***Rapport de projet de Synthèse 1***

*Stock prediction: Time series analysis with web implementation*

*Elaboré par :*

*Mohamed Toumi et Roua Grissa*

***Summary***

List of figures List of tables

List of abbreviations General Introduction

Chapter 1: choosing and manipulating the data

* introduction
* importing data
* understanding and preparing the data
* conclusion

Chapter 2: training and testing the data

* introduction
* training the model
* testing the model
* conclusion

Chapter 3: App implementation

***List Of Figures***

***List Of Tables***

***List Of abbreviations***

***General Introduction***

Stock market is considered chaotic, complex, volatile. It is the market where organized issuance and trading of Stocks take place either through trading or over the counter in electronic or physical form. Undoubtedly, its prediction is one of the most challenging tasks in time series forecasting.

So researchers were enforced to discover a technique which can estimate the effect of this vagueness to the stream of share prices. They analyses various statistical models, Artificial Neural Networks and data mining techniques which are analogous to nonparametric, nonlinear, regression models. They find the potentiality to discriminate mysterious and buried patterns in data which can be very effectual for share market prediction. There are several approaches that have been applied in order to predict financial product value such as Statistical Analysis, Data mining, Neural Networks and Genetic Algorithms.

In this project, we focused on the Statistical Analysis Approach. We tried both the ARIMA and the GRU model to train our data and obtain the best possible prediction.

The report is organized into three chapters, in the first chapter we introduce our code and how it is prepared to be tested. In the second chapter we use our model for training and testing to obtain the best accuracy . In the last chapter, we made a web implementation .

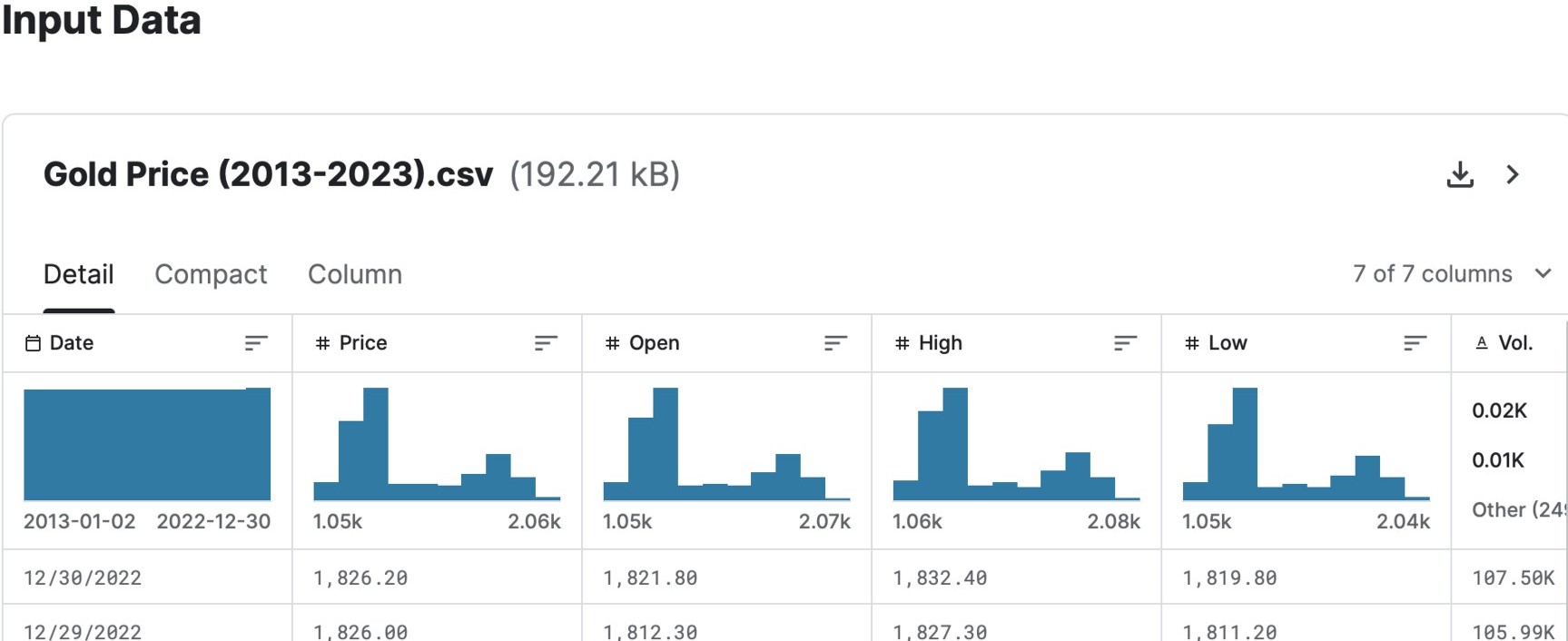
# Chapter 1: choosing and manipulating the data

