**DS620 Machine Learning and Deep Learning**

**HOS02A Classification**

03/25/2021 Developed by Minh Nguyen

04/08/2021 Reviewed by Shanshan Yu

School of Technology & Computing @City University of Seattle (CityU)

**Learning outcome**

* Machine Learning process for Classification problem
* Model selection with Cross validation
* Fine Tuning model
* Evaluate Classification models

**Resources**

* Cross Validation: Data Science Concepts. (2020, October 19). [Video]. YouTube. <https://www.youtube.com/watch?v=wjILv3-UGM8>
* K-Fold Cross Validation - Intro to Machine Learning. (2015, February 23). [Video]. YouTube. <https://www.youtube.com/watch?v=TIgfjmp-4BA>
* scikit-learn: machine learning in Python — scikit-learn 0.24.1 documentation. (n.d.). Scikit-Learn. <https://scikit-learn.org/stable/index.html>
* Cross-validation: evaluating estimator performance — scikit-learn 0.24.1 documentation. Scikit-Learn. <https://scikit-learn.org/stable/modules/cross_validation.html>
* Massaron, L., & Boschetti, A. (2016). Regression Analysis with Python. Packt Publishing.
* Géron, A. (2019). Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow. O'Reilly Media, Inc.
* Data source: <https://www.kaggle.com/blastchar/telco-customer-churn>

**Introduction**

In the previous Hands-on-Skill, we focused on the first 3 steps of a Machine Learning project especially the preprocessing portion. For this week’s exercise, I will walk you through the remaining steps of a Machine Learning project for a Classification problem - which is about predicting a discrete variable. Let’s refresh our memory by revising the overview of a ML project.

**Overview of a Machine Learning project**

1. Get the data.
2. Discover and visualize the data to gain insights.
3. Prepare the data for Machine Learning algorithms.
4. **Select a model and train it.**
5. **Fine-tune your model.**
6. Present your solution.
7. Launch, monitor, and maintain your system.

For this HOS, we will focus more on Step 4 and 5 which is the main part of a Machine Learning project

