**Classification Exercises**

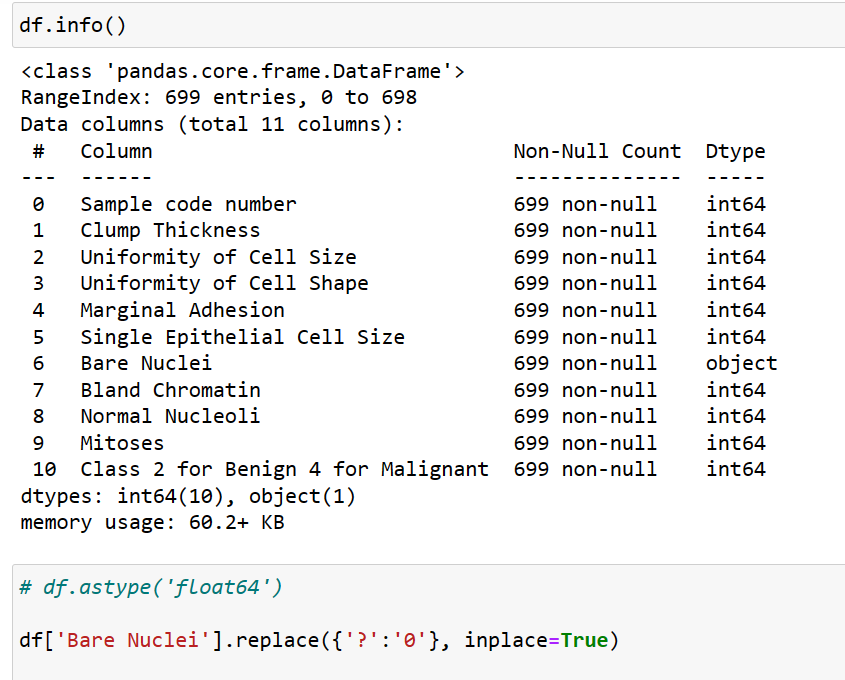
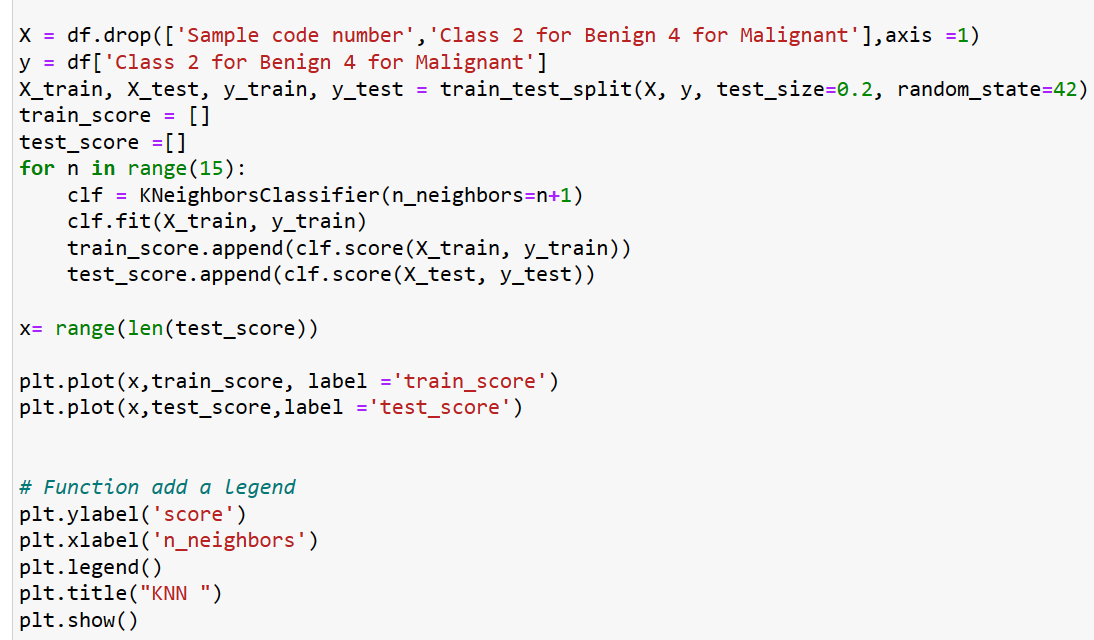
B08504052 化工四 賴正鎧