## Introduction:

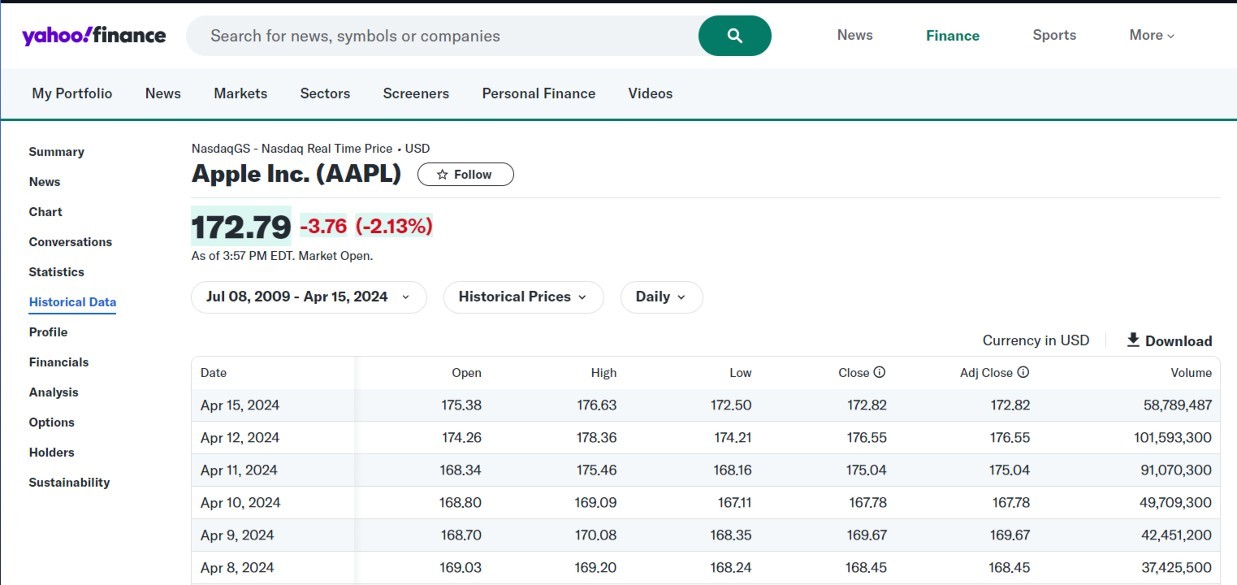
In this project , we are going to build a time series model to predict the future price of gold and apple stock, which can be very useful for traders. For this purpose, we use historical price data. In the following steps we are only using one stock data andone model because the steps of manipulating data are the same in each model.

## Importing Data:

In this section , we downloded historical gold data from Kaggle.