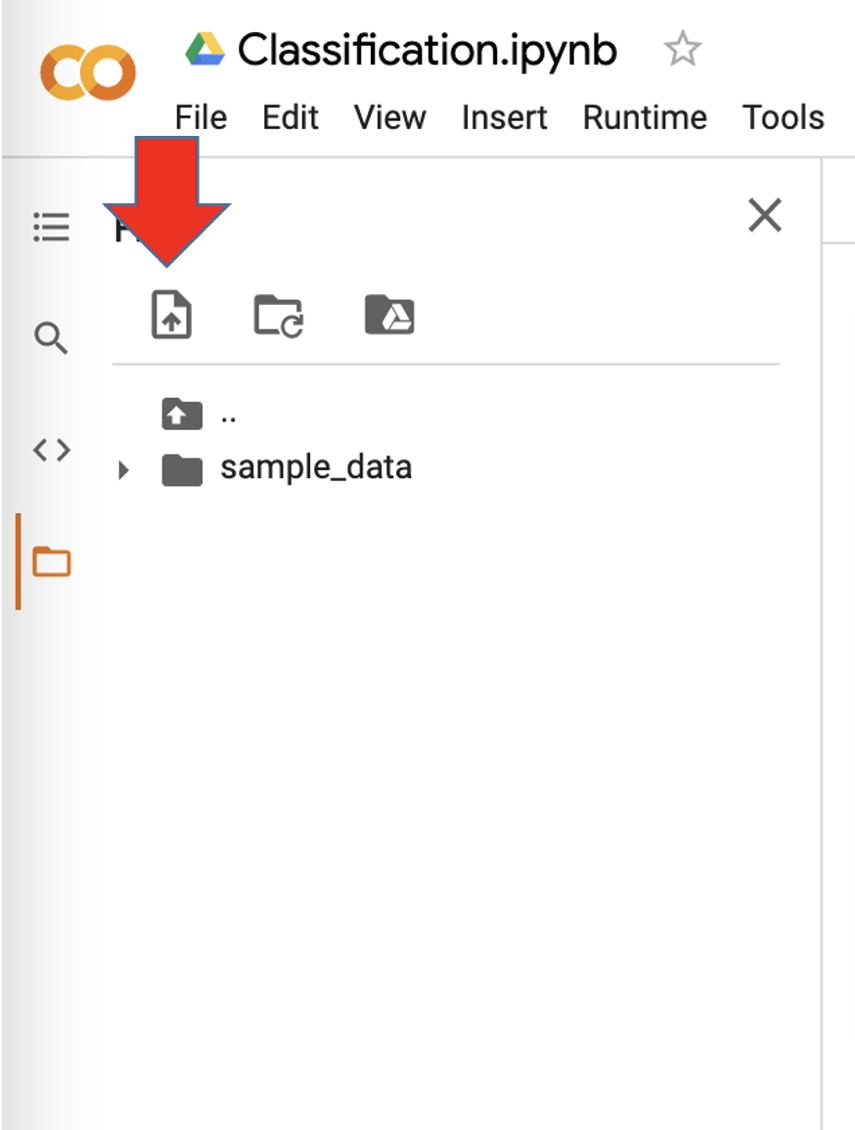
***Preparing development environment***

1. From [Google Colab](https://colab.research.google.com/), create a new notebook, name it “Classification.ipynb”
2. Type the following codes to import libraries.

