**DS620 Machine Learning and Deep Learning**

**HOS04A Support Vector Machines and Ensemble Learning**

04/06/2021 Developed by Minh Nguyen

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

**Learning objectives**

* Support Vector Machine Classifier
* Ensemble Learning
* Voting Classifiers
* Bagging

**Resources**

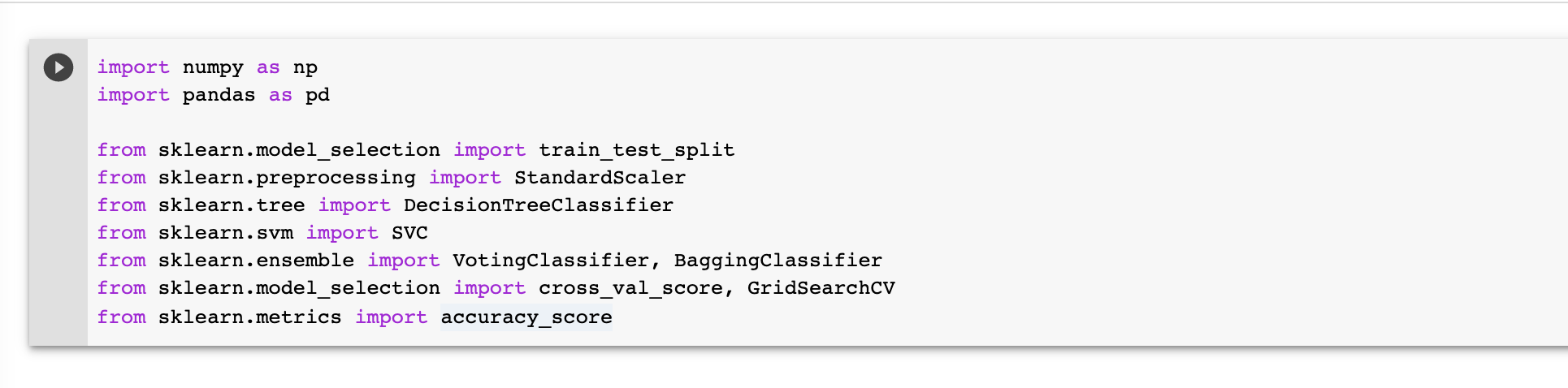
* scikit-learn: machine learning in Python — scikit-learn 0.24.1 documentation. (n.d.). Scikit-Learn. <https://scikit-learn.org/stable/index.html>
* Géron, A. (2019). Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow. O'Reilly Media, Inc.
* Bootstrap aggregating bagging. (2016, June 6). [Video]. YouTube. <https://www.youtube.com/watch?v=2Mg8QD0F1dQ>

**Introduction**

*Wisdom of the crowd* which refer to the concept that the collective opinion of a group of individuals is always better than that of a single expert. This often hold true when applying to Machine Learning Models, a better prediction can be made if you aggregate the predictions of a group of models, this is the basis for famous algorithms such as Decision Tree, XGBoost. In this HOS, we will cover the training process of the support vector machine algorithm and go over how