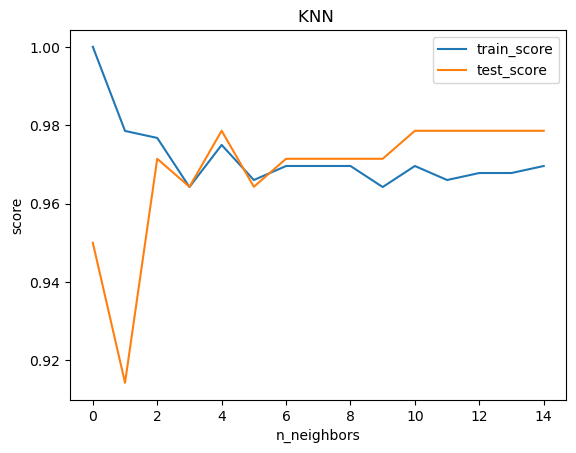
1. **Breast cancer data: Use the dataset “Breast Cancer Wisconsin.xls” to build a model to diagnose breast cancer.**

Build a classification model using various methods below and compare your models.

* + 1. **KNN .. Try a few K values. Report your best K**

Here is my script. First, there is some weird data inside, so I replaced them first. Left hand side is the data cleaning, right hand side is the KNN classify script.

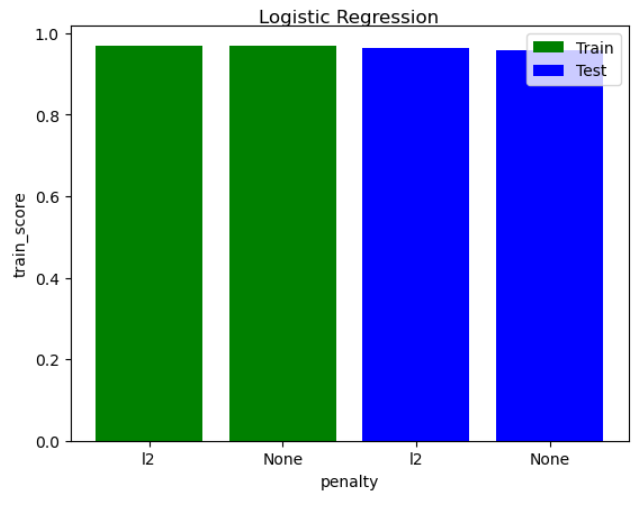
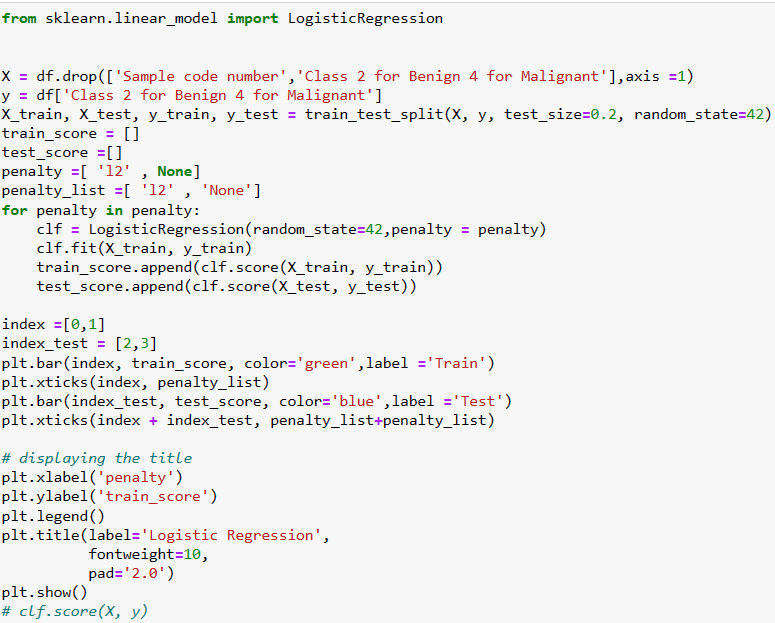




at n\_neighbors = 4, the test set performance is the best !

