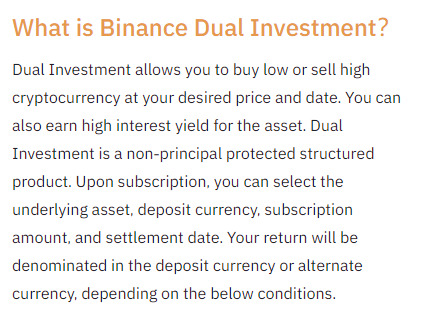
# A.I—BASED DUAL INVESTMENT STRATEGY FOR BINANCE

### How To Save Money with A.I—BASED DUAL INVESTMENT STRATEGY FOR BINANCE

### **introduction**

The numbers can be overwhelming, you see them all over in the financial markets. But what do they actually mean? In this post the goal is to explore the number you will see in the Dual investment platform provided by Binance by using two approaches Artificial Intelligence and Statistics to bring sense to the numbers. Before proceeding further let me define Binance Dual Investment, but instead of repeating myself I will let Binance speak to you



You can see that the investment allows you to buy and sell at your desired price, that fine but how do you know that this desired price will guarantee you a return on investment (ROI) at the end of the trading period? Well, that’s where Artificial Intelligence comes in by using the scientific techniques in this technology one can predict the target price at the settlement date.

Before proceeding further;

**Disclaimer: Any information in this post is meant for educational purpose only. The numbers are meant for financial advice only and cannot serve as investment advice. However, this information can serve as a reference. Before making any investment consult your financial partners and do your personal research.**

From the Binance Dual Investment, we devise a strategy with the following objectives;

1. Develop an Artificial Intelligence model for predicting the desired prices.
2. Train the model with historical cryptocurrency data.
3. Test the developed model with the available real-time data.
4. Compare the Binance target price with the model’s predicted price.
5. Use statistics to quantity the model performance i.e., the deviation of the predicted target price and the actual price.
6. Tweak the strategy based on the obtained outputs.
7. Make the investment.

Note: The main goal is to predict the target price in the Dual Investment.

## LIBRARIES AND REQUIREMENTS

1. Pandas
2. SKLearn
3. Datetime
4. Matplotlib
5. Cryptocurrency/Forex Account.

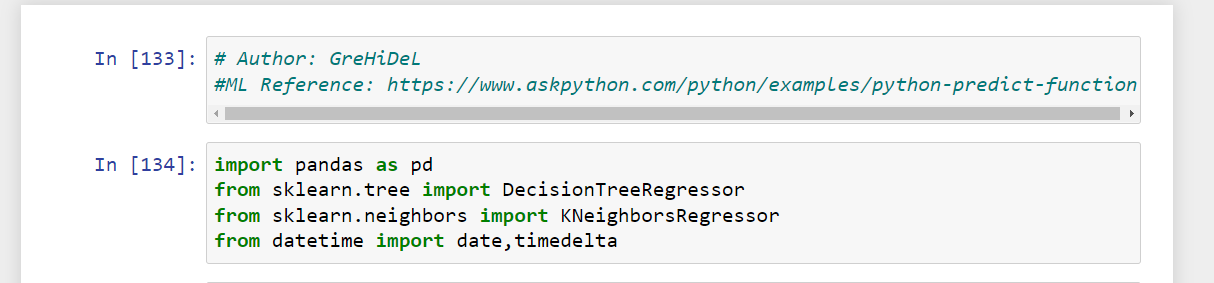
## METHODOLOGY

1. Import the software libraries – Python i.e., Pandas, SKLearn, Datetime
2. Data collection of the recent cryptocurrency data.
3. Use pandas to read the csv file
4. Use the datetime library to determine today’s date
5. Format today’s date into the Year, Month and Day
6. Use the pandas.dataFrame.loc() function fit the data passed as object dtype into a list from the R.H.S.
7. Determine the number of rows in the pandasDataFrame and store them as row\_count (variable)
8. Use the conventional 70-30 rule for training and validation data set by taking 70% of the row\_count and 30% of row\_count.
9. Return 70% of the data and store it as a variable i.e., data using the pandas.DataFrame.head() function that returns the first sections of the data table
10. Using the data\_test variable that stores the last 30% of the rows for validating the model through pandas.dataFrame.tail() function
11. Reshape the dataFrame as a 1-D array by accessing the required information (‘open’ and ‘close’ values) using the dot notation.
12. Store the accessed ‘open’ and ‘close’ information as btc\_close\_data and btc\_open\_data.
13. Implement the Decision Tree Regressor model
14. Implement the KNeighboursRegressor