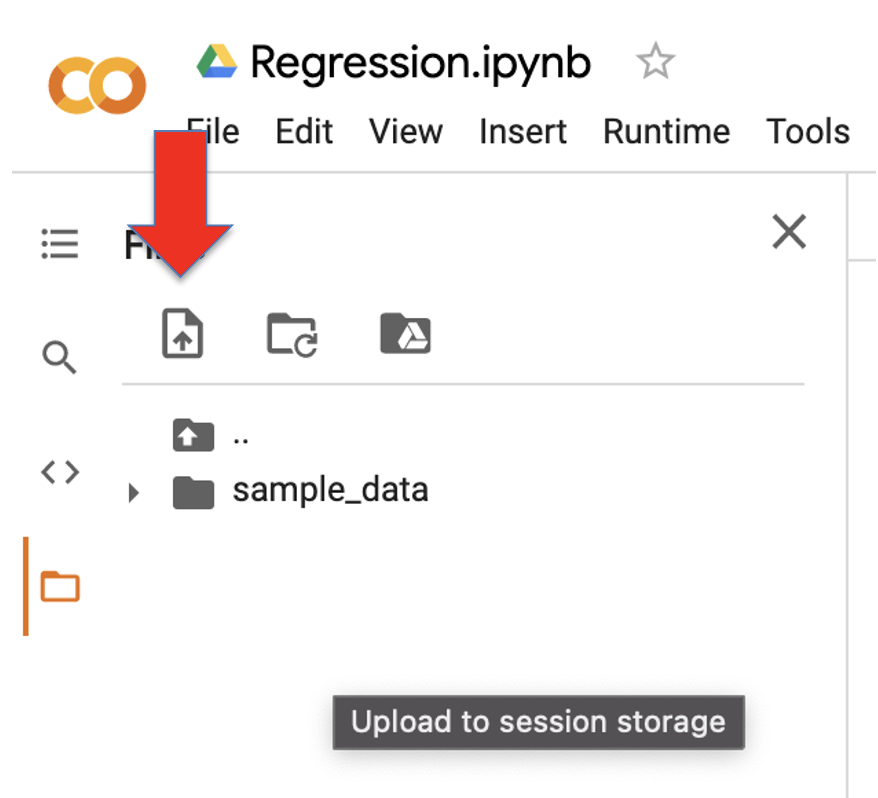
***Preparing development environment***

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

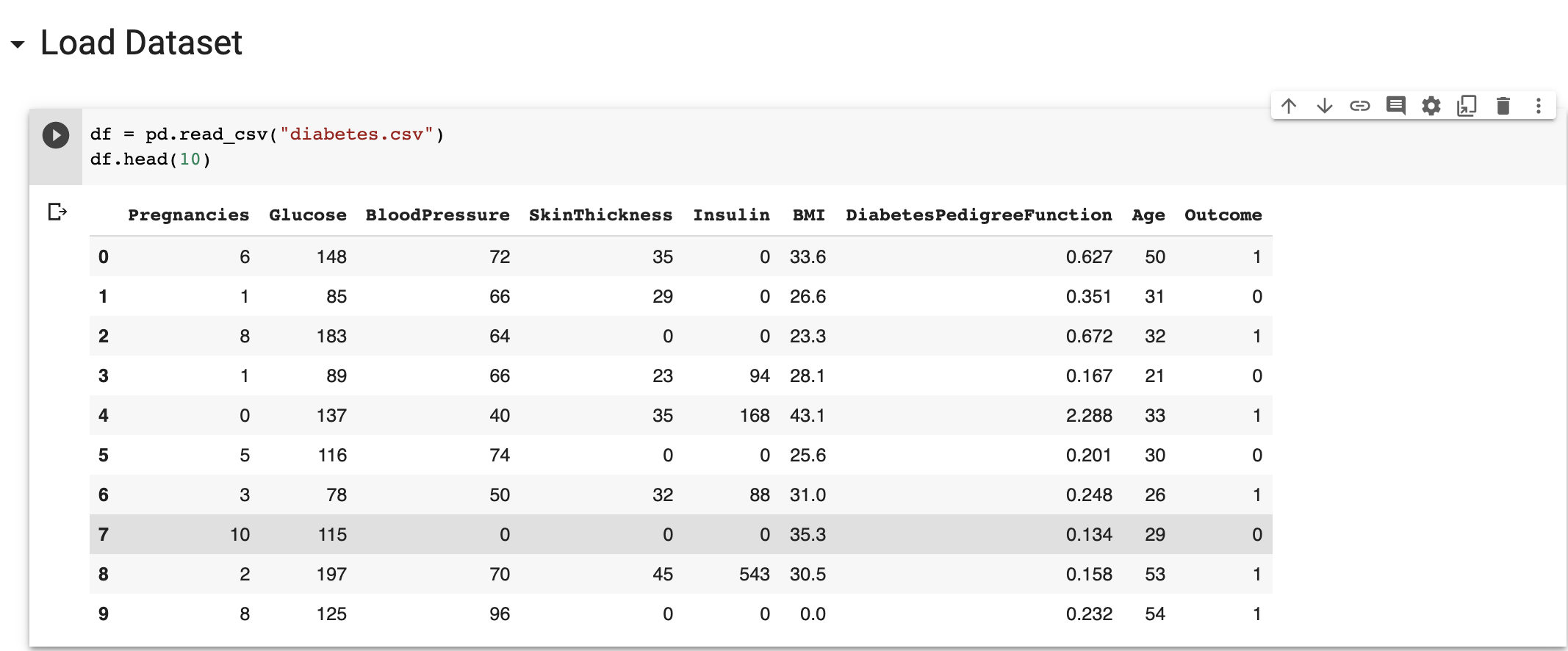


**1. Get the data**

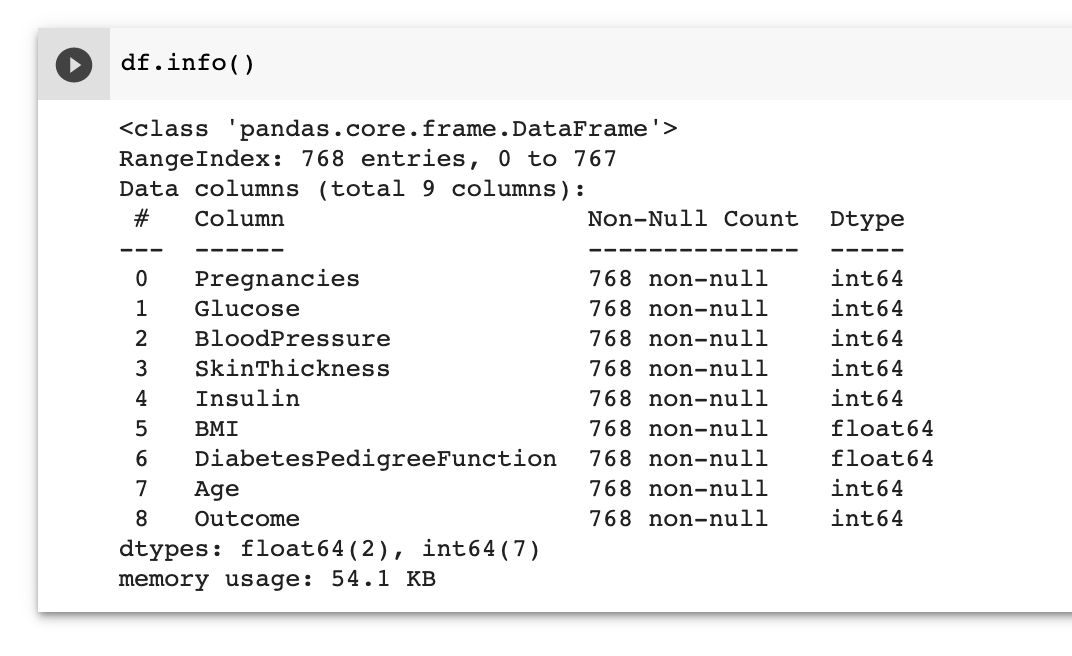
1. Upload the diabetes dataset to Google Colab