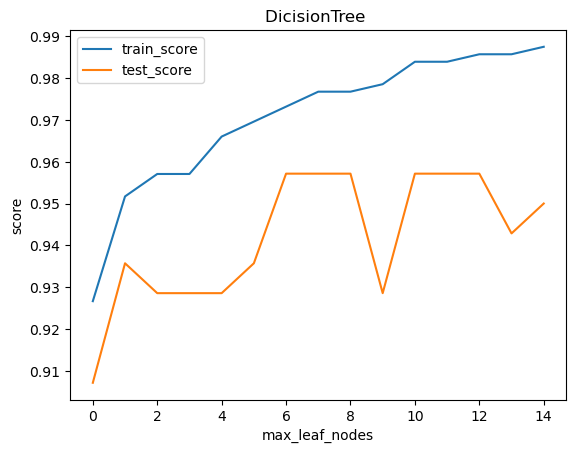
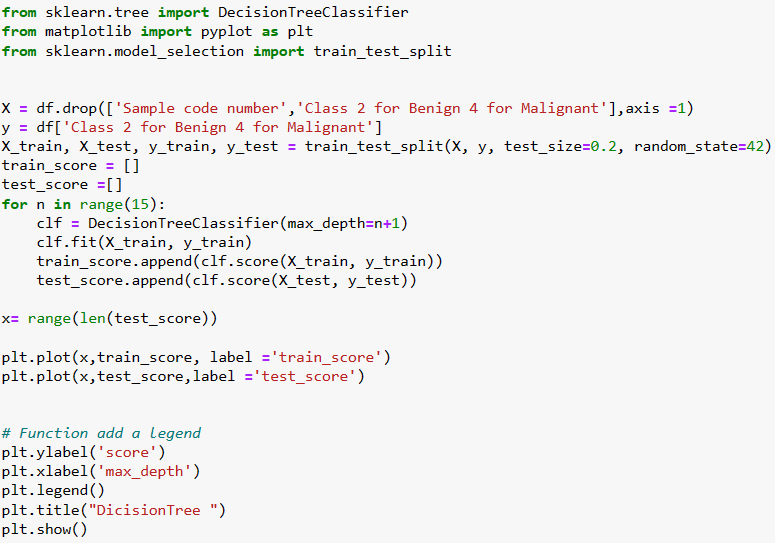
* + 1. **Logistic Regression**

Left hand side is the Logistic Regression classification script. Right hand side is the figure comparing the performance between different penalty type and train and test set, in this dataset, there isn’t a big different, everyone did a great job!



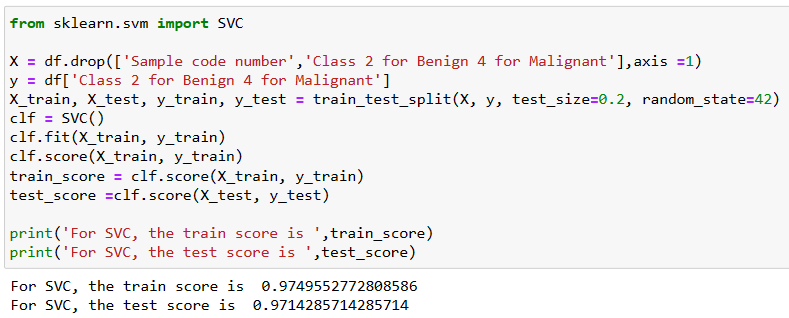
* + 1. **Decision Tree.. Try a few “Max Number of Splits”. Report your best choice.**

Left hand side is the Decision Tree classification script. Right hand side is the figure comparing the performance between different “max\_leaf\_nodes”, it showed that the test set performance stop growing when “max\_leaf\_nodes”6, and the score is approximately 0.96.