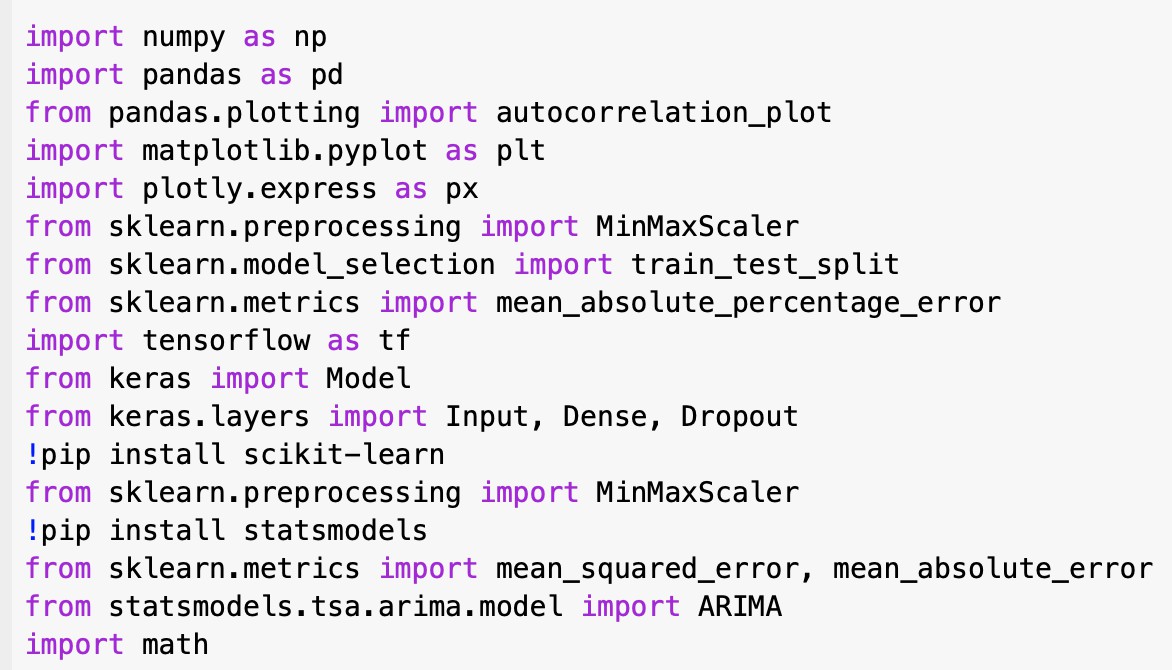
## Figure.1

[](https://finance.yahoo.com/quote/AAPL/history/?guccounter=1&guce_referrer=aHR0cHM6Ly93d3cuZ29vZ2xlLmNvbS8&guce_referrer_sig=AQAAACShK6XMsV3jfE9SaRqboHFz4IIBSrX5XJ5_0lh977id_ljnNIv3cdeNDEJaXZZFoJQkF7dcADZ3z8gYzdhEXRxfYSkacS0v7SnJzAxqBR9salGDloSixujJZ_8D_O4IK36JNG2Ha1D6RwjPO-_ovpr1uT7yU-yfI6YCYXXVQEvA&period1=1247011200&period2=1713210589)As for the histrotical price for apple stocks , we got it from Yahoo finance:

## Figure.2

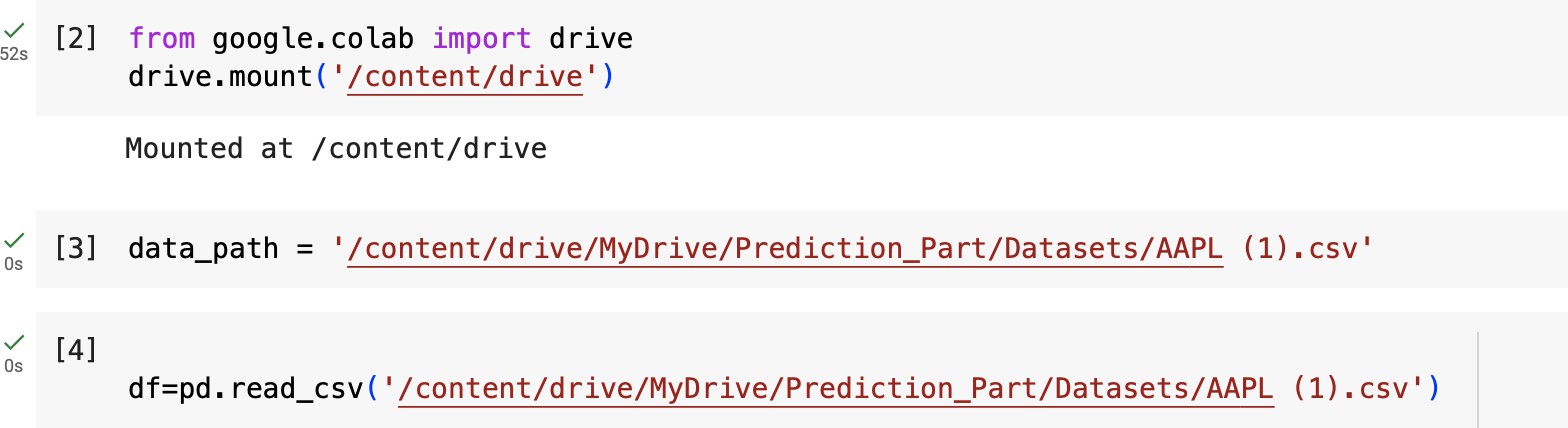
1. **understanding and preparing the data :**

First of all,we imported all libraries and modules that we are going to use:



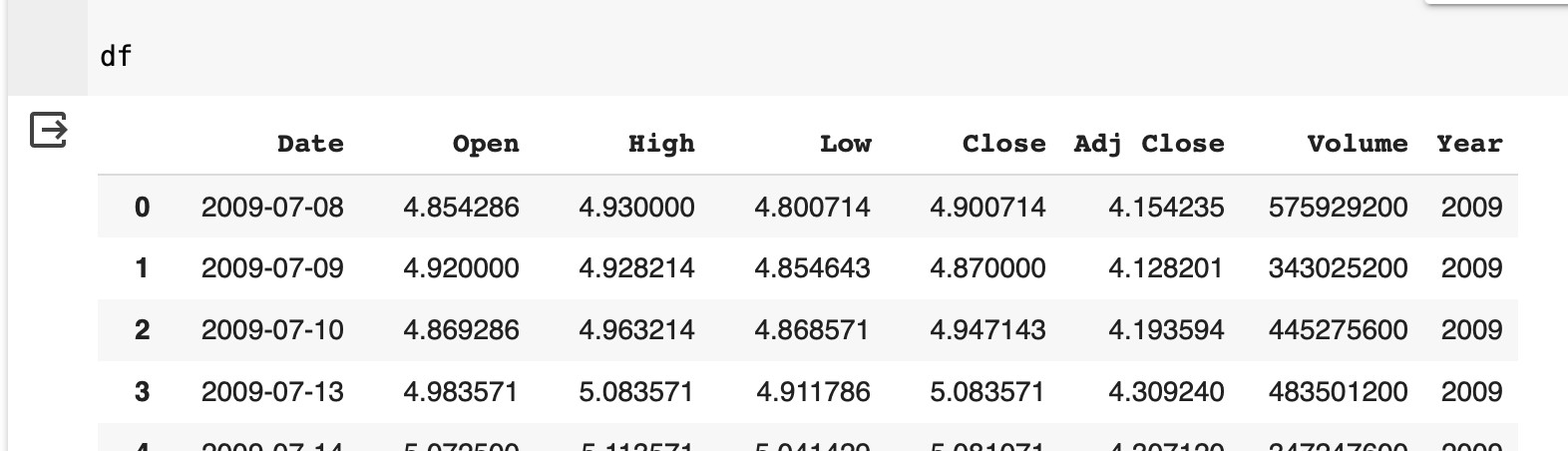
## Figure.3

Then , we imported the downloaded data and loaded the CSV files to python dataframe :