1. Import dataset



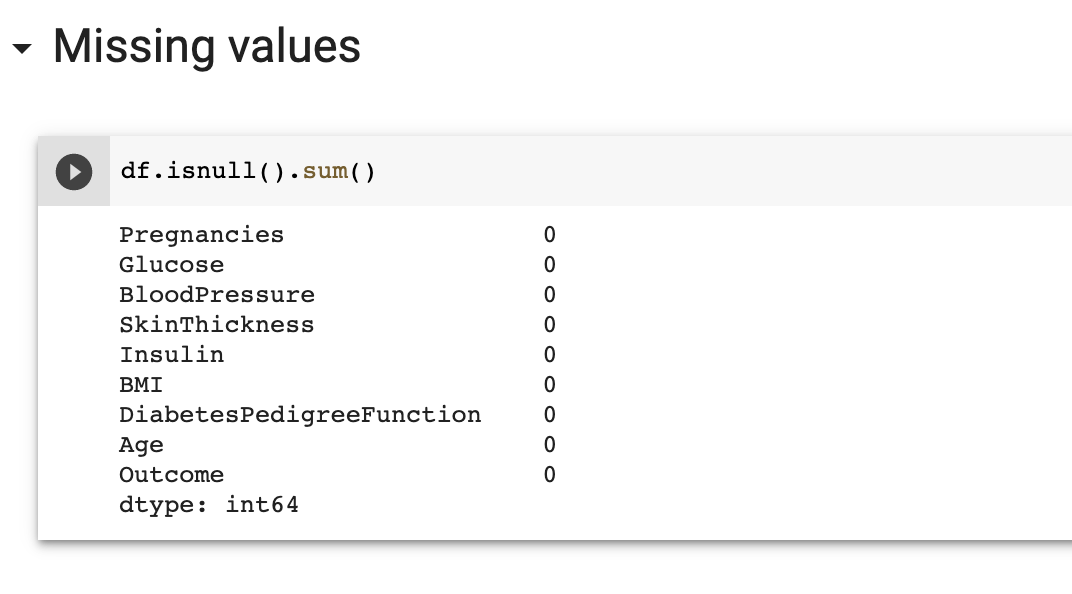
1. Data Summary



*We can see that all of the columns have numeric values. The “Outcome” column, which is the label column is already encoded as binary numeric data. There are 767 rows in this dataset which is relatively low, so we should leave more sample during the spliting step.*

**3. Check for Missing Value**

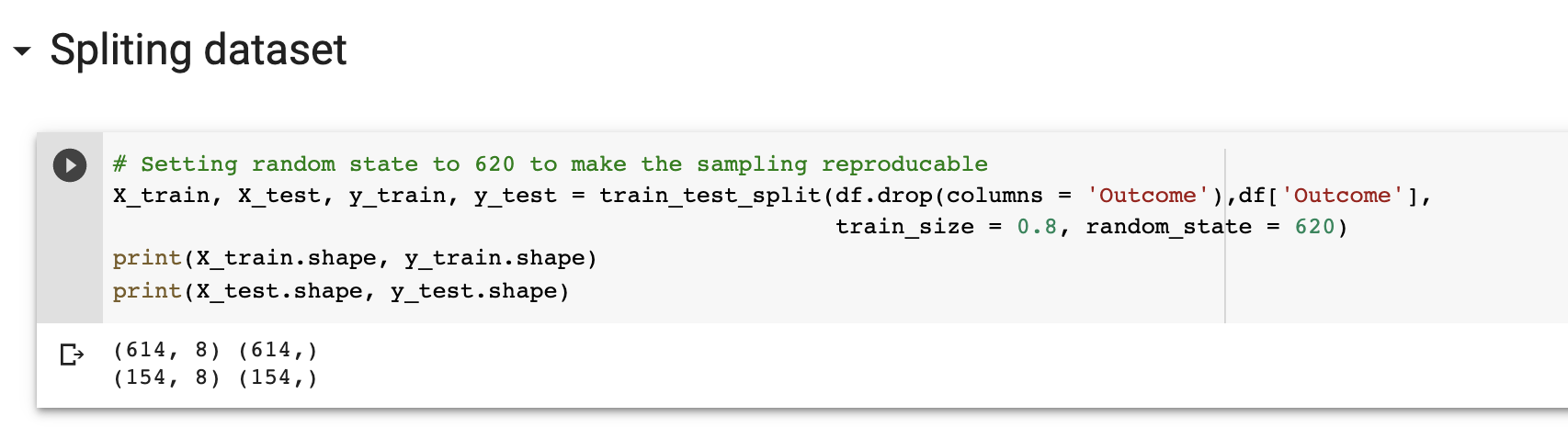
* Type the following code to check for null values



