**Part B**

Q1.

1.1 At the above BPTT function, it is observably that ‘self,ForwardPass(sample, slide)’ includes two parameter ‘sample’ and ‘slide’, the reasons will list below:

1. Processing sequence data: BPTT is designed for processing sequence data, especially in recurrent neural networks (RNN). In this case, 'sample' stands for a sample in a sequence, such as a time series of data or a sentence.
2. Processing time step: The slide parameter represents the time step or a specific moment in the sequence. In sequence processing the network needs to process a portion of the input data at each time step. For example, when working with a piece of text, each "slide" might represent a word in the sentence.

In conclusion, the two parameters ‘sample’ and ‘slide’ in self.ForwardPass(sample, slide) enable the algorithm to process the input correctly at each point in time in the d sequence and maintain continuity and dependency in time, which is critical for processing time series data or any form of sequence data.

**Reference:**

1. <https://www.wrike.com/project-management-guide/faq/what-is-forward-pass-in-project-management/>
2. <https://datascience.stackexchange.com/questions/116973/whats-a-forward-pass-in-a-neural-net>

1.2 In the provided K-Means algorithm Python code, the formation and refinement of clusters are explained through the following steps:

1. Initialization of Centroids (Lines 11-13):

- The `fit` function starts by selecting the first `k` points from the dataset as the initial centroids.

2. \*\*Iteration Process (Lines 15-50):

- The algorithm performs a loop for a maximum number of iterations, controlled by the `max\_iter` parameter(in line8, 11, 21). Each iteration updates the cluster assignments and the positions of the centroids.

3. \*\*Assigning Points to the Nearest Centroid (Lines 22-29):

- For each point (`featureset`) in the dataset, the Euclidean distance to each centroid is calculated. The point is then assigned to the cluster of the closest centroid.

4. \*\*Updating Centroid Positions (Lines 33-35):

- For each cluster, the average position of all its points is calculated, and this average is set as the new position of the centroid.

5. \*\*Checking for Convergence (Lines 37-46):

- The algorithm checks if the movement of all centroids is less than a set tolerance (`tol`). If the movement of all centroids is below this threshold, it indicates that the algorithm has converged, and the iteration ends.

6. \*\*Termination and Returning Centroids (Lines 48-49):

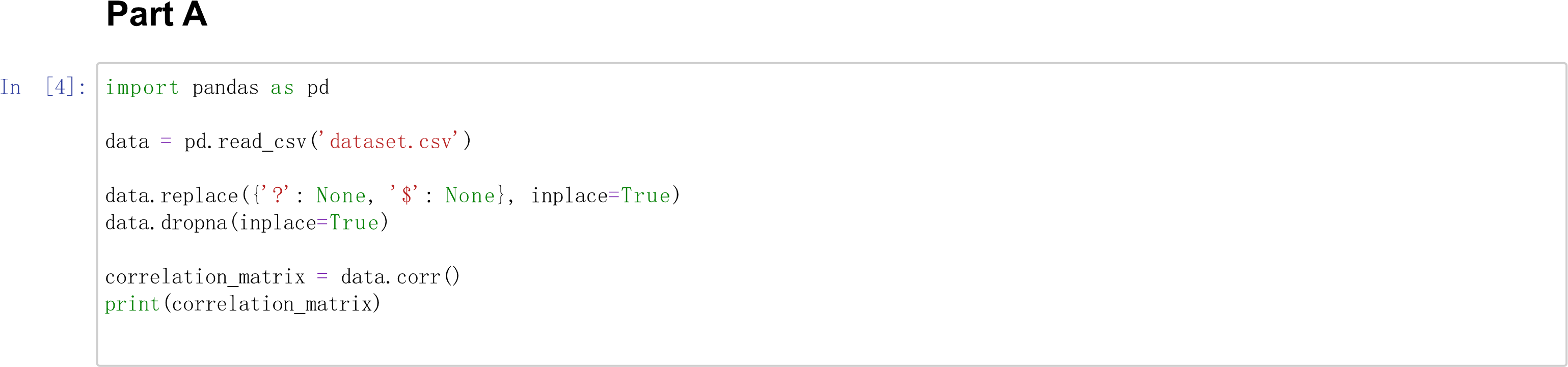
- Once the algorithm converges or reaches the maximum number of iterations, the iteration stops, and the current positions of the centroids are returned.