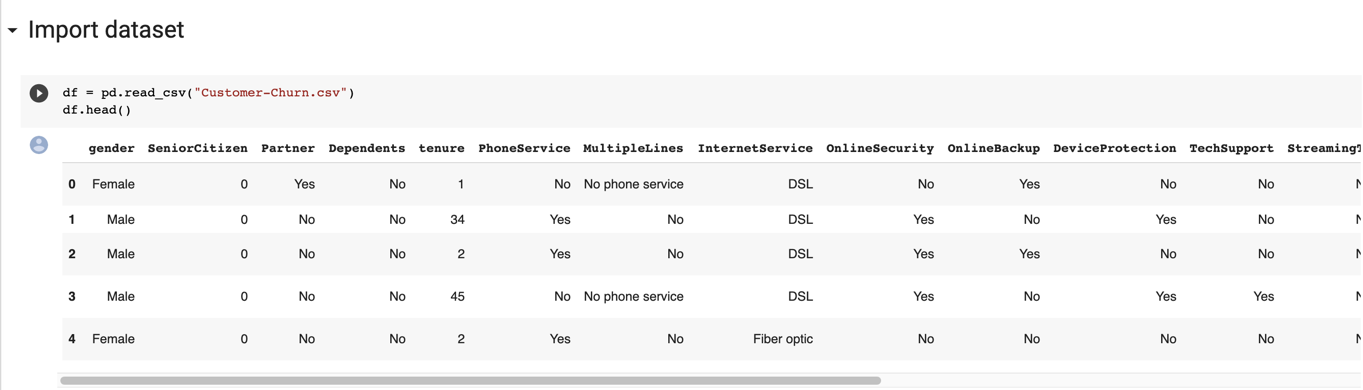


***Get the data***

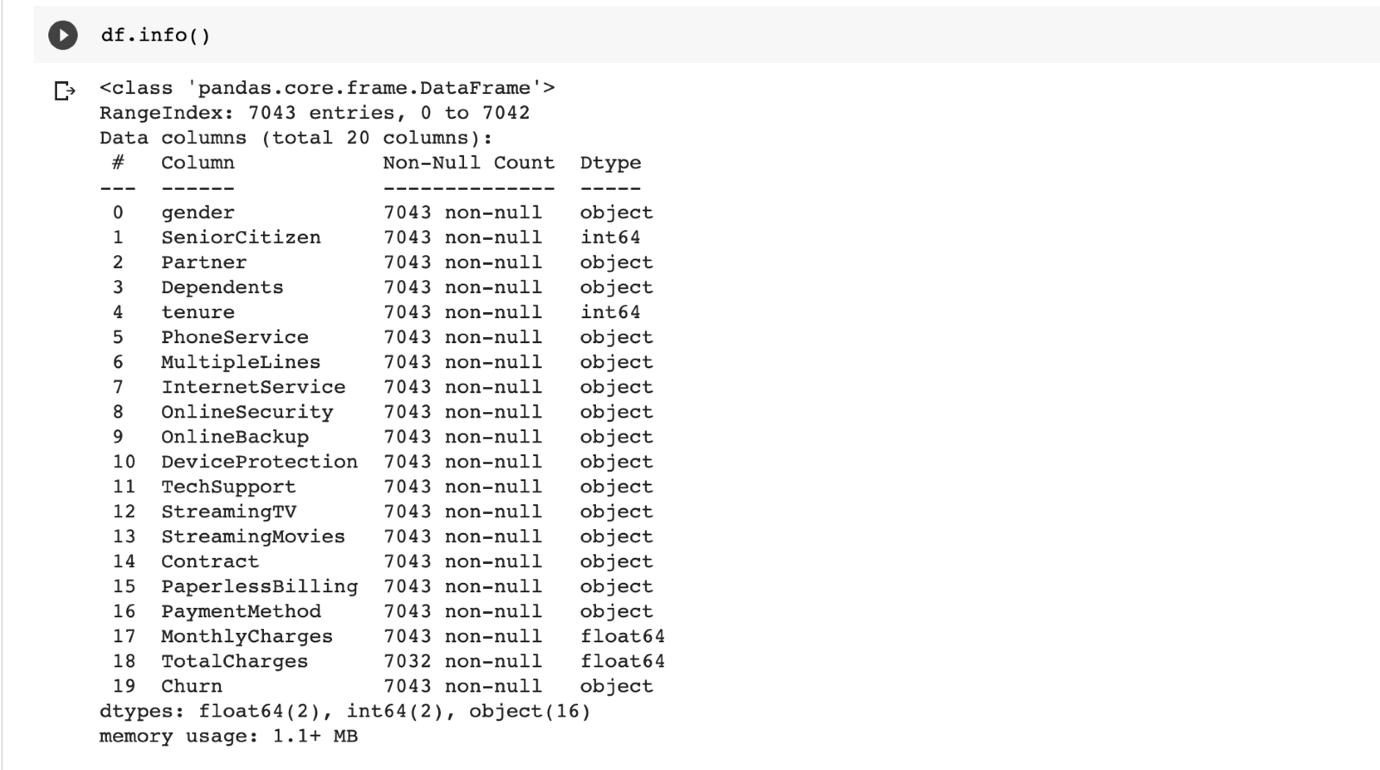
1. Upload the Customer-Churn dataset to Google Colab



