NAME: Amish Faldu

NJIT UCID: af557

Email Address: af557@njit.edu

04/10/2024

Professor: Yasser Abduallah

CS 634 104 Data Mining

**Final term Project Report**

**Implementation and Code Usage**

**Supervised Machine Learning (Classification)**

**Abstract:**

In this project, I delve into the various machine learning algorithms for classifying the different dry bean. The algorithms are evaluated using F1-Score metric.

**Introduction:**

Supervised machine learning learns the patterns in the dataset and tries to best-fit the dataset so that it can accurately differentiate between items.

In this implementation, I’ve applied Random Forest, SVM, KNN and Conv1D algorithm to a dry bean dataset. Key steps in this process included:

* Loading the dataset from CSV files.
* Data analysis.
* Data pre-processing
* Model training

**Core Concepts and Principles:**

**Supervised Machine Learning:**

Supervised Machine learning is a part of Machine learning where the algorithm has access to the data points as well as label associated with that data point.

**Neuron:**

Neuron in neural network is machine representation of the biological neuron. The neuron accepts inputs, performs computation that is dot product of the input and weights, outputs activated computation.

**Neural Network:**

Neural network is an interconnection of the neurons.

**F1-Score:**

It is calculated based on model’s precision and recall. The precision and recall are calculated on TP, FP, TN and TP.

**K-Fold Cross Validation:**

This is a technique to divide the training dataset into train and validation dataset. This way we can train the model as well as check how well it is performing on the unseen data without exposing it to the test data.

**Project Workflow:**

Our project follows a structured workflow involving various stages and the application of the various machine learning algorithm**s.**

**Data Loading**

We begin by loading csv dry bean dataset. Each item contains of 16 attributes and a label. Dataset used in this project is Dry Bean dataset that can be found [here](https://archive.ics.uci.edu/dataset/602/dry+bean+dataset).

**Data Analysis**

We analyse data for any missing values, observe the attribute distribution, label distribution and correlations between different attributes.

**Data Pre-processing:**

To ensure clean and compatible data, we pre-process the dataset by converting string to numerical representation, normalizing dataset for SVM and Neural Networks.

**Machine Learning Algorithms**

**Random Forest**

**SVM**

**KNN**

**Conv1D Deep Learning**

**Results and Evaluation:**

The machine learning’s accuracy and efficiency are evaluated based on performance measures such as F1-score, precision, recall and confusion metrics.

**Conclusion:**

In conclusion, this project demonstrates the application and evaluation of supervised data mining (Machine Learning) concepts and methods on dry bean classification dataset.

**Directory Structure:**

-------- dataset/ => This contains all the data files needed to run the code

-------- models/ => This is the saved neural net model.

-------- docs/ => This contains images for documentation

-------- .gitignore => Ignore files / folders to include in git history

-------- main.ipynb => This is the main Jupyter notebook that contains the code to run brute force and library’s algorithm

-------- Finalterm-Project-Report.pdf => Obviously, project report

-------- README.md => This contains all the instructions to setup the project locally

-------- requirements.txt => This contains all the packages to run the code

**Steps to run the program:**

1. Make sure that you’ve python installed on your machine. To download and install Python, go to <https://www.python.org/>. To check if you’ve python installed, run `python3 --version`.

**NOTE - I have used python version 3.11.7 to run the program**

1. Create a virtual environment using command `python3 -m venv .venv`.
2. Activate the virtual environment using command `source .venv/bin/activate`.
3. Install all the packages in **requirements.txt** using `pip install -r requirements.txt`.
4. Select existing `.venv` python environment as the kernel for the `main.ipynb` Jupyter notebook.
5. Run cells in `main.ipynb` Jupyter notebook.