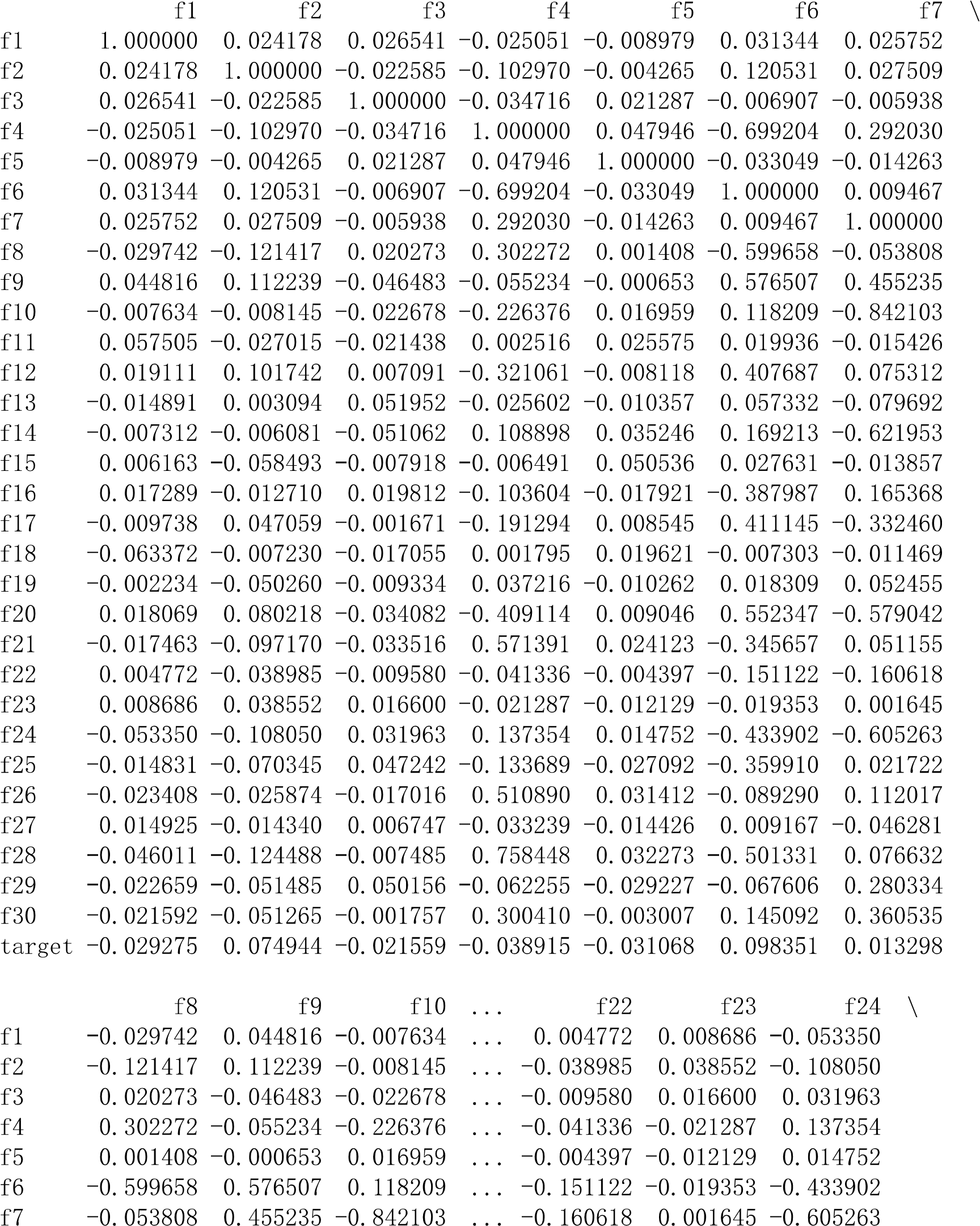
This process effectively involves iteratively reassigning data points to their nearest centroids and updating the positions of centroids based on these assignments. This repeated process ensures that over time, clusters gradually stabilize and ultimately converge to a relatively stable state, forming the final clusters.