*Luckily, there is no missing value in this dataset*

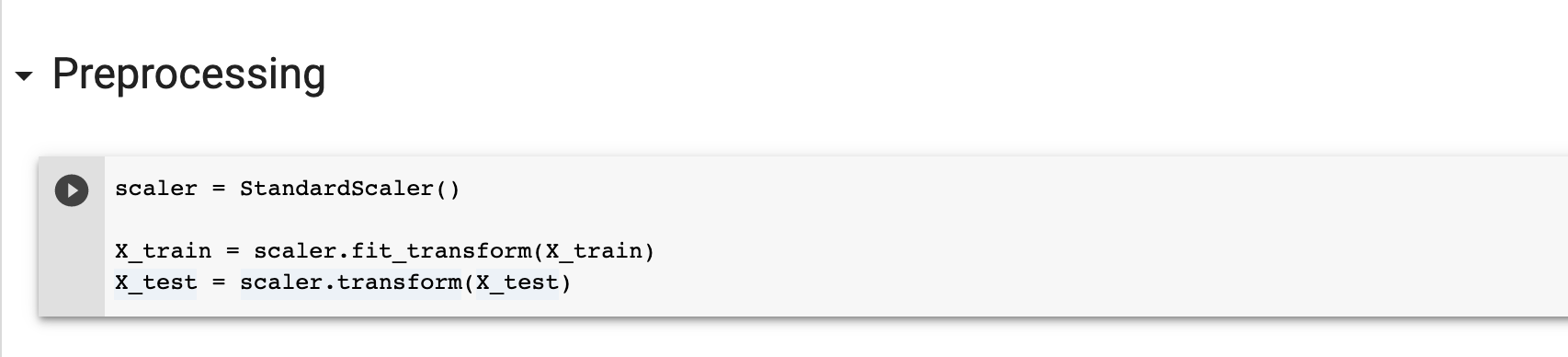
**4. Splitting data**

* Splitting data set leaving 80% of the data for training.