**Screenshots:**

**Pre-requisite steps**

Step – 1



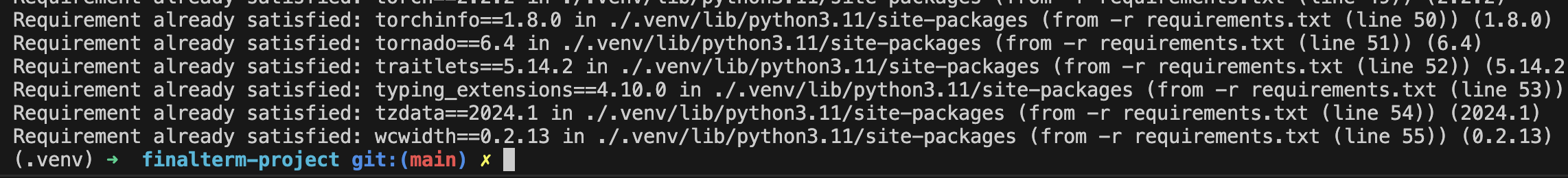
Step – 2



Step – 3



Step – 4



Step – 5

Click on the top-right of the notebook where it would say “Select Kernel”. By clicking it will prompt you to select a env.

A screenshot of a computer

Description automatically generated

Now select the virtual we created in step – 2 as the environment. After that, it would show you the selected kernel in top-right section.