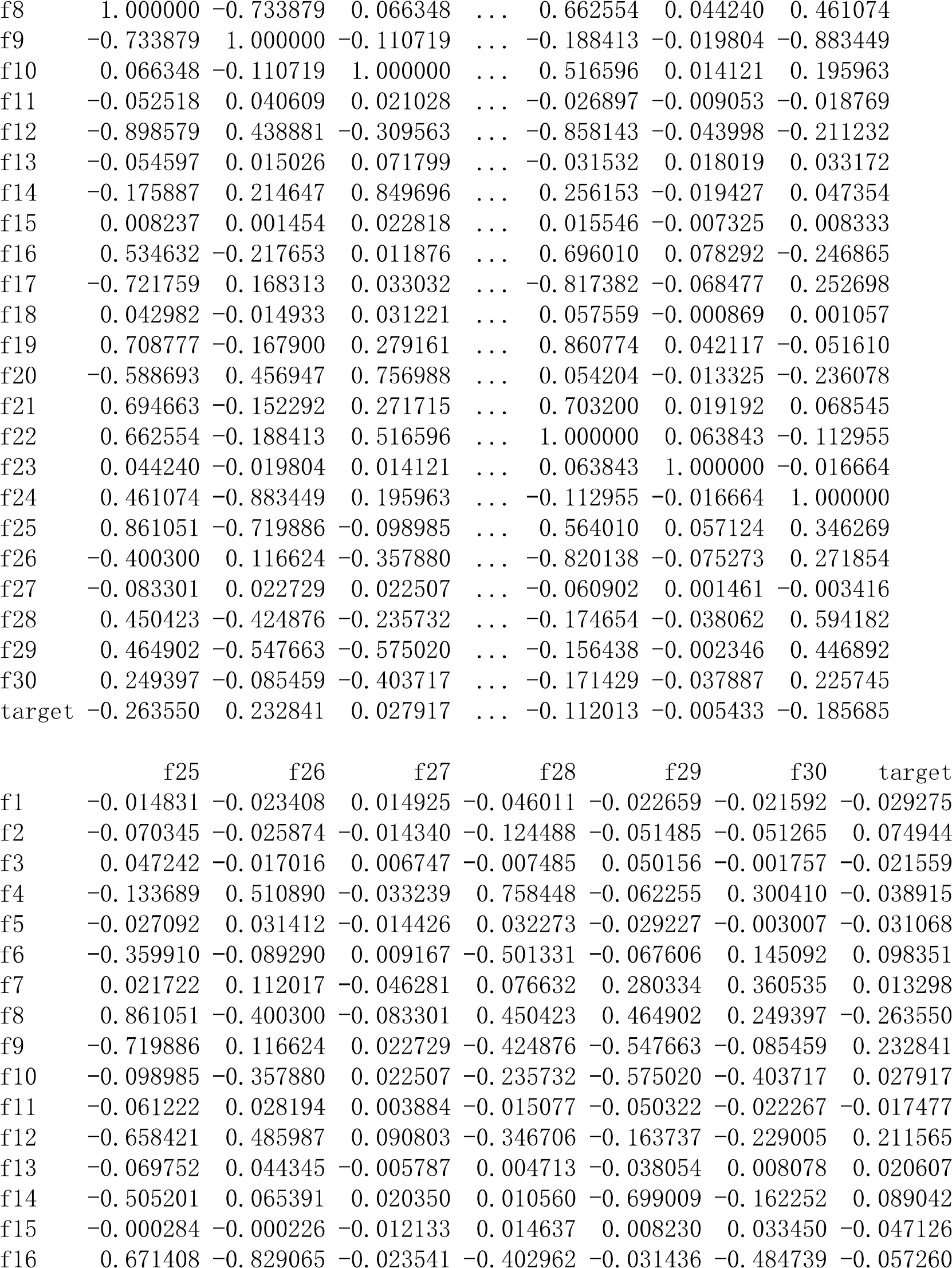
**Reference:**

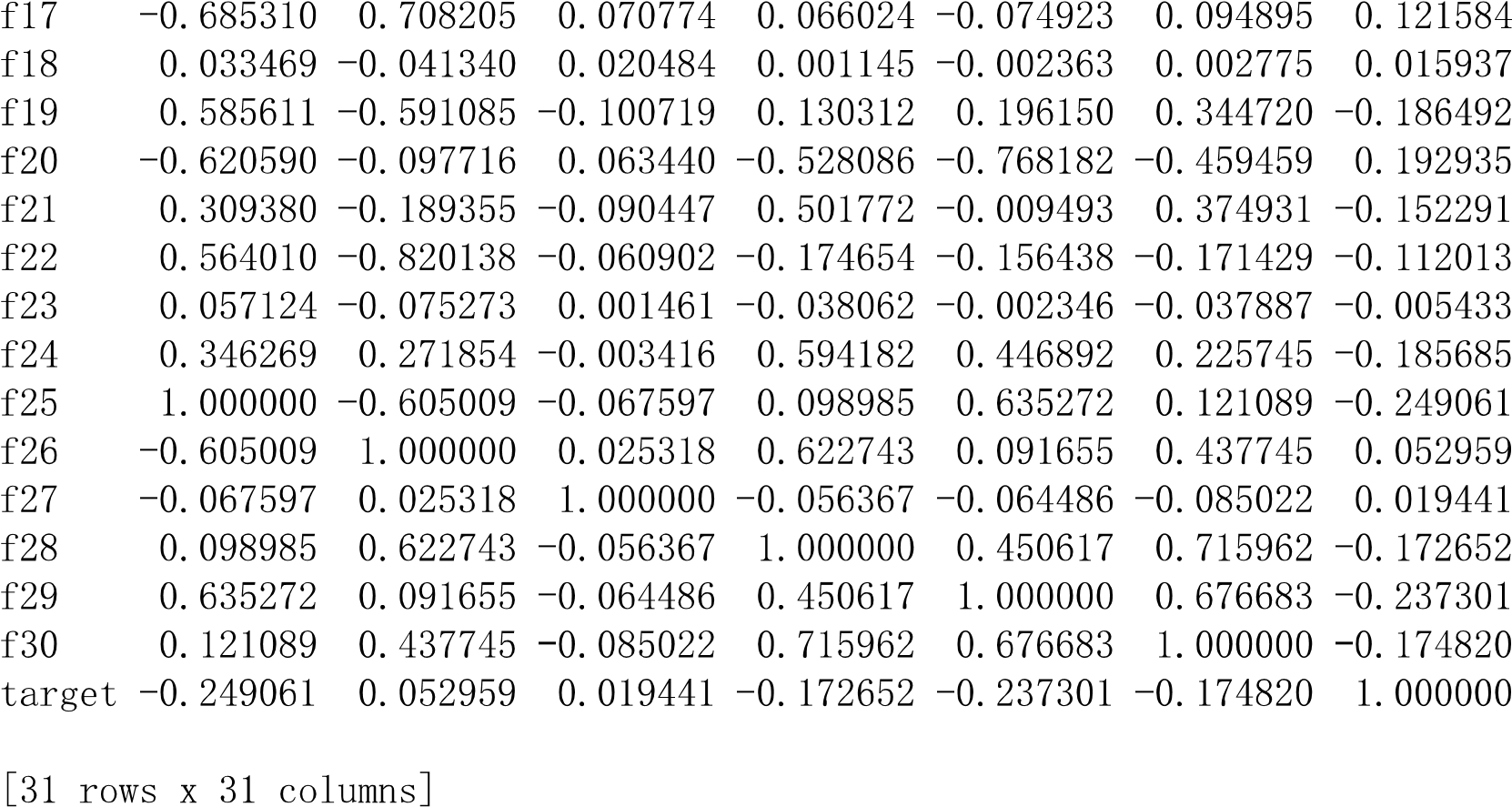
1. [**https://en.wikipedia.org/wiki/K-means\_clustering**](https://en.wikipedia.org/wiki/K-means_clustering)
2. [**https://www.analyticsvidhya.com/blog/2019/08/comprehensive-guide-k-means-clustering/**](https://www.analyticsvidhya.com/blog/2019/08/comprehensive-guide-k-means-clustering/)
3. [**https://towardsdatascience.com/understanding-k-means-clustering-in-machine-learning-6a6e67336aa1**](https://towardsdatascience.com/understanding-k-means-clustering-in-machine-learning-6a6e67336aa1)
4. [**https://scikit-learn.org/stable/modules/generated/sklearn.linear\_model.LogisticRegression.html**](https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html)
5. [**https://www.stata.com/manuals13/rmaximize.pdf**](https://www.stata.com/manuals13/rmaximize.pdf)