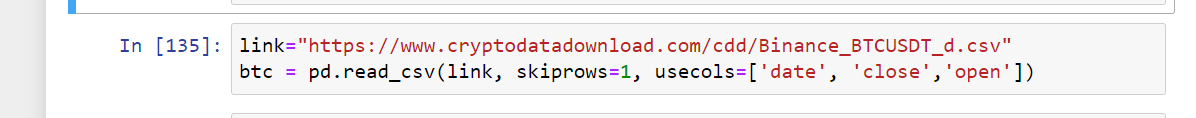
## SOFTWARE

### **access the data using PANDAS**

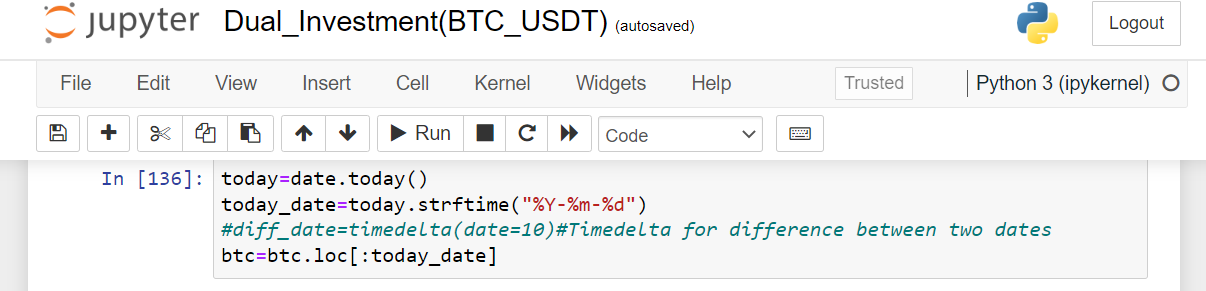
1. Import the software libraries – Python i.e., Pandas, SKLearn, Datetime

****

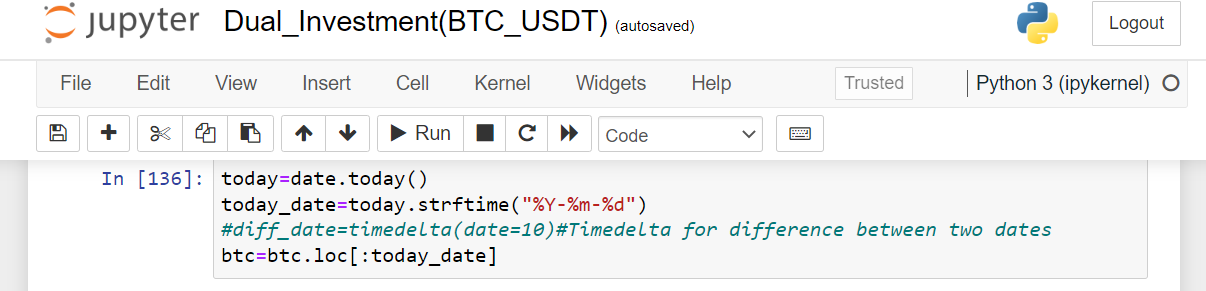
1. Data collection of the recent cryptocurrency data.
2. Use pandas to read the csv file

****

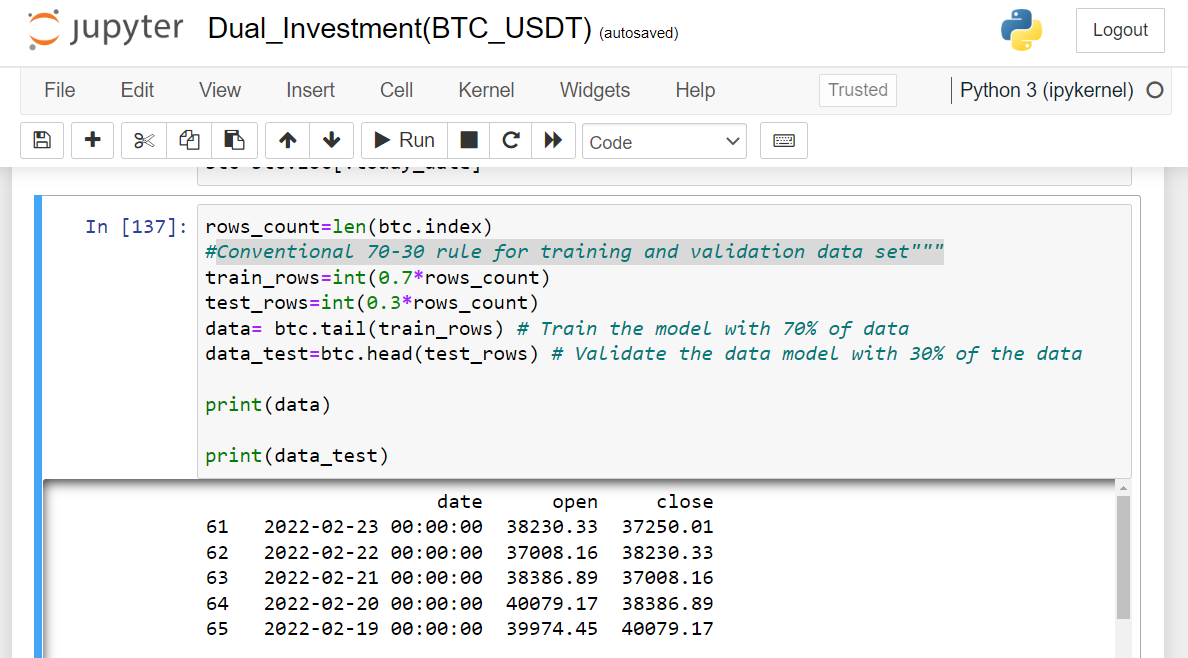
1. Use the datetime library to determine today’s date