A screenshot of a computer

Description automatically generated

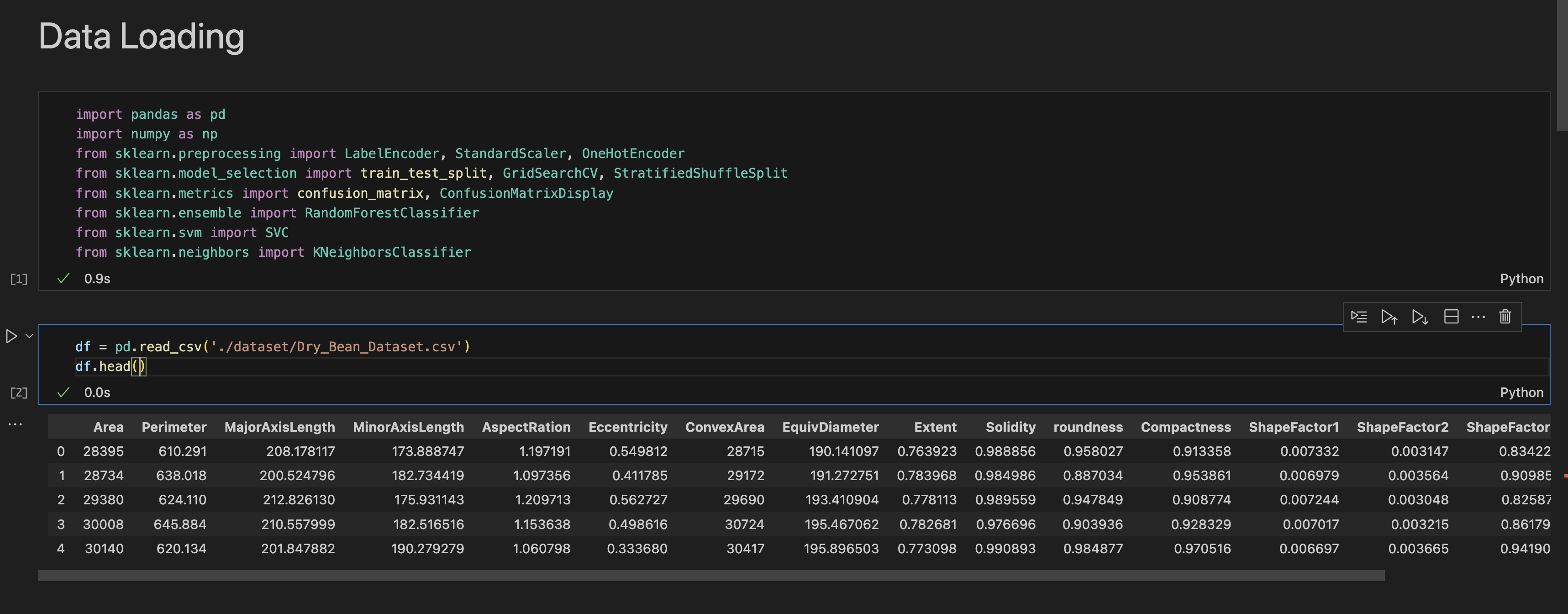
Step – 6

A screenshot of a computer