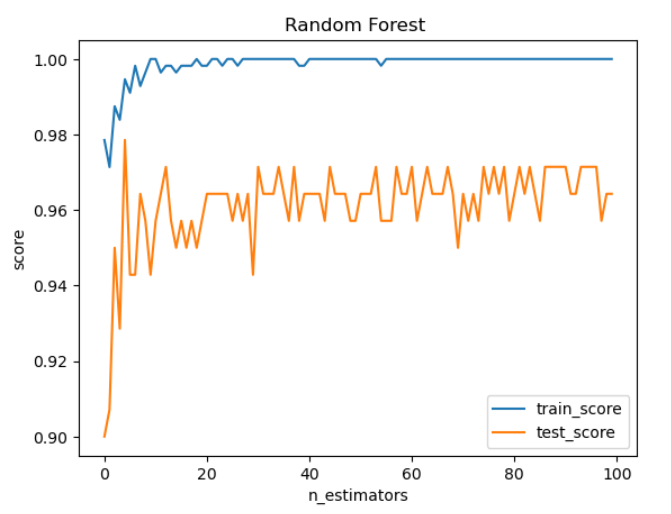
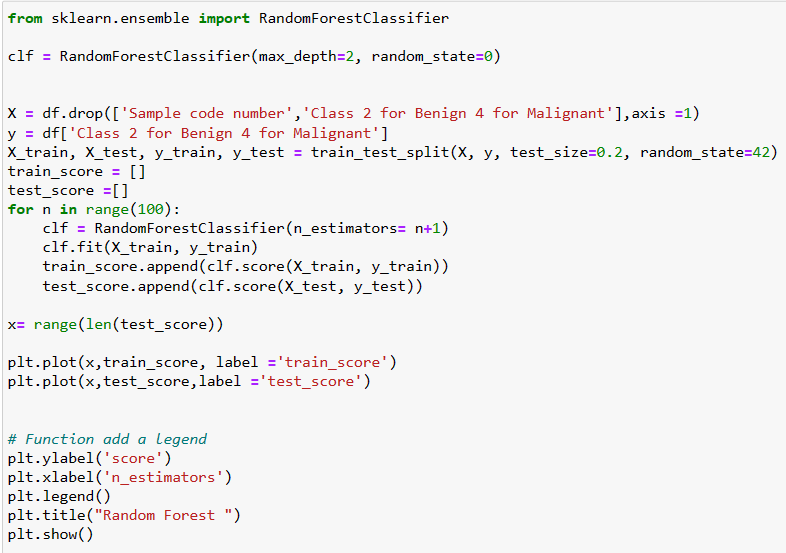
* + 1. **SVM.**

The SVM script is shown below. I think SVM is a really strong algorithm, it often perform well in all of the situation. The score is 0.975, such a nice performance!!



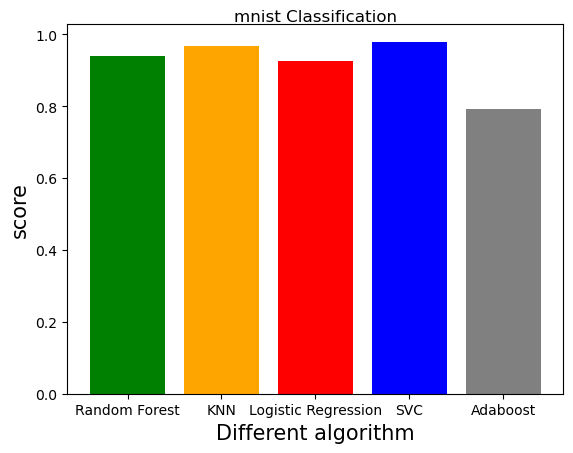
* + 1. **Ensemble of your choice: Boosted Tree and Bagged Tree.**

The figure showed the relationship between accuracy score and n\_estimators. To me, it’s not surprised that random forest is better than decision tree.

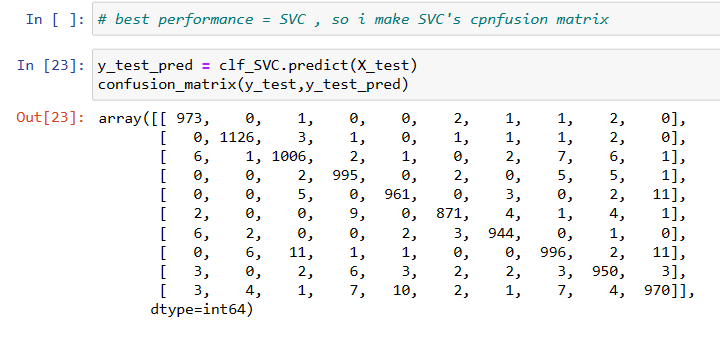


1. **Handwritten digits recognition**

I have tried five classification algorithms, they are Random Forest, KNN, Logistic Regression, SVC and Adaboost. And I organized the performance in the histogram below