1. Import the dataset

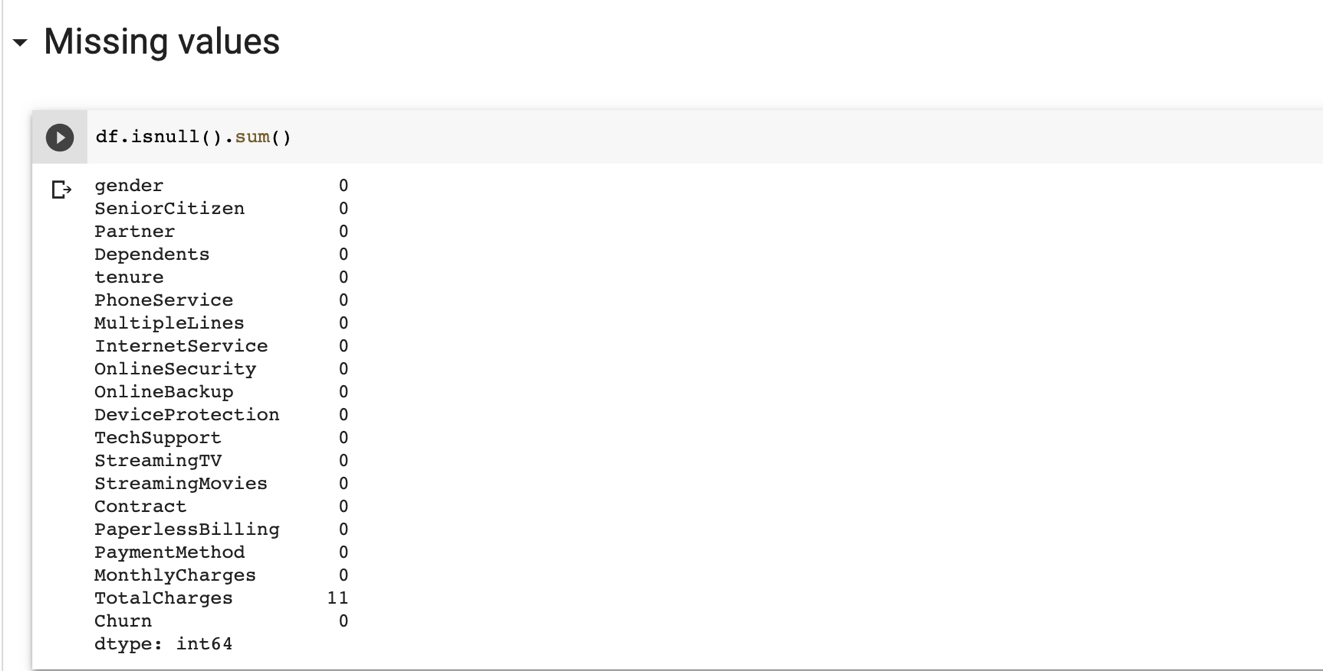


1. Get information about columns of the data.

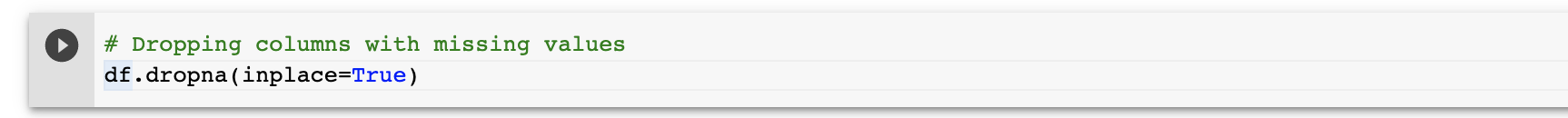


***Missing value***

* Like the previous HOS, use the isnull function to look for null value.



* In this exercise, however. We are going to drop the columns



*Remember to set the in place parameter to True to ensure that the adjustment is permanent.*

***Splitting data set***

* Use the train\_test\_split function to split the data to a training set and test set

Graphical user interface, application

Description automatically generated

*Remember to set the random\_state parameter to make the sampling reproducible*

***Preprocessing***

* The following codes build a preprocessor object to process both the training data and testing data