****

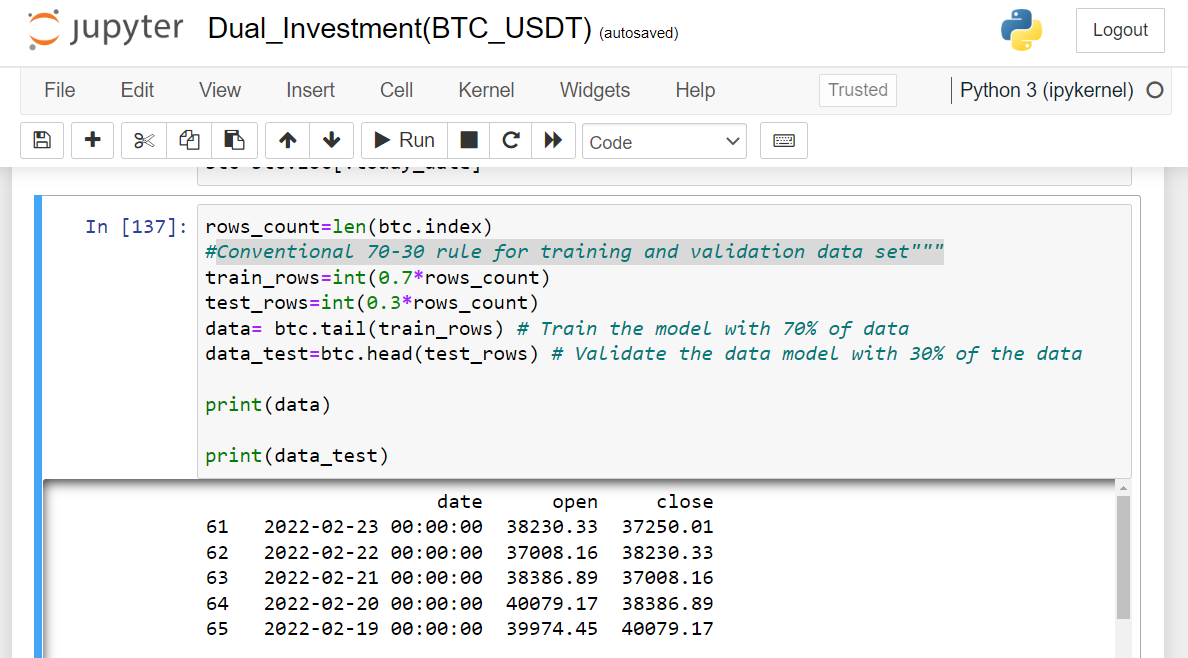
1. Format today’s date into the Year, Month and Day
2. Use the pandas.dataFrame.loc() function fit the data passed as object dtype into a list from the R.H.S.

****

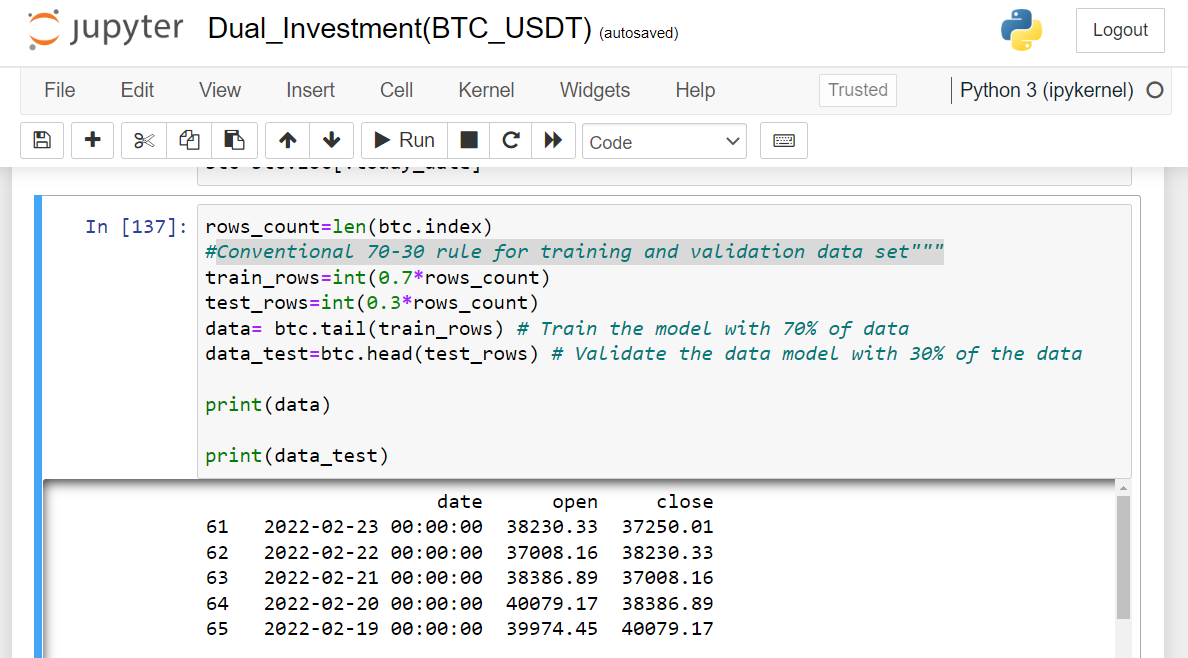
1. Determine the number of rows in the pandasDataFrame and store them as row\_count (variable)