## Figure.4

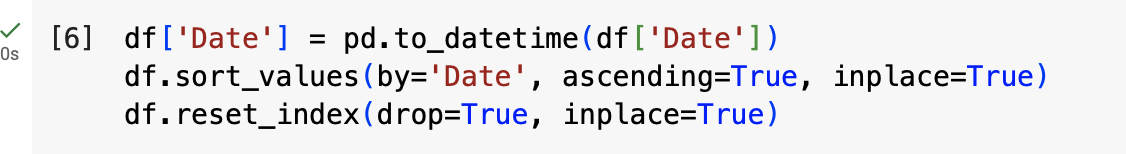
Now the dataset overview :



## Figure.5

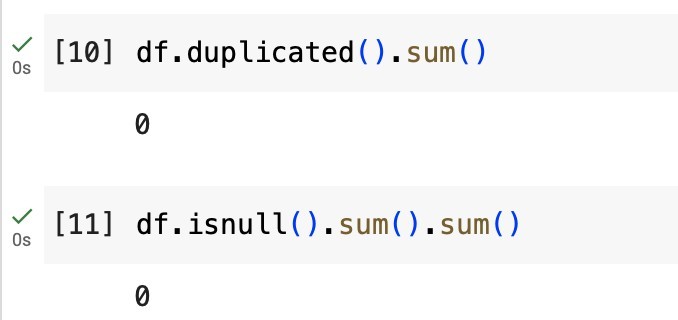
As you can see, the data set includes daily Apple price information including daily Open, High ,Low and Close price of each day along with the volume of transactions and price changes in each day.

Date feature is stored as object in the data frame. To increase the speed of calculations, we convert it's data type to datetime and then sort this feature in ascending order :



## Figure.6

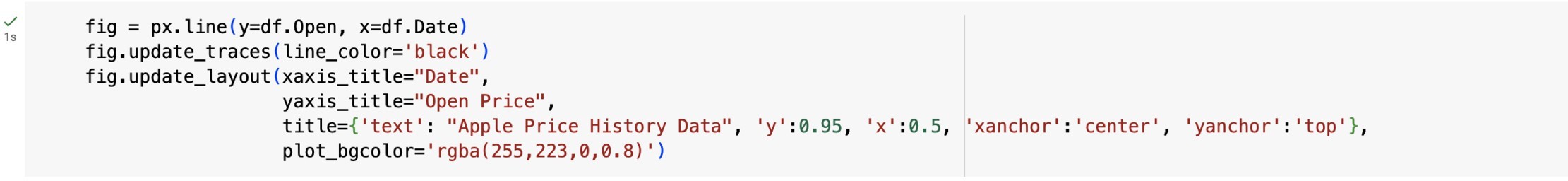
Next,we check if there are any duplicates or missing values in our dataset:



## Figure.7

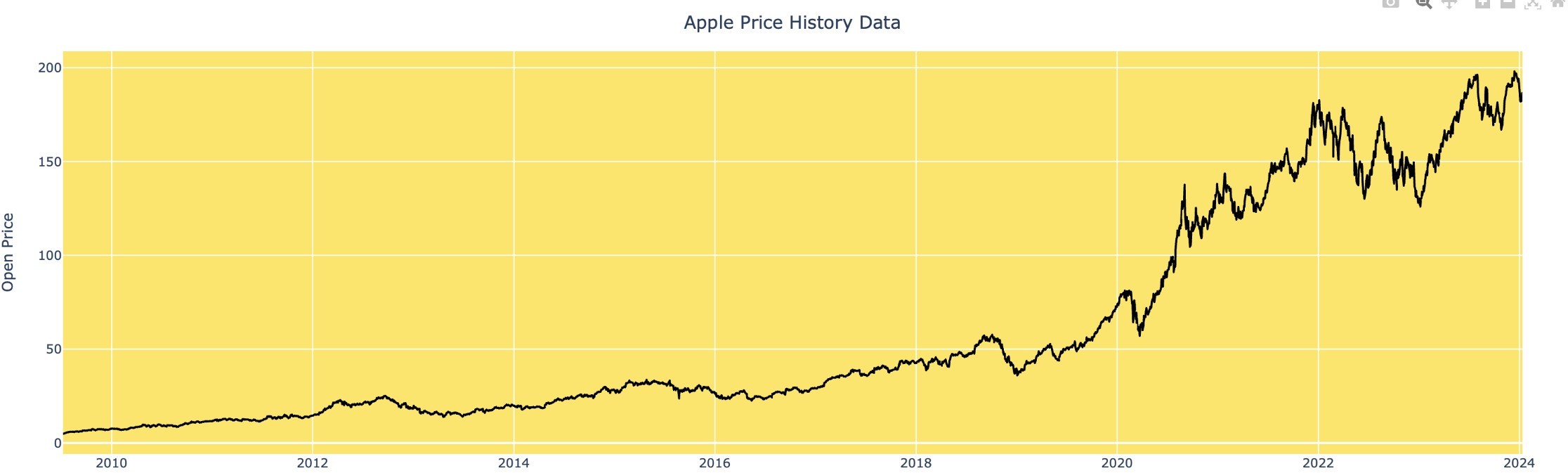
There are no duplicate samples nor missing values .