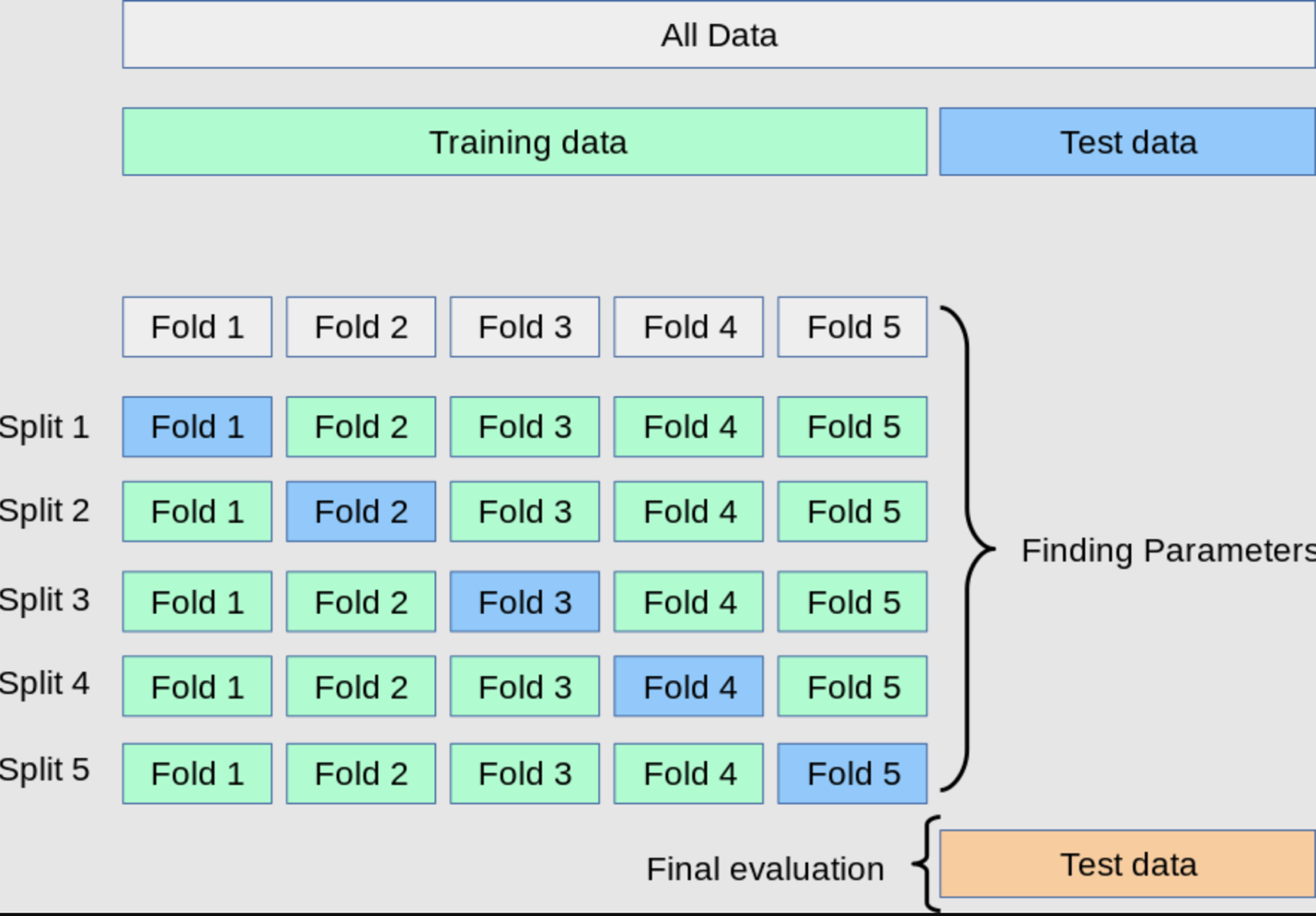
Graphical user interface, application

Description automatically generated

*Noticing that for X\_test and y\_test, I use transform instead of fit\_transform. The reason being is these classes for data preprocessing need some information from the data in order to make the transformation. For the scaler, it would be the mean and standard deviation from the continuous variables, for the encoder, it would be the categories names from the categorical variable. Fit\_transform takes this information and uses it to transform the respective columns while transform uses the previously saved information to make the transformation. We always use fit for testing data because we want to ensure that any new data being fed to the model is transformed in the same way as the training data.*