****

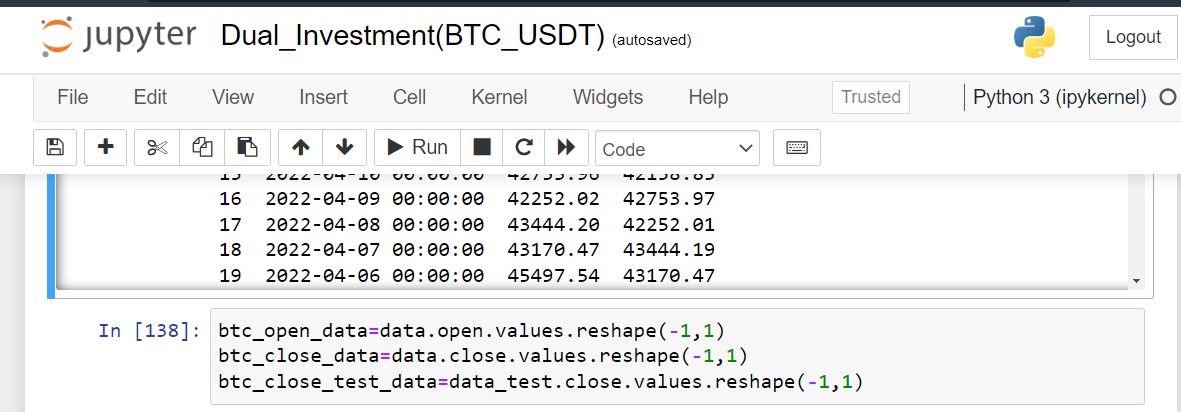
1. Use the conventional 70-30 rule for training and validation data set by taking 70% of the row\_count and 30% of row\_count.



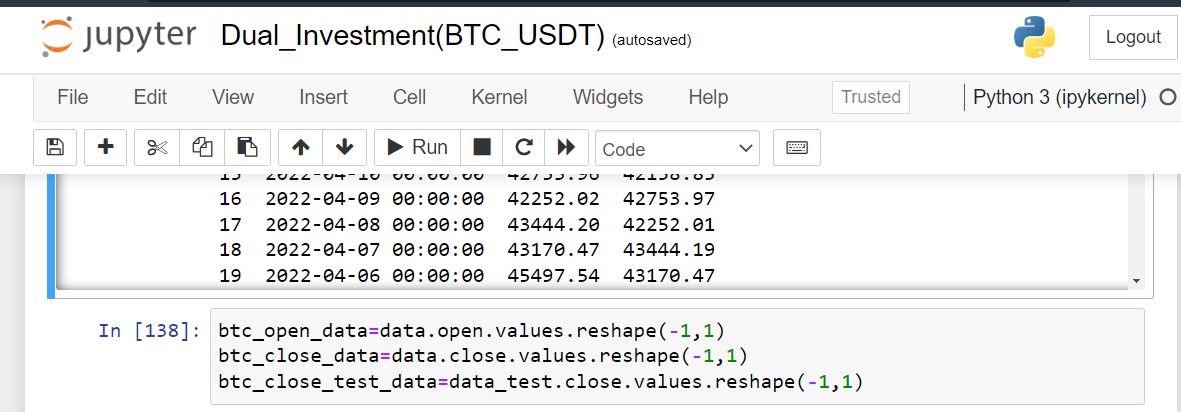
1. Return 70% of the data and store it as a variable i.e., data using the pandas.DataFrame.head() function that returns the first sections of the data table
2. Using the data\_test variable stores the last 30% of the rows for validating the model through pandas.dataFrame.tail() function



1. Reshape the dataFrame as an 1-D array by accessing the required information (‘open’ and ‘close’ values) using the dot notation.



1. Store the accessed ‘open’ and ‘close’ information as btc\_close\_data and btc\_open\_data.

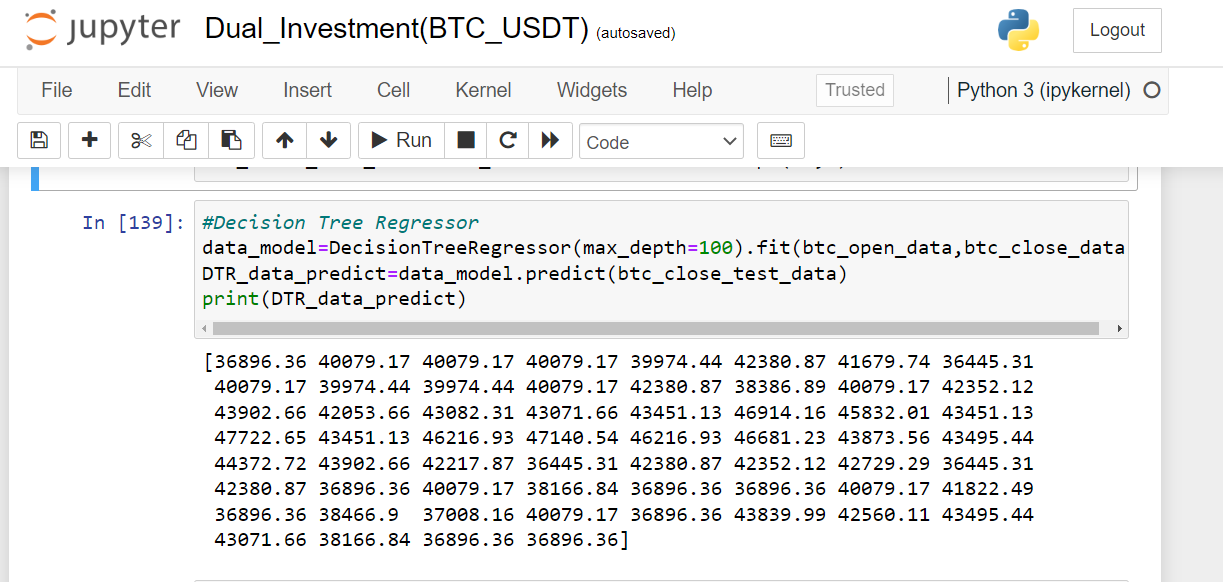


### **ARTIFICIAL INTELLIGENCE – MACHINE LEARNING MODEL**

There are two problems in AI either the **classification** for discrete or categorical data or **regression** for numerical/continuous data. In this case the target price is regression problem.