The Next Step is the visualization of Apple open price history data:



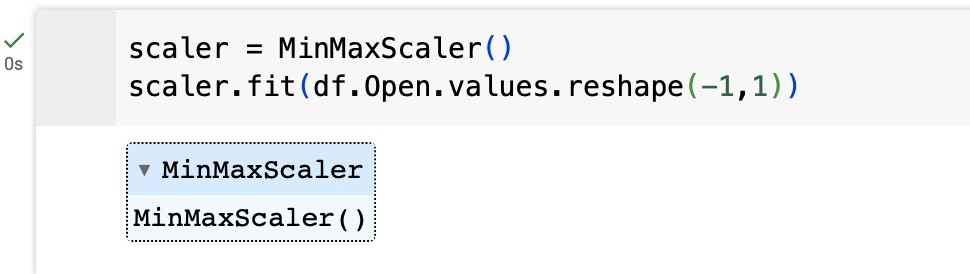
## Figure.8.1

As a result, we obtain this figure :



## Figure.8.2

Now,Since we aim to predict Price only based on its historical data, we scale Price using MinMaxScaler to avoid intensive computations:



## Figure.9

1. **Conclusion:**

In this chapter we provided all the steps taken to make our dataset ready for the training .

# Chapter 2:training and testing the data

## Introduction:

In this chapter , we are going to introduce The ARIMA and The Gru model and train and test our data with each model. We chose to work with the Open prices of apple stock as data in training and testing .

The ARIMA model :It stands for Auto Regressive Integratedd Moving Average . It is a statistical model used for analyzing and predicting Time Series data and it provides a simple yet powerful method for time series forecasts.

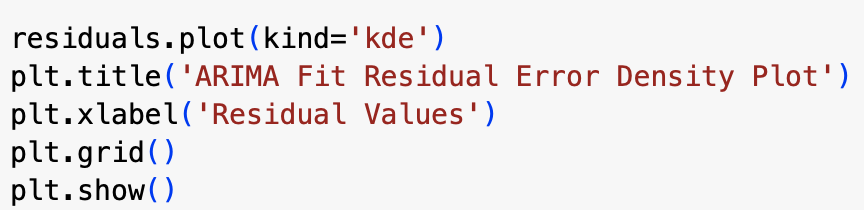
The Gru model : The Gated Recurrent unit or GRU is a kind of current Neural Network.It is capable of capturing complex nonlinear Patterns in time series data.