**PART C OPTION 1**

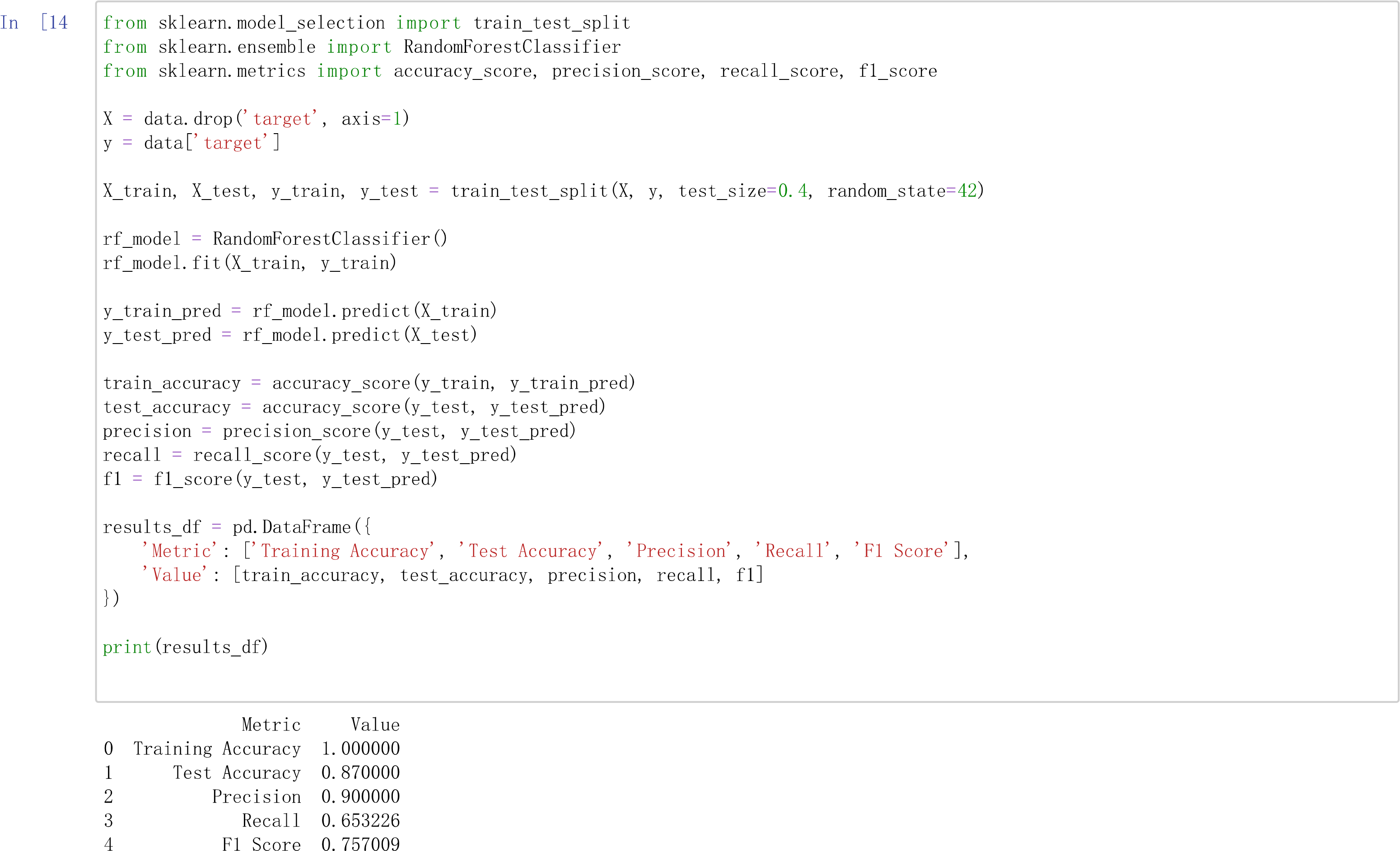


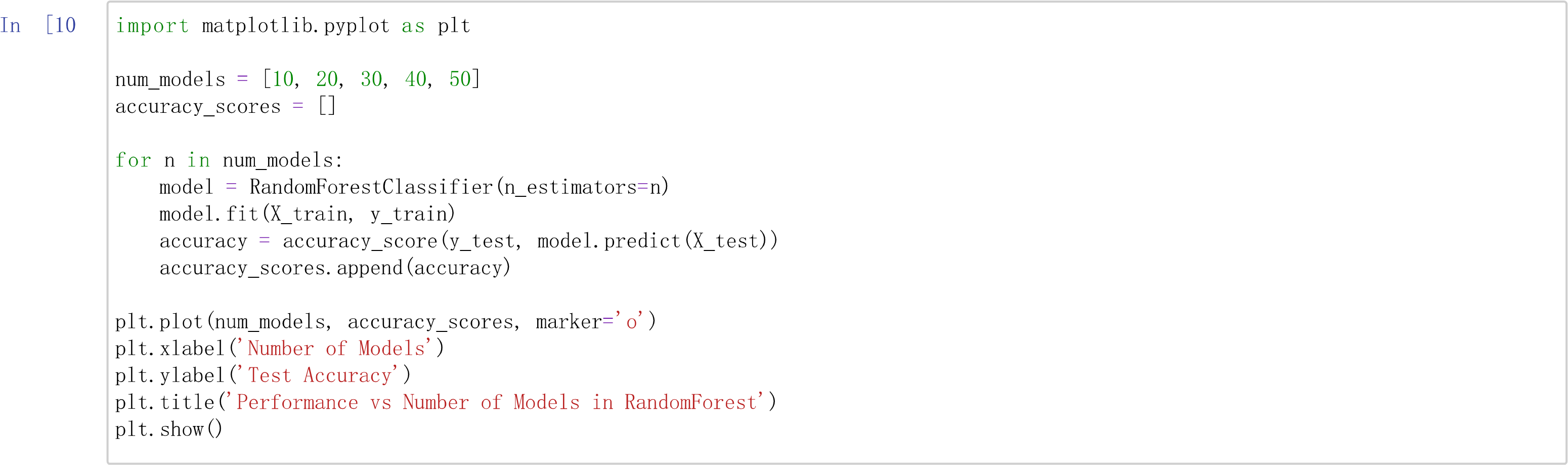


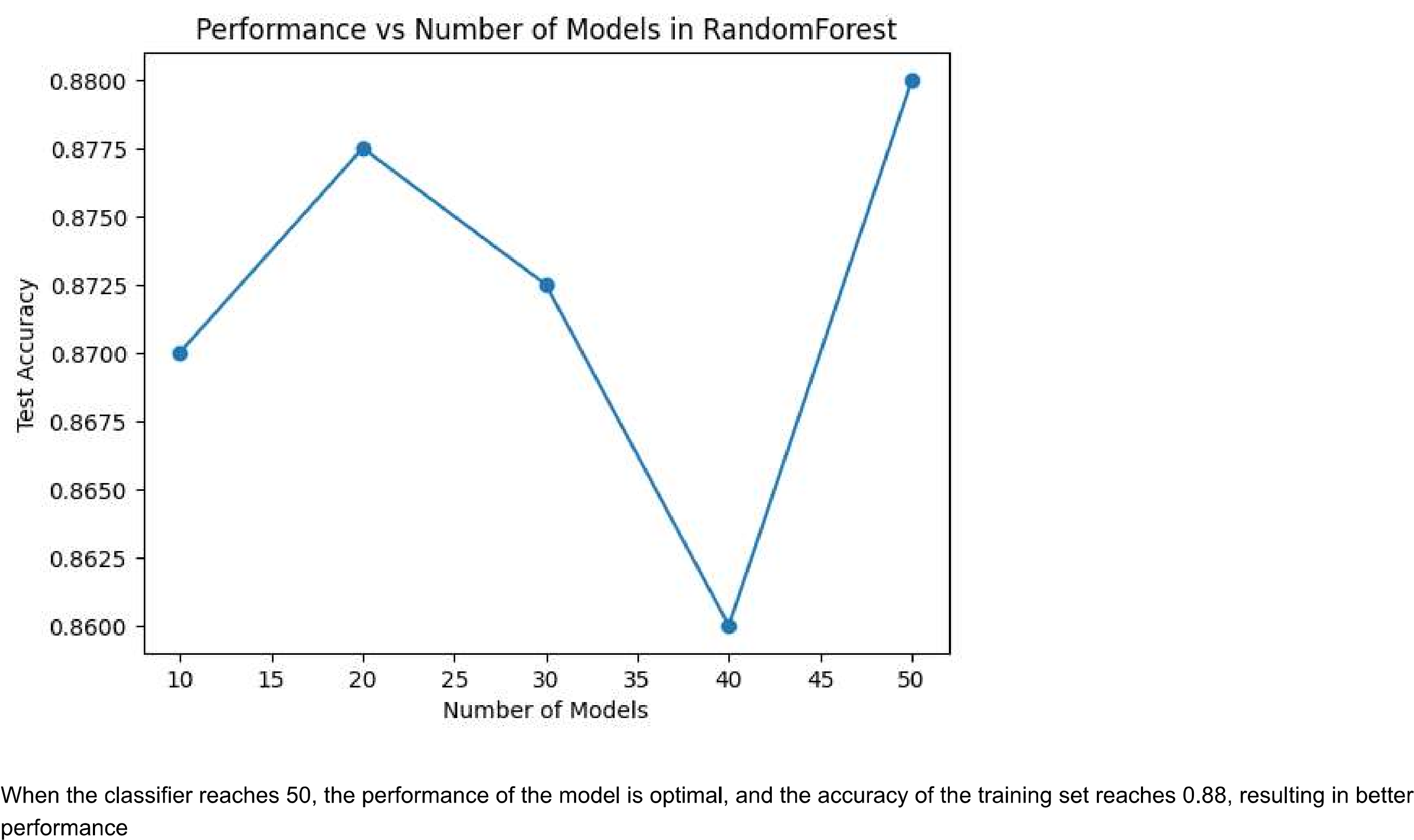


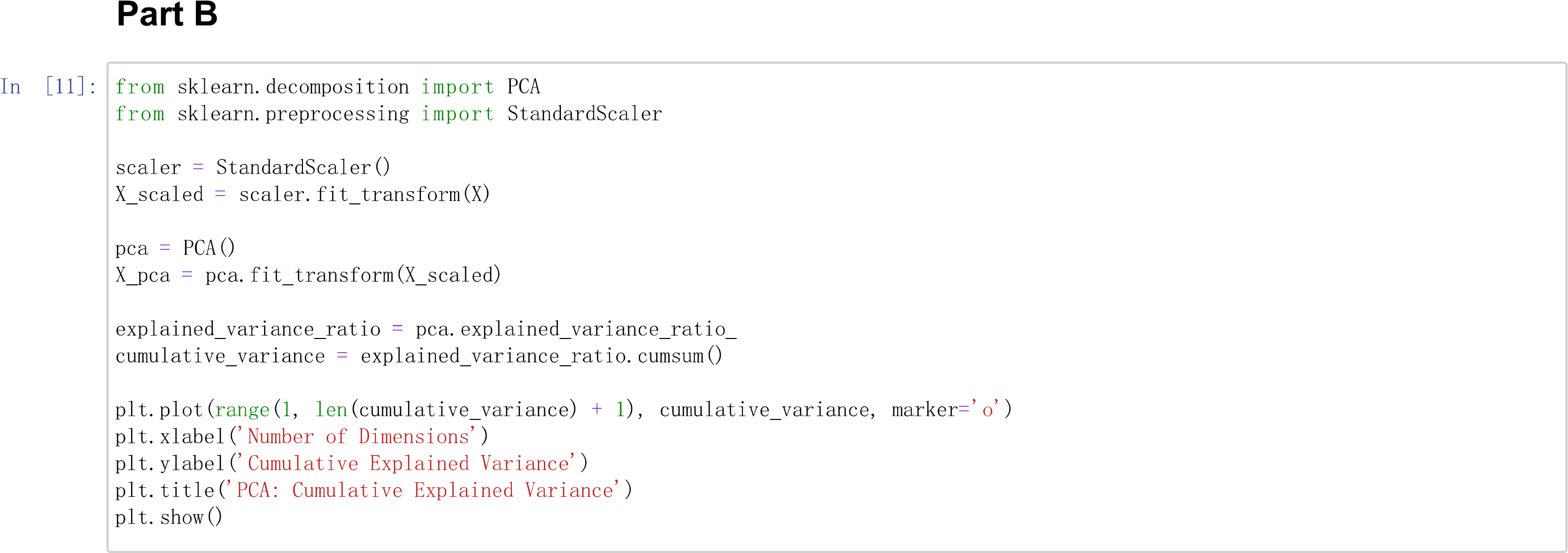


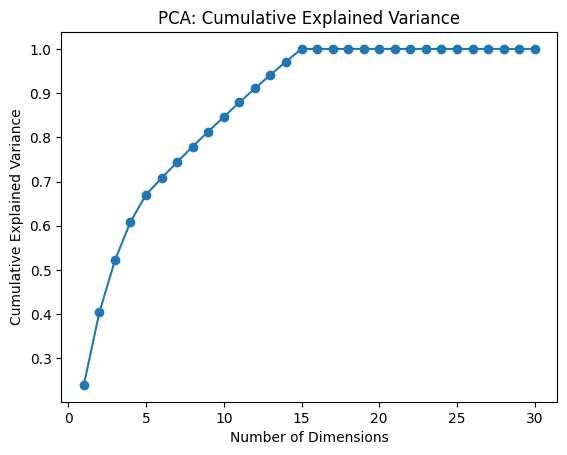












**Reference:**

1. <https://scikit-learn.org/stable/>
2. <https://pypi.org/project/scikit-learn/>
3. <https://www.digitalocean.com/community/tutorials/python-scikit-learn-tutorial>
4. <https://www.tutorialspoint.com/scikit_learn/scikit_learn_estimator_api.htm>
5. <https://mlflow.org/docs/latest/python_api/mlflow.sklearn.html>
6. <https://www.analyticsvidhya.com/blog/2015/01/scikit-learn-python-machine-learning-tool/>
7. <https://jakevdp.github.io/PythonDataScienceHandbook/05.02-introducing-scikit-learn.html>
8. <https://www.activestate.com/resources/quick-reads/what-is-scikit-learn-in-python/>
9. <https://www.geeksforgeeks.org/learning-model-building-scikit-learn-python-machine-learning-library/>
10. <https://en.wikipedia.org/wiki/Random_forest>
11. <https://en.wikipedia.org/wiki/Cluster_analysis>