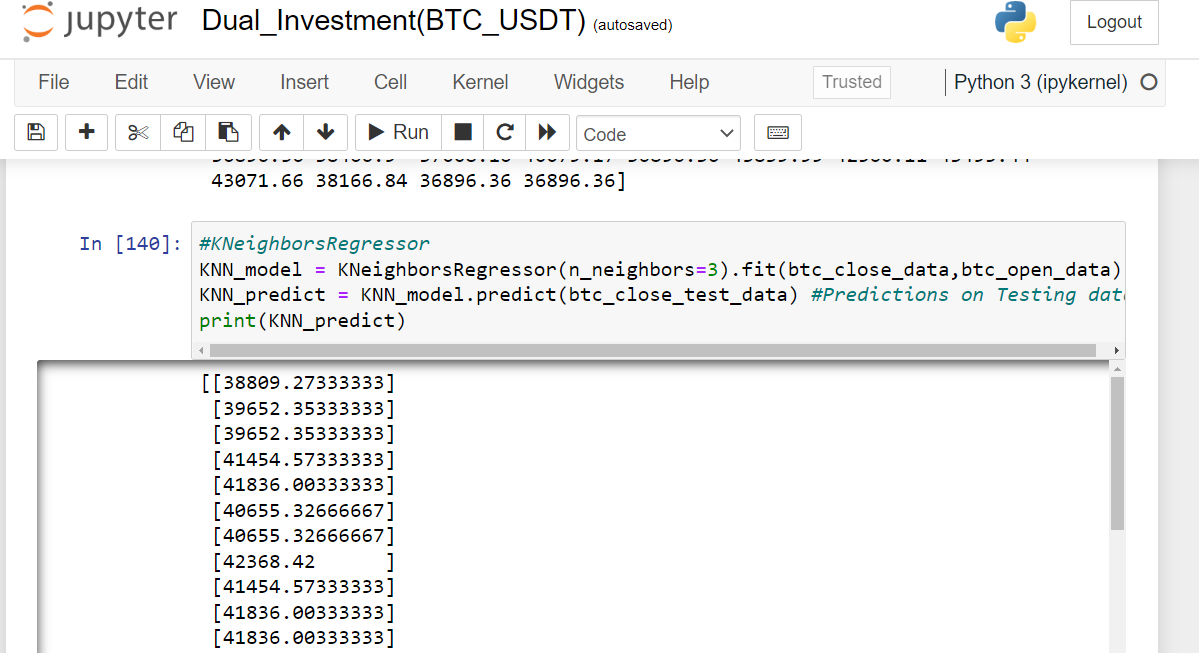
#### Technique 1: Implement the Decision Tree Regressor model

The decision tree model is a flowchart-like structure with an internal node that represents a feature test and outcome of an attribute, the leaf node that denotes the class label.



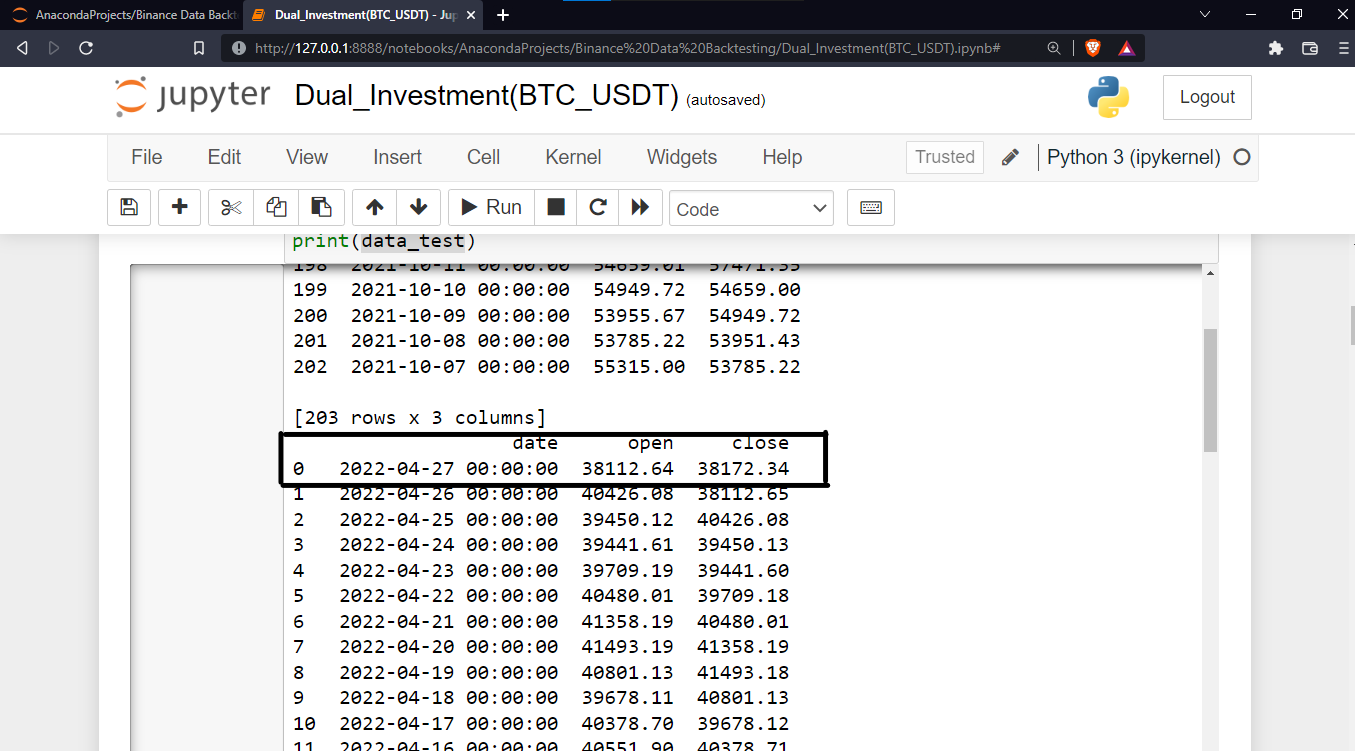
#### Technique 2: K Nearest Neighbour (K-NN) Regressor