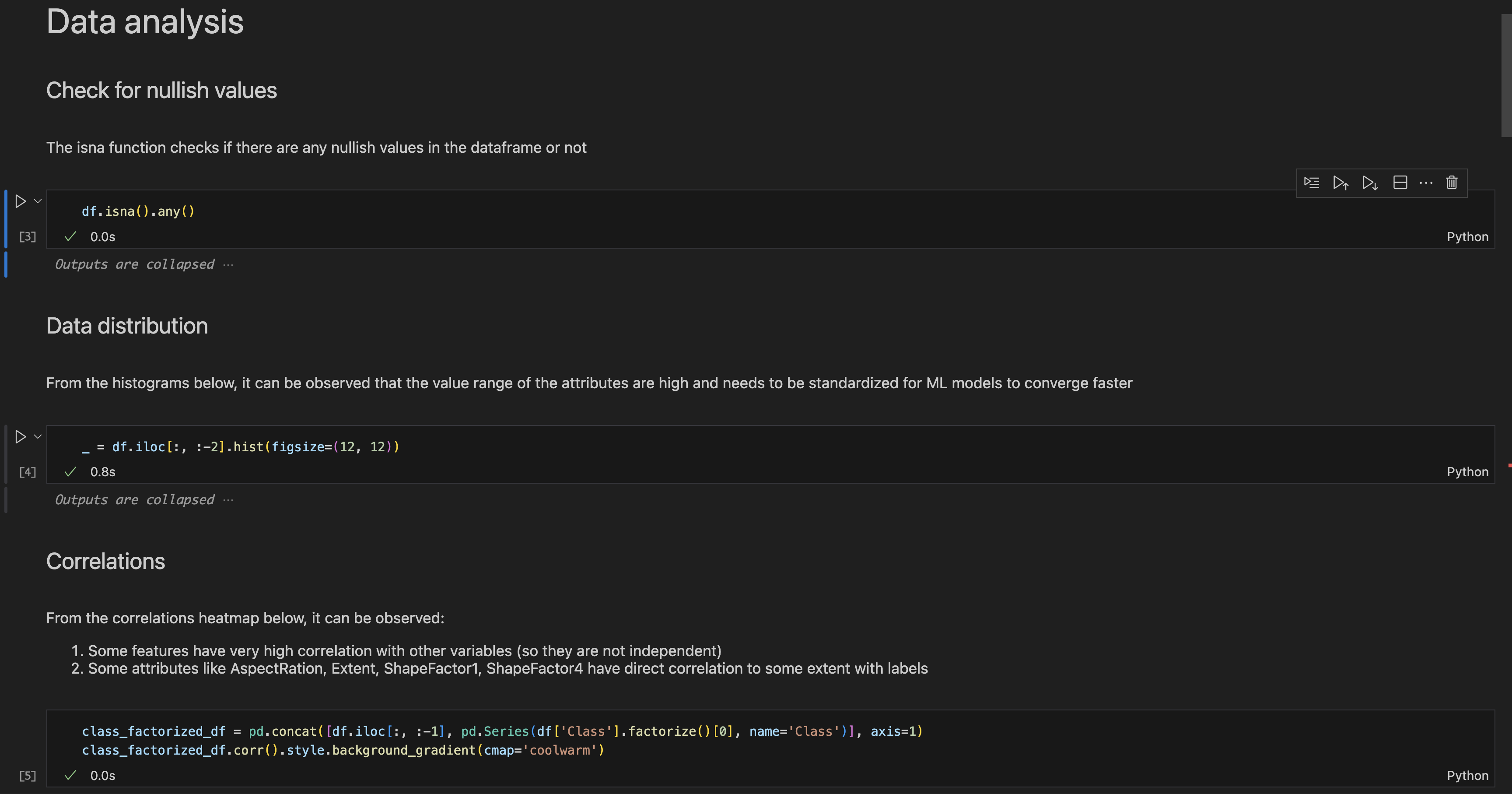
Description automatically generated

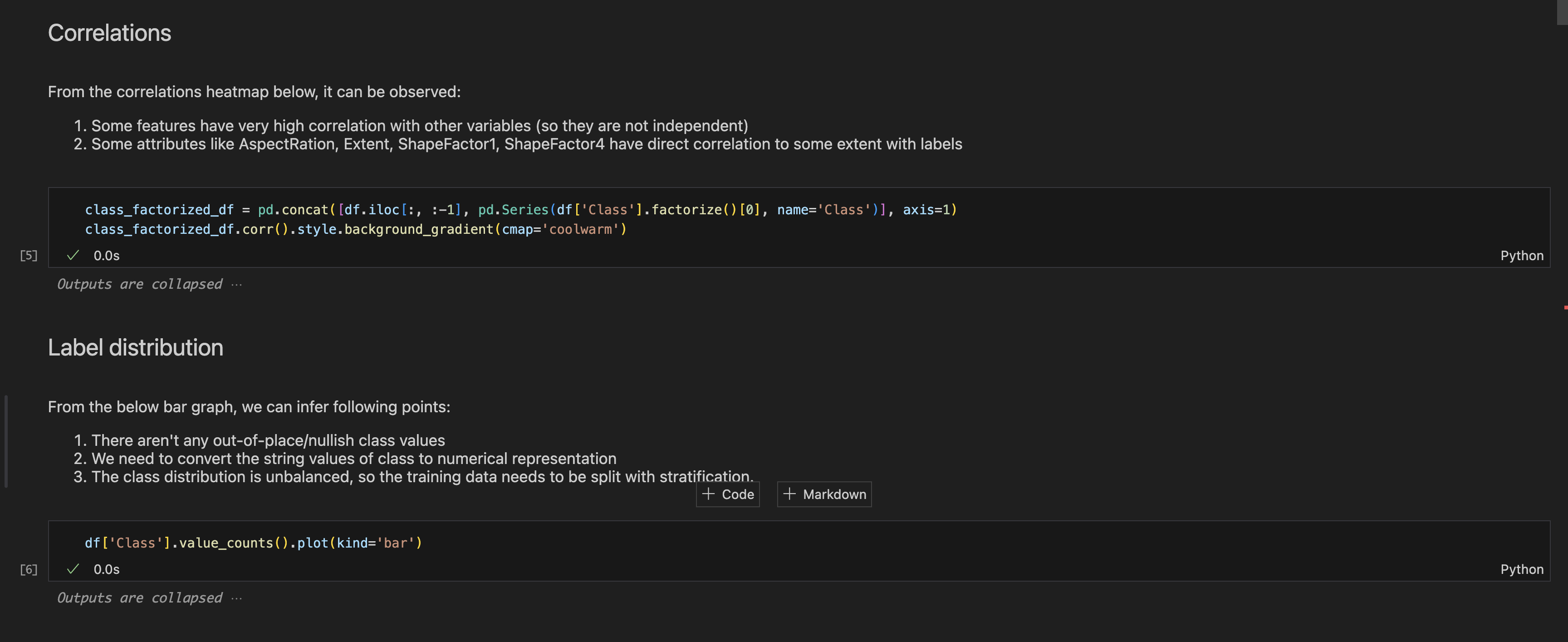
Below are screenshots of the running code from Jupyter notebook:

Data loading

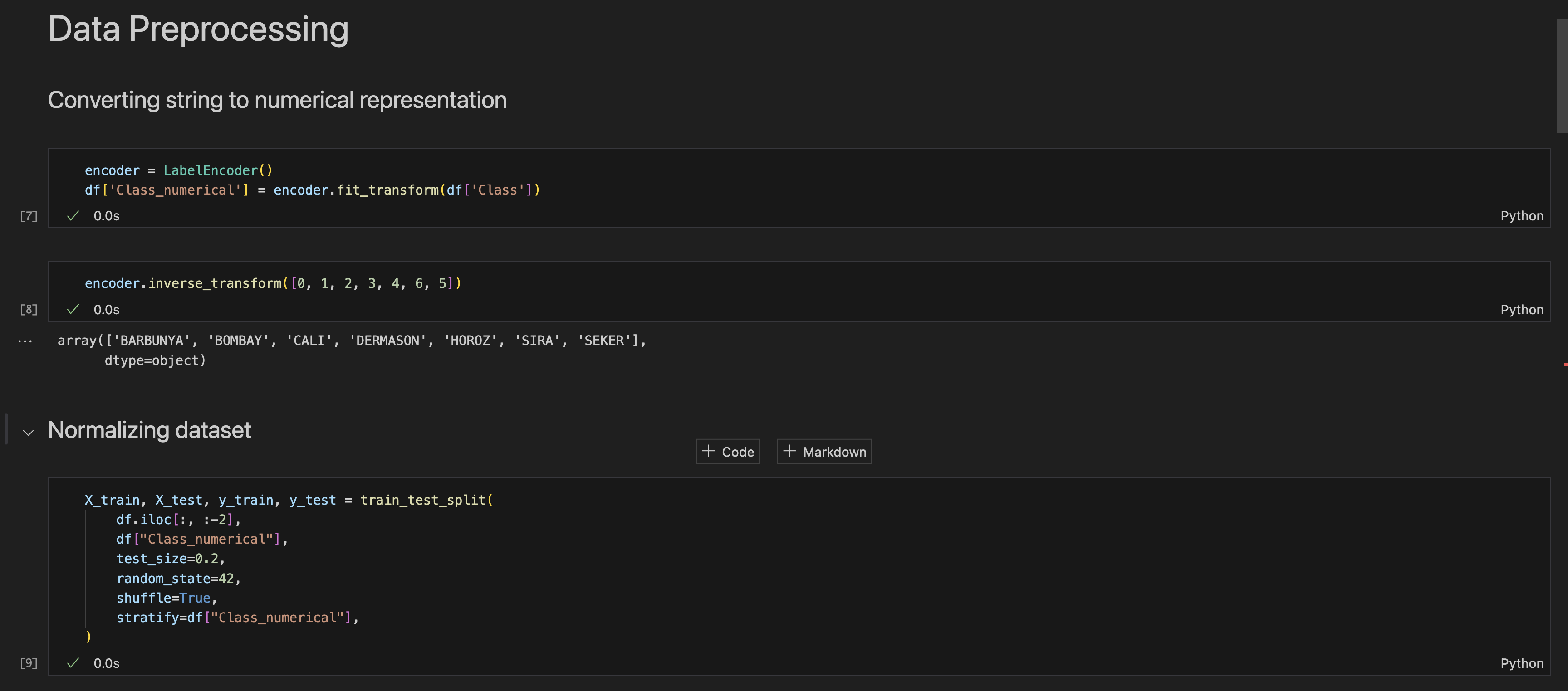


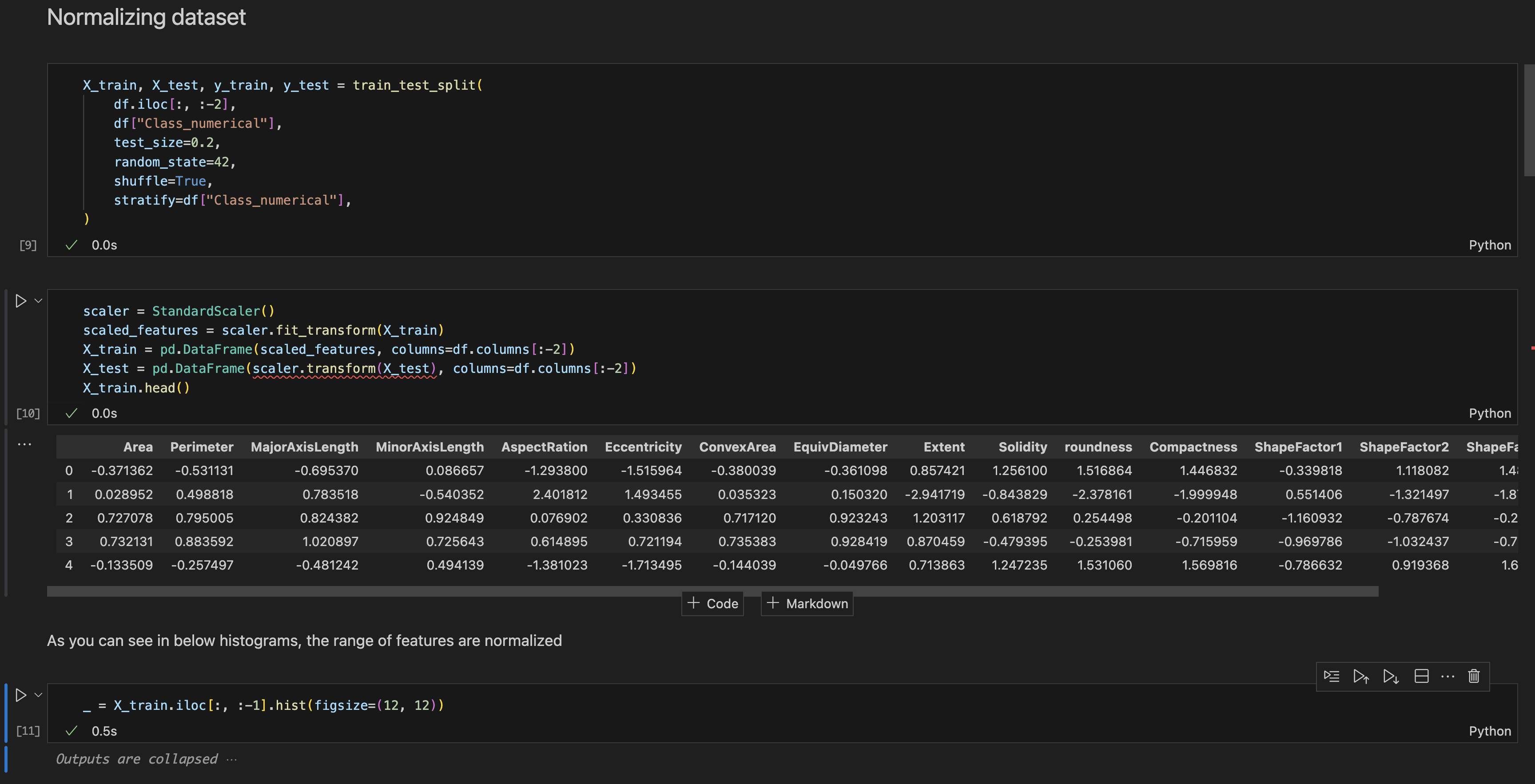
Data analysis





Data pre-processing

****



Model Training

Random Forest

A screenshot of a computer

Description automatically generated

A screenshot of a computer program

Description automatically generated

SVM

A screenshot of a computer

Description automatically generated

A screenshot of a computer program

Description automatically generated

KNN

A screenshot of a computer

Description automatically generated