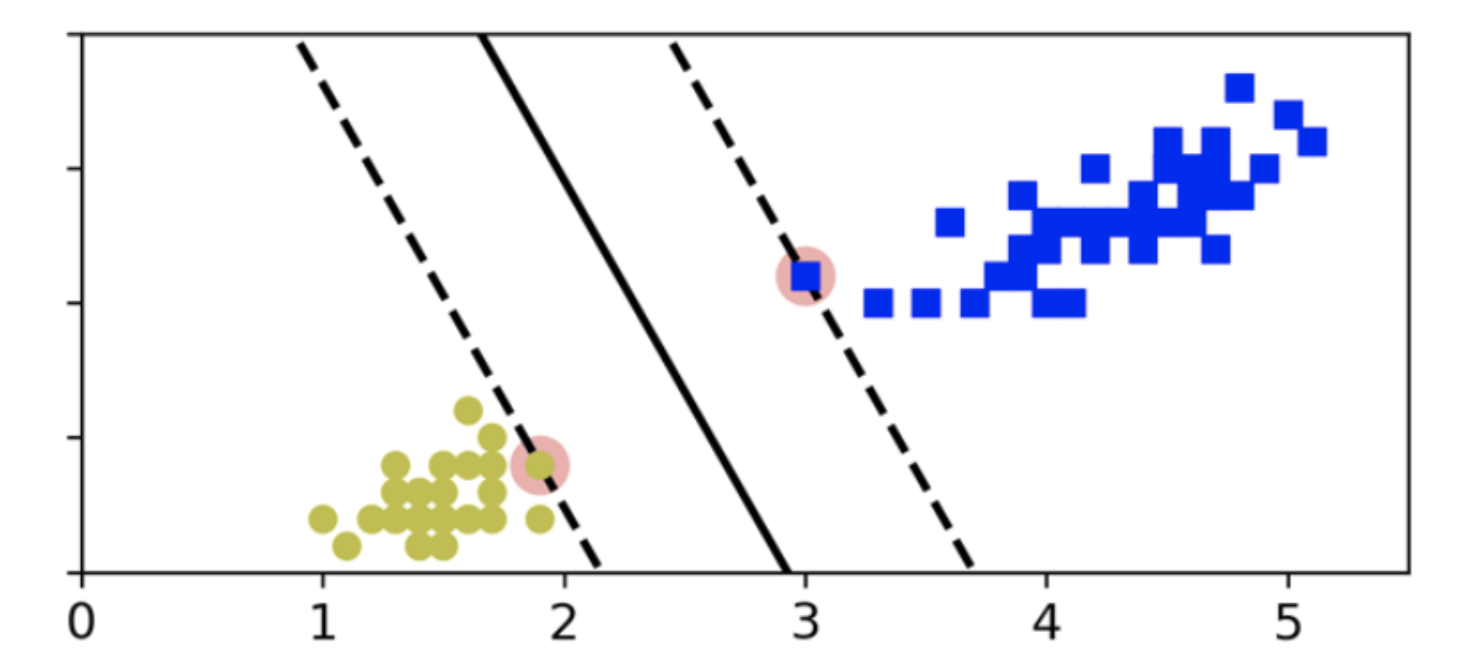
**5. Preprocessing**

* Rescale all input features

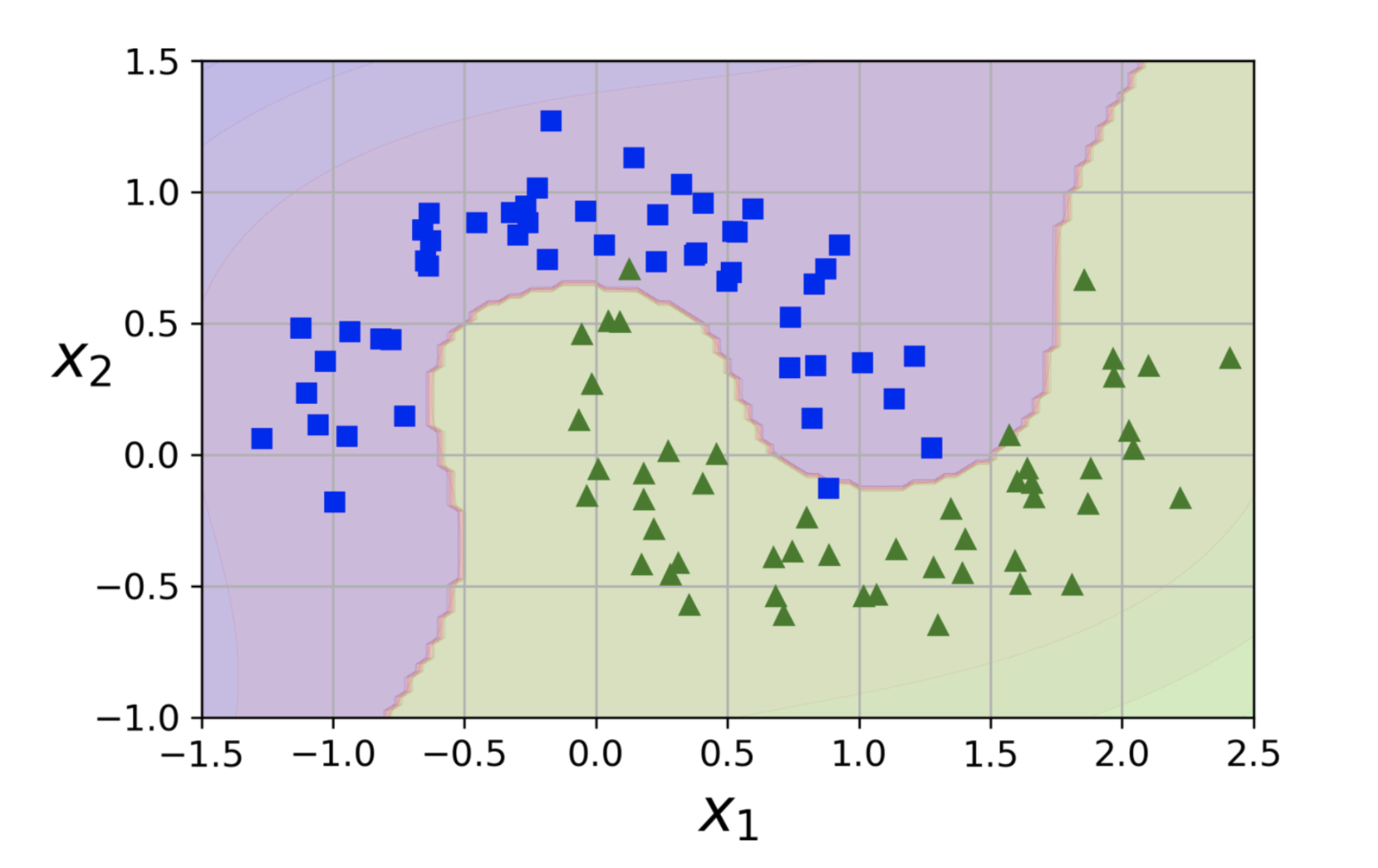


**6. Support Vector Machine**

Support Vector Machines are perhaps one of the most popular and talked about machine learning algorithms. They were extremely popular around the time they were developed in the 1990s and continue to be the go-to method for a high-performing algorithm with little tuning. This algorithm works by separating the 2 classes of the data with a decision boundary. Here’s the representation of the decision boundary in 2-dimensional space.



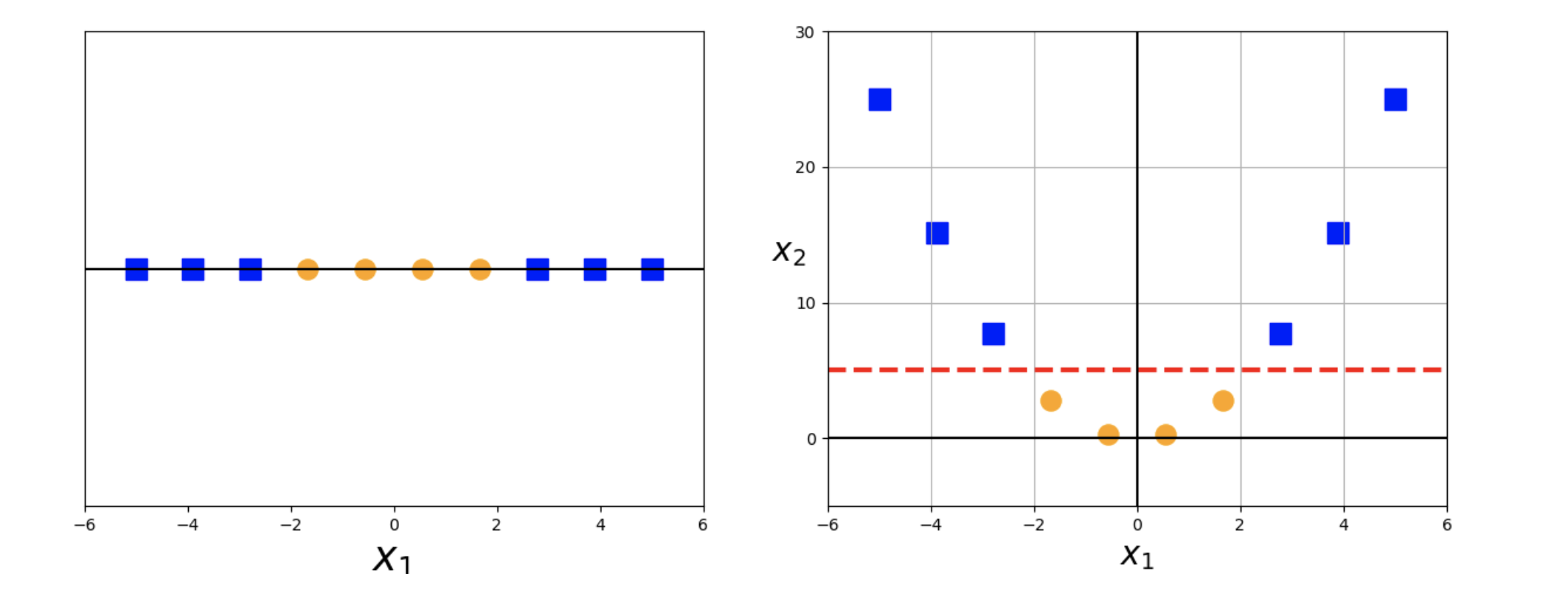
In real life the decision boundary is not always a line, it could take many forms, such as a squiggle.