## Training Model:

ARIMA Model :

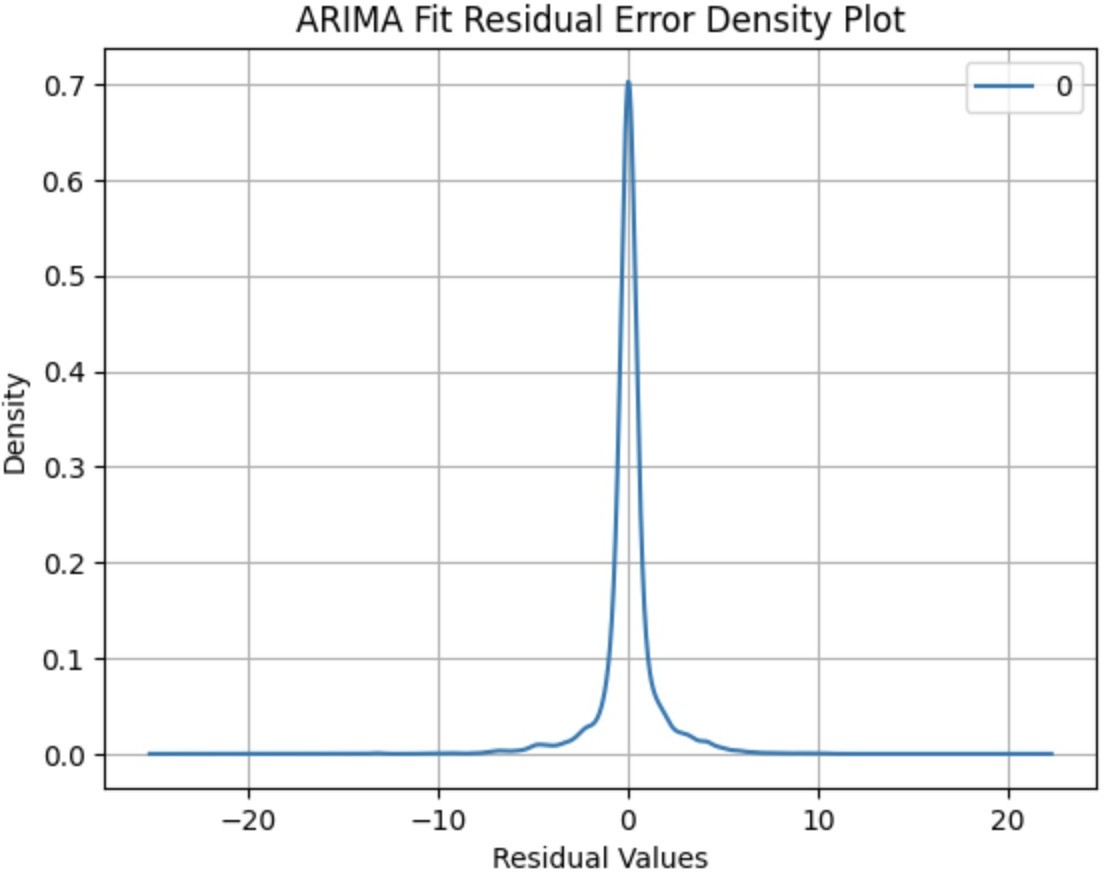
First thing to do when working with the model is Analysing residuals. It is crucial for assessing the models quality and validity .

Ideally the residuals should be random , unbiased and centered around zero. Let's observe The ARIMA Fit residual error density plot :



## Figure.10

we obtain this figure :



## Figure.11

This plot shows that the ARIMA model captures the underlying trend and seasonality in our data very well since the residuals are centered around zero.

The next step is devising our data into training data and testing data :

# Chapter 3: choosing and manipulating the data

## Introduction:

In this chapter , we are going to

## 2-Architecture:

## Microservices architecture (often shortened to microservices) refers to an architectural style for developing applications. Microservices allow a large application to be separated into smaller independent parts, with each part having its own realm of responsibility. To serve a single user request, a microservices-based application can call on many internal microservices to compose its response. Here are some key characteristics of microservices architecture:

## Independently Deployable

## Loosely Coupled

## Communication Through Simple Interfaces

## 

## 3-Software Design:

## 3-Version Control:

We t

## 3-Technologies used:

We t

## 3-Technologies used:

We t

## 4-Authentication and Authorization:

We t

## 5-REST API:

We

## 6-Security:

We t

## 7-Dockerization of the application:

We t