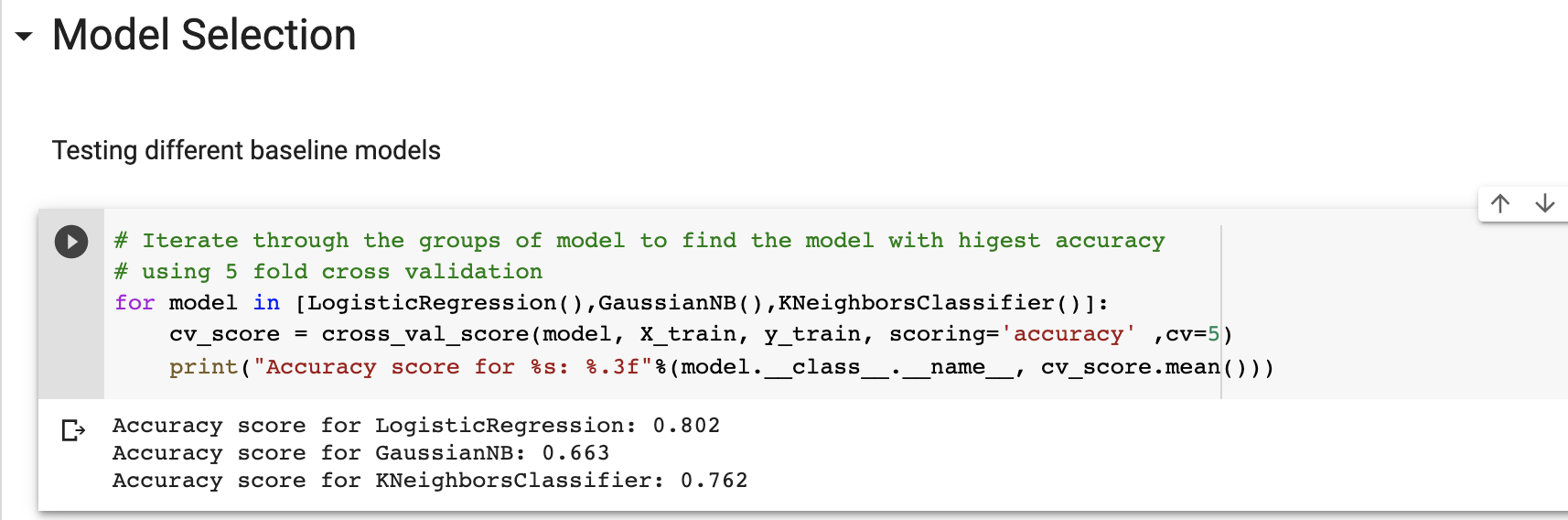
**I. Model Selection with Cross Validation**

K-fold cross validation is a method to quickly validate the performance of a model by training the model k time. For example, for 5 – fold cross validation, the dataset will be split in 5 different ways. For each method of splitting, the data set will be trained and validated, each method will return a performance indicator, the performance indicator for classification is accuracy score. After 5-fold cross validation is completed, we will get a list of 5 accuracy score which can be used to estimate the performance of the model. K-fold cross validation provide an unbiased way of evaluating a model performance as it returns a range of value instead of just 1. Scikit-learn provides cross\_val\_score function to help you with this.



*Scikit-Learn. K-Fold Cross validation [Illustration].* [*https://scikit-learn.org/stable/modules/cross\_validation.html*](https://scikit-learn.org/stable/modules/cross_validation.html)

* Get cross validation score of 3 different classification algorithms: Logistic Regression, Gaussian Naïve Bayes, and K Nearest Neighbor. Setting accuracy as the scoring metrics for classification models