In practice, The SVM algorithm is implemented using a kernel. The learning of the hyperplane in linear SVM is done by transforming the problem using some linear algebra. The 2 most commonly-used kernels are polynomial kernels and Radia basis function kernels

**6.1. Polynomial kernel**

The polynomial works by adding polynomial interaction between the features of the input space thus adding new dimensions which make it easier to setting the decision boundary. Here’s an example of transforming a 1-dimensional inputs space into 2 dimensional



Belows are the hyperparameter to keep in mind of polynomial kernels

***Hyperparameter:***

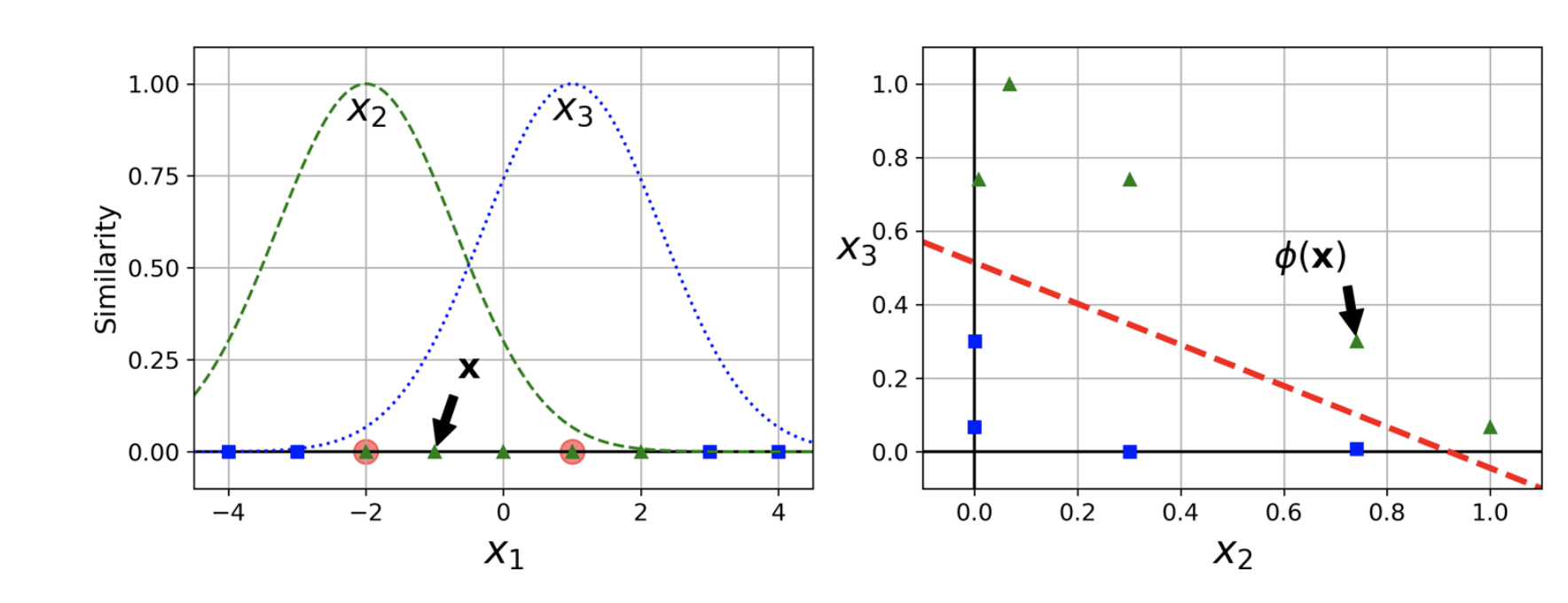
‘C’**:** This determines the weight of each error. The bigger C is, the less error the model is allowed to make ⇒ smaller margin. The smaller C is, the more error the model is allowed to make ⇒ Larger margin

‘degree’: The degree of the polynomial function, the higher the degree, the more curve there is to the data which might also lead to overfitting.