The K-NN (K Nearest Neighbour) is an Machine Learning (Subset of A.I technology) approach of analyzing and predicting values using the closest data points. This approach is used in this case of target price forecast since it complies with the assumption that the historical prices are not far off from the future (predicted values) in the sense they are close neighbours.

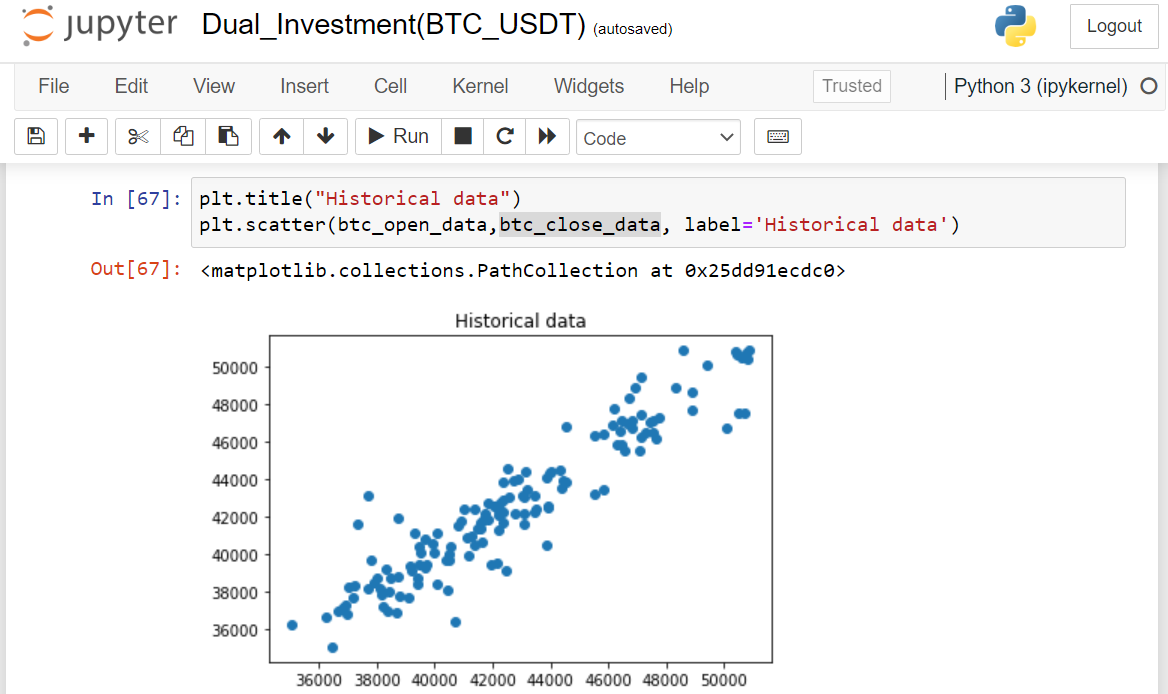


##### DATA PRESENTATION

The values used in the analysis is as displayed below;