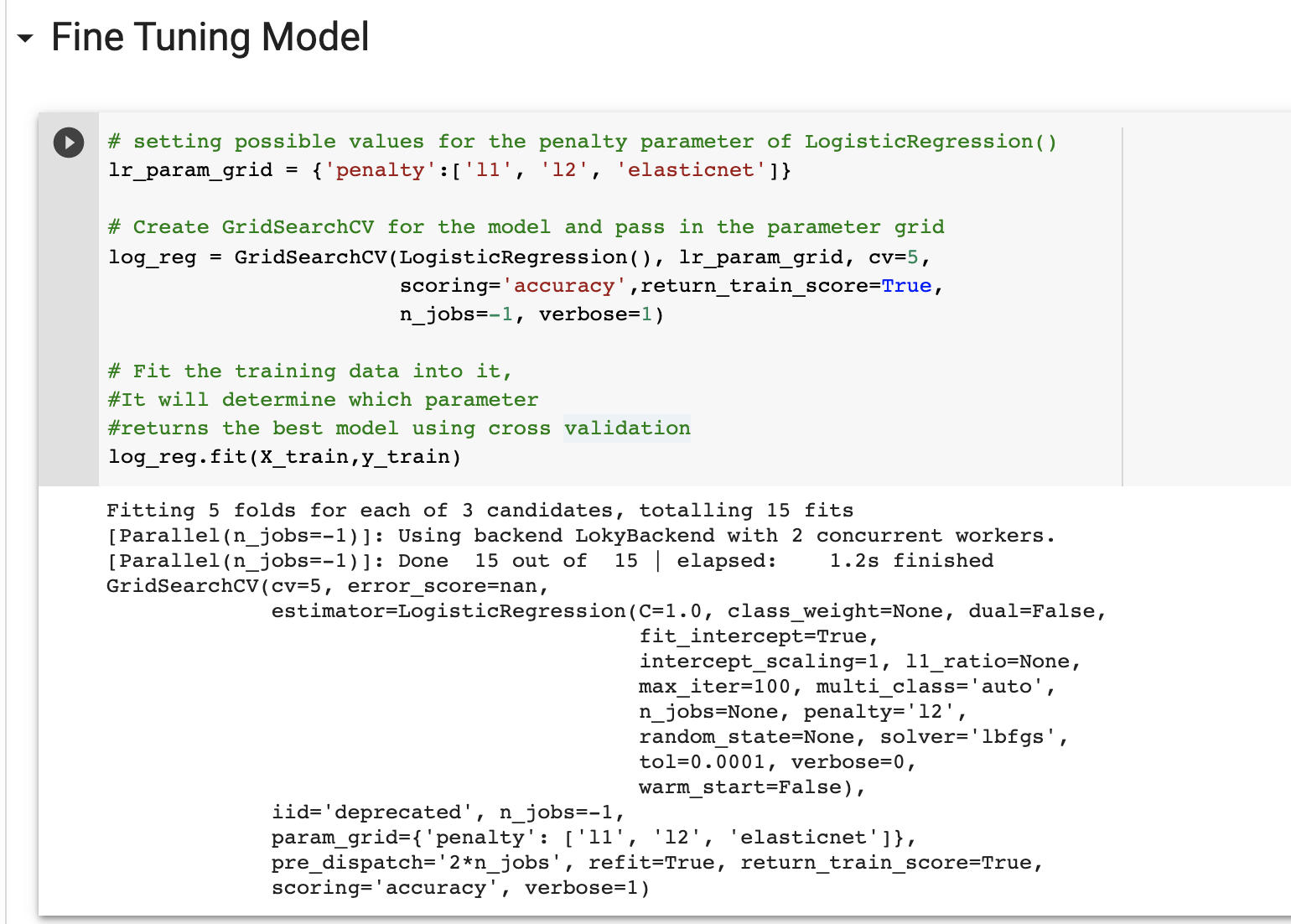
*Notice that we did not set any parameter for any of the models above. They are all baseline models. The goal of this is to find which model performs the best in the simplest form. We can see that logistic regression has the highest accuracy score. Therefore, we are going to tune the parameters to find the one that maximizes its performance.*

**II. Tuning parameters with Grid Search CV**

After you’ve selected the best baseline model, the next step is to improve it by setting the correct combination of parameters. The parameters to set are greatly different from one model to another and it requires the knowledge about that specific model to do so. It is a good practice to always look at the documentation from Scikit-Learn website to have a deeper understanding of the model you are tuning.