‘coef0’: Independent term in kernel function

**6.2 Radial basis function kernel**

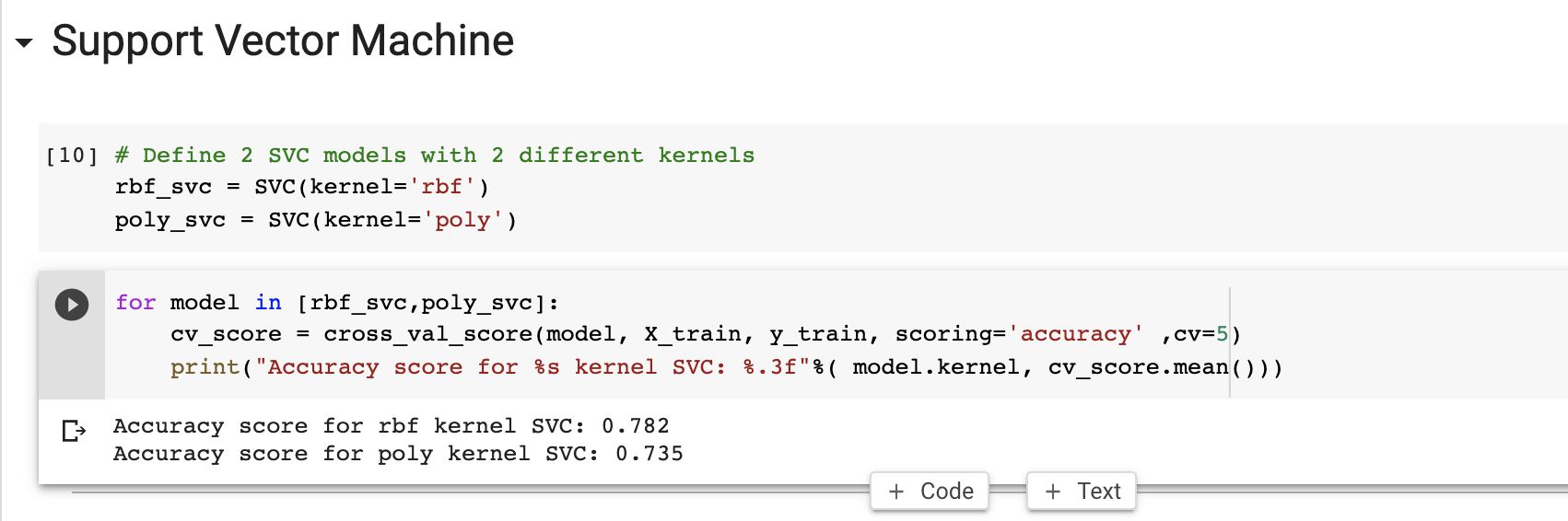
The rbf kernel is a bell-shaped function varying from 0 (very far away from the landmark) to 1 (at the landmark).



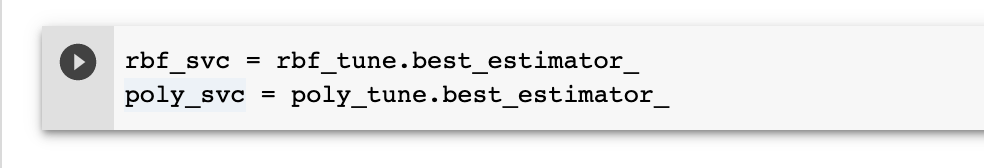
***Hyper parameter:***

‘C’**:** This determines the weight of each error. The bigger C is, the less error the model is allowed to make ⇒ smaller margin. The smaller C is, the more error the model is allowed to make ⇒ Larger margin

‘gamma’: Act as the regularization parameter of the model. Large gamma overfits, small gamma under-fits