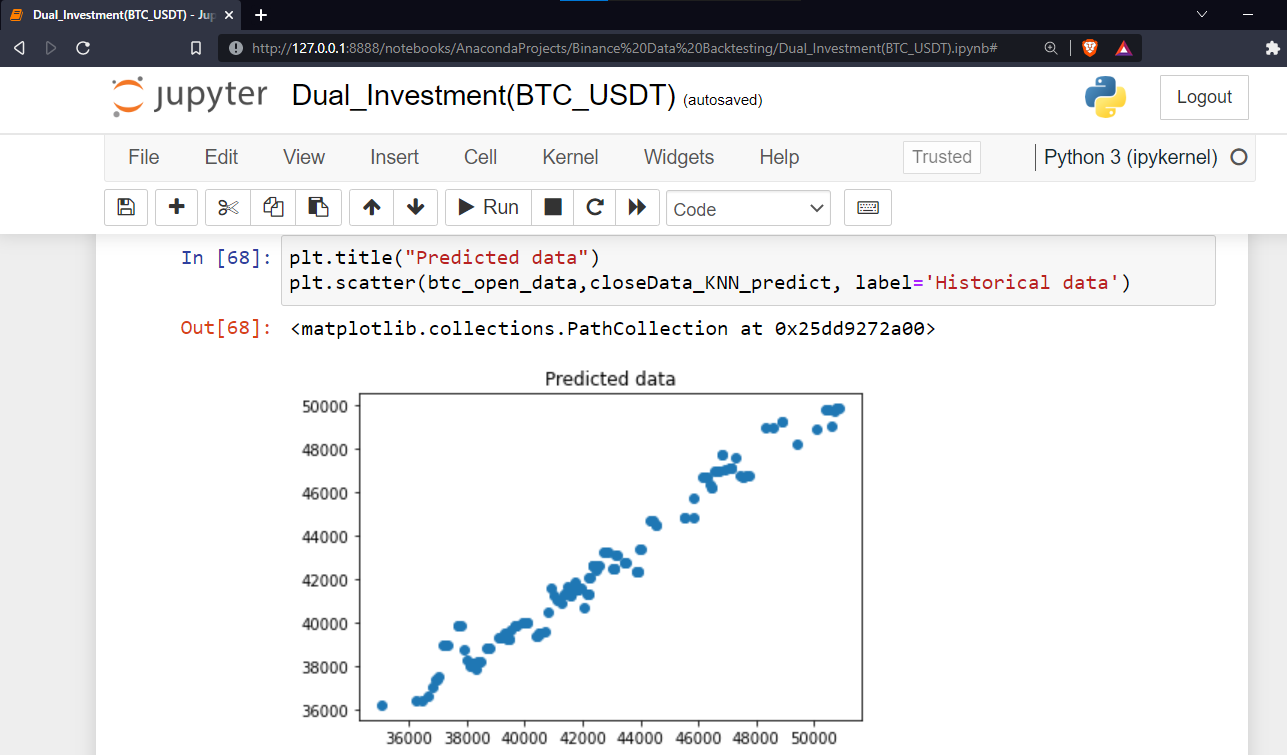


The historical scatter plot is as shown below;



*Figure showing the Historical data scatter plot.*

The K-Nearest Neighbour Predictions of the data is as shown below.



*Figure showing the Predicted data Scatter plot from the K-Nearest Neigbour*

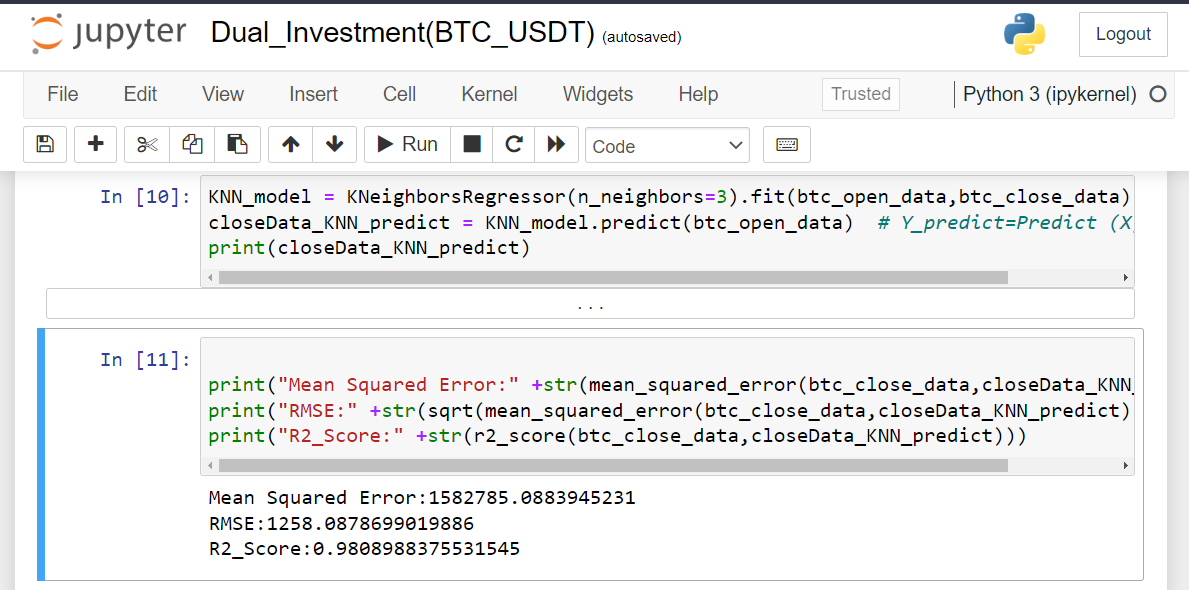
##### MODEL VALIDATION

1. Mean Squared Error (MSE) and Root Mean Square Error (RMSE)

These are statistical measures that quantify the model’s performance. A lower RMSE and MSE value indicates a higher accuracy while higher ones indicates the vice versa.

1. R2 Score

The R2\_ score in sklearn.metrics is the co-efficient of determination that also quantifies the model’s accuracy. This is the measure that determines the level of proportion between the dependent and independent variable. Its values ranges from 0 to 1 where the latter indicates a perfect model. But because of errors a value close to 1 is accepted for instance in our case it gives a value of 0.98 (98% is predictable) which indicates a good model



**BINANCE ASSESSMENT**

After logging in your Binance account lets confirm the effectiveness of this strategy.

