and f1 scores. The reliability of models was determined based on the accuracy. The random forest classification and decision tree classification had the highest accuracy 0.98. The KNN classification had the lowest accuracy 0.84. The accuracy with 0.98 was very high and almost close to 1. This indicated the random forest classification model had high reliability on predicting the class of stars based on their properties. Besides the accuracy, the precision, recall and f1 score can also demonstrate the performance of the models. The comparison results were like the accuracy. Random forest classification had the highest performance and slightly better than decision tree classification. The KNN remained the worst performance among all the model, however, it was still not bad.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Classification | accuracy | precision | recall | f1 score |
| random forest | 0.98 | 0.977 | 0.973 | 0.977 |
| decision tree | 0.98 | 0.977 | 0.97 | 0.973 |
| KNN | 0.84 | 0.83 | 0.78 | 0.797 |
| logistics regression | 0.95 | 0.943 | 0.947 | 0.943 |

Table 1 the summary of accuracy, precision, recall and f1 scores among four classification models.

Due to the large sample size, the parameters like the number of estimators and max depth of random forest algorithm can not be very large because of the hardware limitation. It may perform better with larger parameters.

**Conclusion**

The core model random forest algorithm had a great performance and high reliability with the accuracy of 0.98 on the classification of the star based on their properties. This model had no overfitting problem. When comparing this model to other benchmark models, they did not have a significant difference among them. I believe it was due to the large sample sizes to cause all the model having a great performance. I would highly recommend my random forest classification model as a solution of my problem as the reasons above.

From this modelling process, I have learnt a lot. I knew how to use GridSearchCV to tune the algorithm to acquire the best model. Also, I knew how the large data size could significantly influence the performance of the models.

**Appendix**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| column | type | mean | median | standard deviation | max | min |
| class | object | - | - | - | - | - |
| right ascension angle(alpha) | float | 177.63 | 180.9 | 96.5 | 360 | 0.0055 |
| declination angle(delta) | float | 24.13 | 23.64 | 19.64 | 83 | -18.79 |
| ultraviolet filter(u) | float | 22.08 | 22.18 | 2.25 | 32.78 | 11 |
| green filter(g) | float | 20.63 | 21.09 | 2.04 | 31.6 | 10.5 |
| red filter(r) | float | 19.64 | 20.12 | 1.85 | 29.57 | 9.82 |
| near infrared filter (i) | float | 19.08 | 19.41 | 1.76 | 32.14 | 9.47 |
| infrared filter(z) | float | 18.67 | 19 | 31.73 | 29.38 | 9.61 |
| redshift | float | 0.58 | 0.42 | 0.73 | 7.01 | -0.01 |

Table 2 summary of the eight attributes: mean, median, standard deviation, maximum value and minimum value.

表格

描述已自动生成

Table 3 Accuracy, precision recall and f1 scores for random forest classification before ‘GridSearchCV’

表格

描述已自动生成

Table 4 Accuracy, precision recall and f1 scores for random forest classification after ‘GridSearchCV’

表格

描述已自动生成

Table 5 Accuracy, precision recall and f1 scores for decision tree classification before ‘GridSearchCV’

表格

描述已自动生成

Table 6 Accuracy, precision recall and f1 scores for decision tree classification after ‘GridSearchCV’

表格

描述已自动生成

Table 7 Accuracy, precision recall and f1 scores for K nearest neighbor classification before ‘GridSearchCV’

表格

描述已自动生成

Table 8 Accuracy, precision recall and f1 scores for K nearest neighbor classification after ‘GridSearchCV’

表格

描述已自动生成

Table 9 Accuracy, precision recall and f1 scores for multiple logistics classification before ‘GridSearchCV’

表格

描述已自动生成

Table 10 Accuracy, precision recall and f1 scores for multiple logistics classification after ‘GridSearchCV’

文本

描述已自动生成

Output 1 McNemar’s tests results in the performance between random forest classification and other three classification models separately.

Reference

Kaggle 2022, Stellar Classification Dataset - SDSS17, Kaggle, viewed 28 March 2016,

<https://www.kaggle.com/datasets/fedesoriano/stellar-classification-dataset-sdss17>

Sklearn.ensemble.randomforestclassifier. scikit. (n.d.). Retrieved May 22, 2022, from <https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html>

Sklearn.neighbors.kneighborsclassifier. scikit. (n.d.). Retrieved May 22, 2022, from <https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html>

Sklearn.linear\_model.LogisticRegression. scikit. (n.d.). Retrieved May 22, 2022, from <https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html>

Sklearn.tree.DecisionTreeClassifier. scikit. (n.d.). Retrieved May 22, 2022, from <https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html>