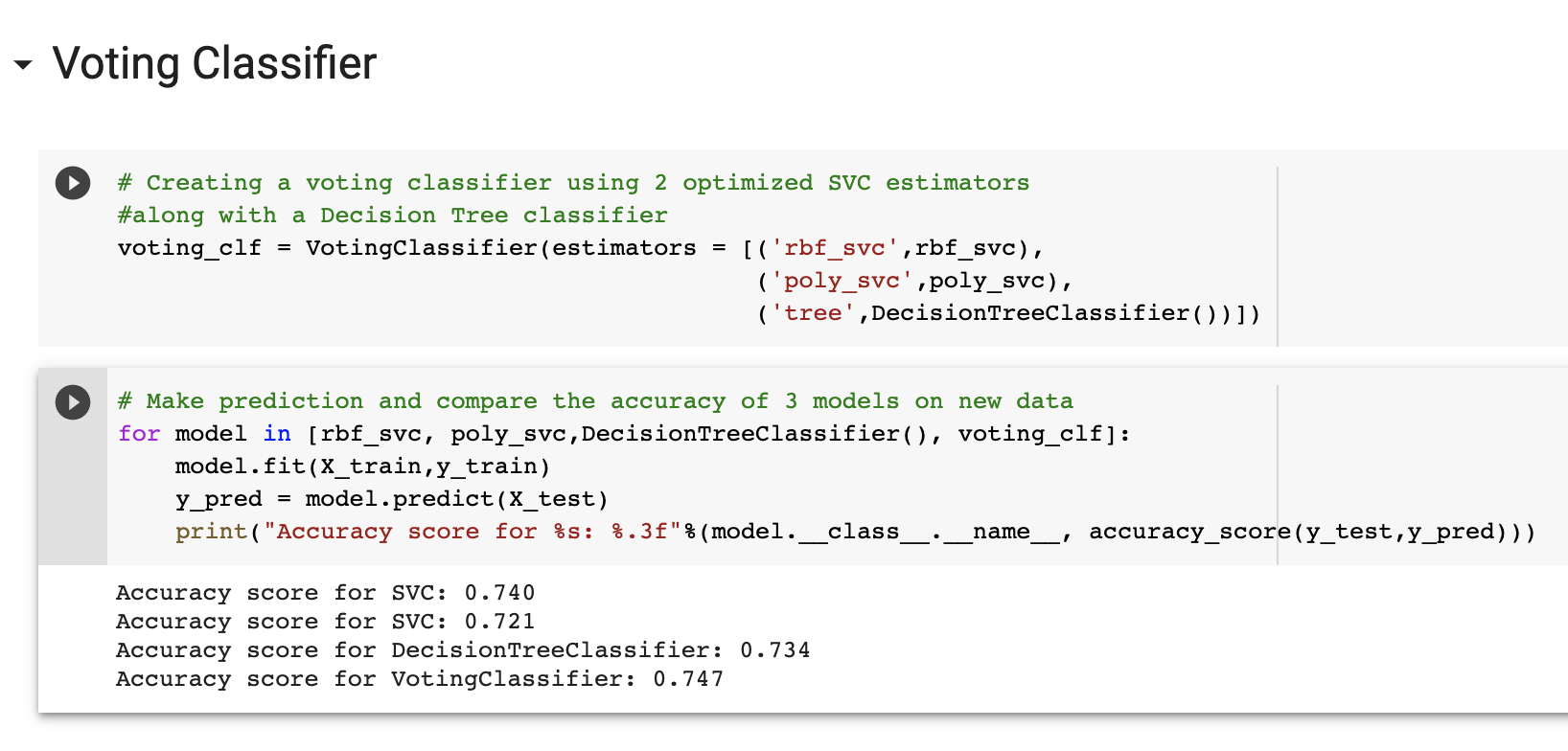




**7. Voting Classifier**

Voting classifier takes the prediction of multiple classifiers and choose the value with the most vote for prediction.

* Belows are the code to use 3 different models for a Voting classifier.



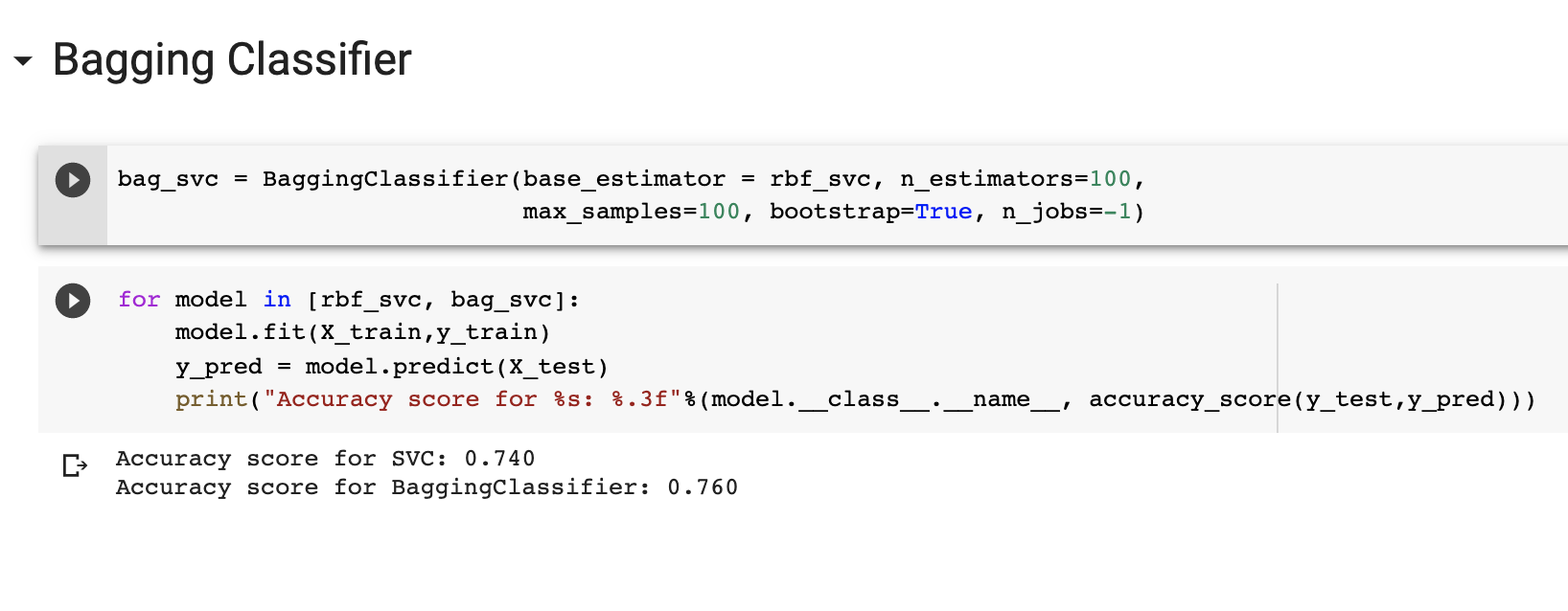
*The Voting classifier run 3 model simultaneously on the same dataset, get the results of the 3 models and use the one with the most vote. For example, if the both the rbf\_svc and the decision tree classifier predict the class as 0, but the poly\_svc predict the class as 1, the VotingClassifier will return 0 as the final result.*

**8. Bagging Classifier**

A limitation of Voting Classifier is that all the models are trained on the same dataset which make them dependent on one another. Independency among the models ensures that they are not making the same mistakes which impossible to achieve when they are all trained on the same dataset. Bagging tackles this limitation, by creating subset of the training dataset **with replacement**, each subset is different from one another, each model will be trained on each of these subsets. Similarly, to Voting classifier, Bagging returns the prediction with the most vote. However, the difference is that Bagging Classifier requires only 1 model but it can use this model multiple times.

More on Bagging [here](https://www.youtube.com/watch?v=2Mg8QD0F1dQ&ab_channel=Udacity)

* Creating a Bagging Classifier by training 100 rbf SVM classification models



*Bagging slightly improve the performance of the SVM classifier*

**Push Your Work to GitHub**

**Download the notebook from Colab**:

File -> SVM.ipynb

Move the downloaded file into your **Module4** working folder.

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

**Make sure the assignment files on the subfolder Module4 of hos04a\_YouGithubUserName folder, enter the following command to upload your work**:

>>>> git add .

>>>> git commit -m “Submission for HOS04”

>>>> git push origin master