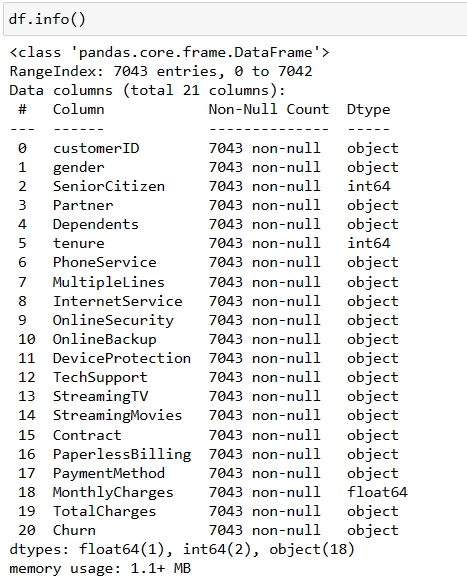
SVC is the champion of mnist !!! The confusion matrix is below.

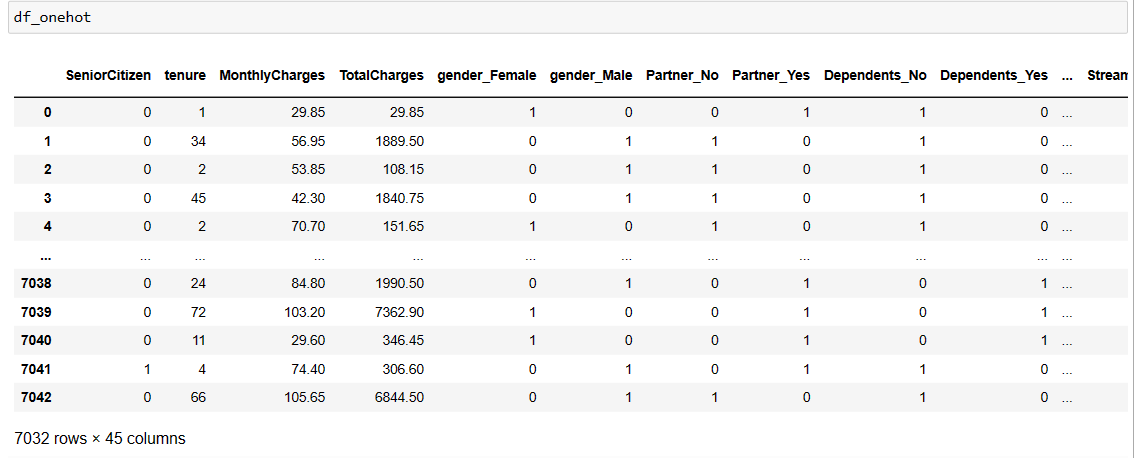


1. **Telco Customer Churn:**

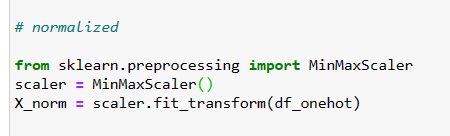
* Split the data set and build binary classifiers using various methods of your choice.
* Use at least one of the Feature Selection methods we mentioned to select your features as part of the model building process.

Compare algorithms and choose your best model. Report the performance metrics and confusion matrix.

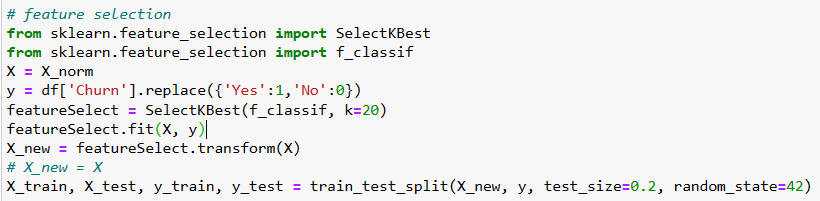
the data contains a lot of categorical part, and it also have some empty value in numerical part (Total Charges), so I clean the empty value by remove the whole row. After that, I one hot encode the categorical part of the data by pd.get\_dummies, it’s really a useful function, so convenient! Below is the feature after encoding.



I normalized the data because I want to apply SVC later, and it’s sensitive to the scale of the data.



Then, I apply select\_k\_best in sklearn.featureselection to find the top 20 best feature in the data.



Then , I started to do the classification. In this question, I have tried six algorithms, they are KNN, random forest, logistic regression, gradient boost, adaboost and SVC, again, SVC is the winner, but the accuracy this time is only 0.8. Below is the performance of all algorithms I applied. In fact, they are quite similar, even KNN.



Below is the confusion matrix and the performance metrics .