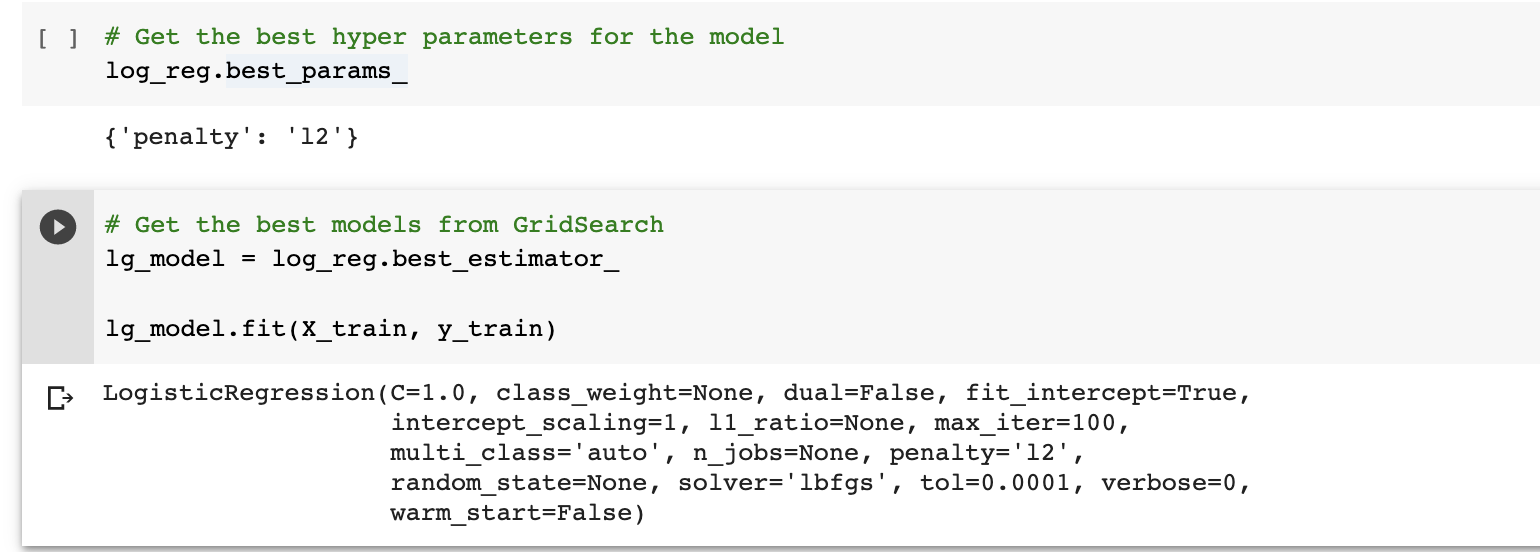
A good method of selecting the values for these model parameters is Grid Search CV. Grid Search CV train multiple models with each possible parameter and uses cross validation to return its accuracy score. Using the accuracy score, it will determine which model parameter is the most suitable for the model and the data. Your job is to set all the possible model parameters for Grid Search to test on.

1. Run the following code to perform Grid Search



The parameter to tune for the Logistic Regression is penalty. For other models, there could be more than 1 parameter with a different name.

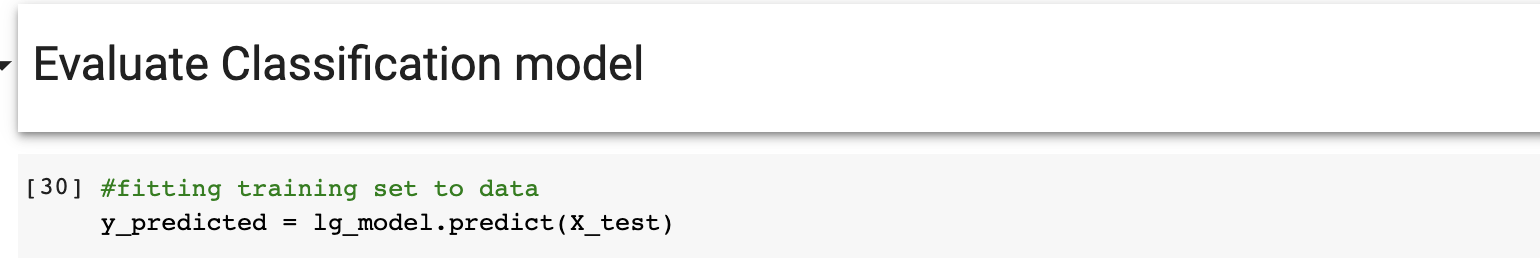
2. Run the following code to get the best parameter values as well as the best models then fit the training data to the model



**III. Evaluate Classification model**

There are various ways to evaluate a classification model, the most common one is accuracy score which is the percentage of correct classification. However, when your business goal requires more specific evaluation, a confusion matrix and a classification report can do the job. These 2 items will specify the True Positive, True Negative, False Positive, False Negative classification and metrics that derive from them.

1. Run the following code to use the model to make predictions on test data.



1. Run the following code to use 3 methods to evaluate the classification model.

Table

Description automatically generated

**Push Your Work to GitHub**

**Download the notebook from Colab**:

File -> Classification.ipynb

Move the downloaded file into your **Module2** working folder.

Open terminal and Navigate to the GitHub folder of this week HOS.

**Make sure the assignment files on the subfolder Module2 of hos02a\_YouGithubUserName folder, enter the following command to upload your work**:

>>>> git add .

>>>> git commit -m “Submission for HOS02”

>>>> git push origin master