A computer screen shot of a black screen

Description automatically generated

A black screen with text on it

Description automatically generated

Conv1D

A screenshot of a computer

Description automatically generated

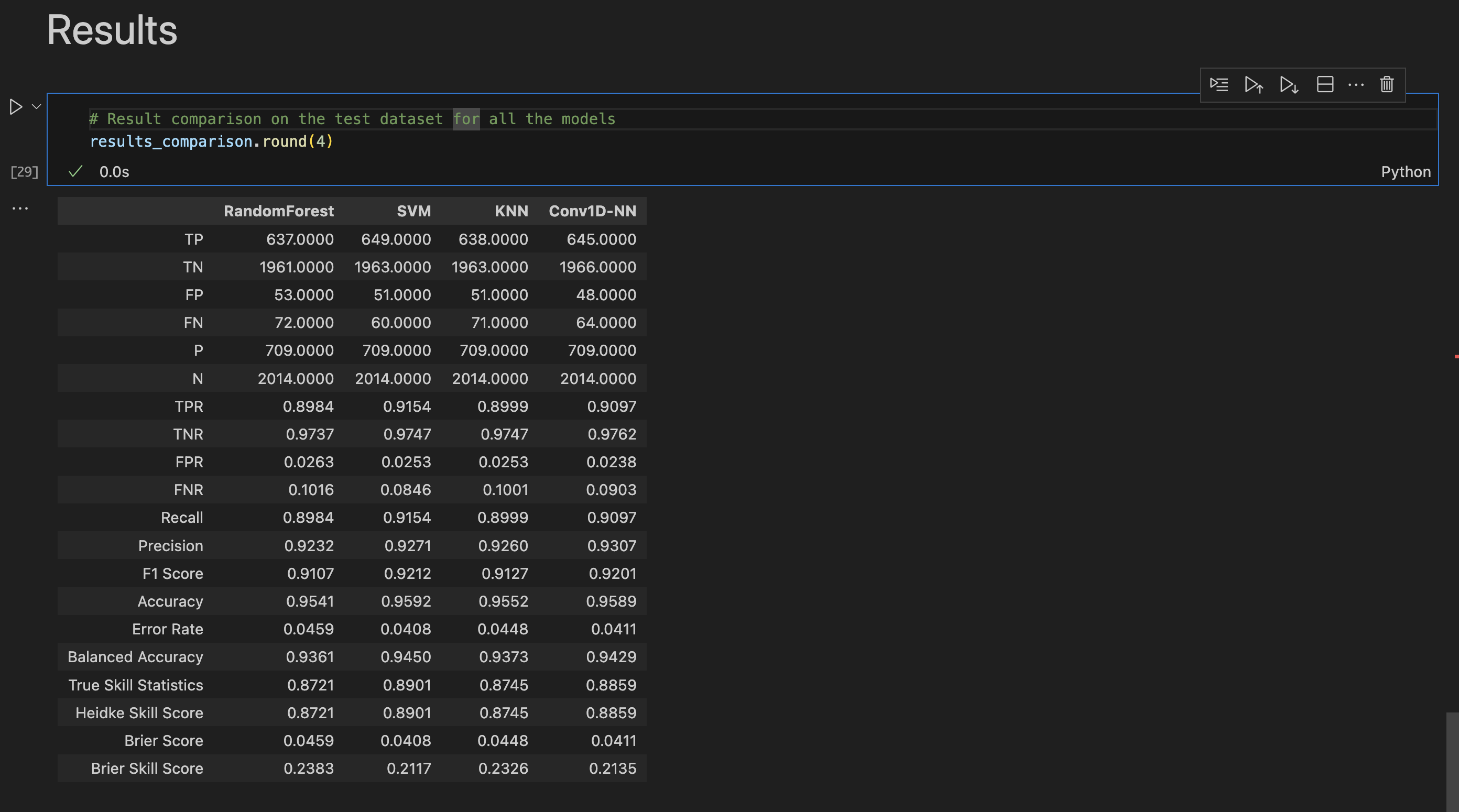
A screenshot of a computer program

Description automatically generated

A black screen with text on it

Description automatically generated

Results



Conv-1D and SVM performed better overall than other algorithms. This is because neural networks has ability to determine and fit complex relationships among different attributes and label. SVM with RBF kernel function separates data into higher dimensions, that’s why it was able to predict accurately.

KNN and Random Forest also did good job and were very close to the Conv-1D neural network and SVM, this is because the correlation between attributes and labels were proportional.

**Link to Git Repository:** <https://github.com/amishfaldu-njit/cs-634104-final-term-project>

**Dataset can be found at:** <https://archive.ics.uci.edu/dataset/602/dry+bean+dataset>