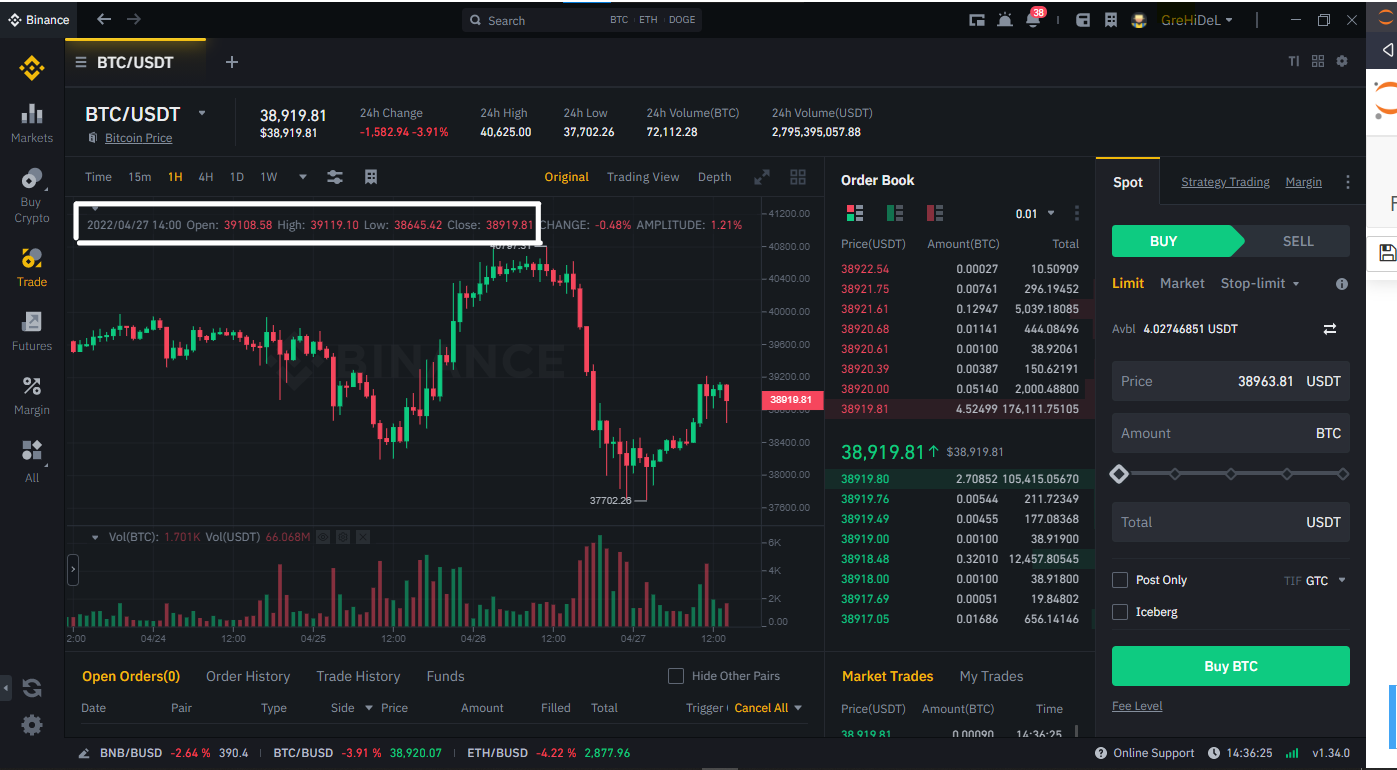
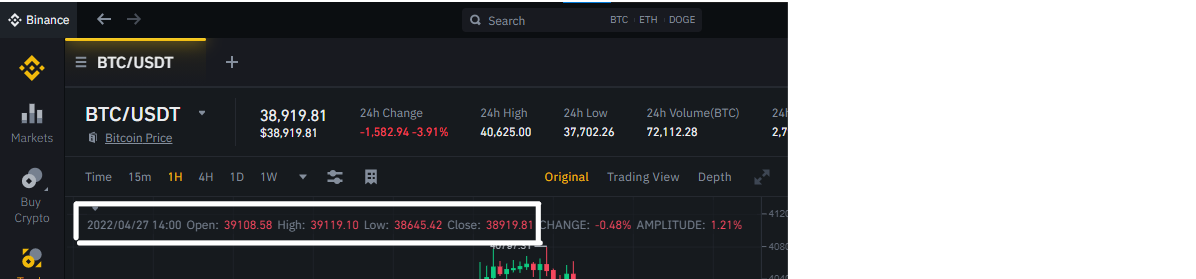
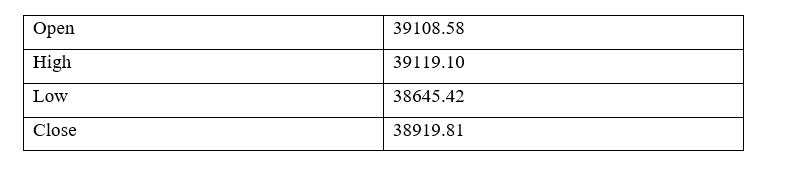


Figure showing the BTC/USDT Trade Performance from Binance.



*Figure showing a closer look at the Open and Close values from Binance*.

In the illustration above the values for 27th April 2022 are as follows



*Figure showing the table values from Binance*

Recall the goal was to predict the closing price, so we use that to quantify the strategy’s effectiveness. The model’s prediction was 38008.685 while that from Binance is 38919.81 as shown with the figure below;

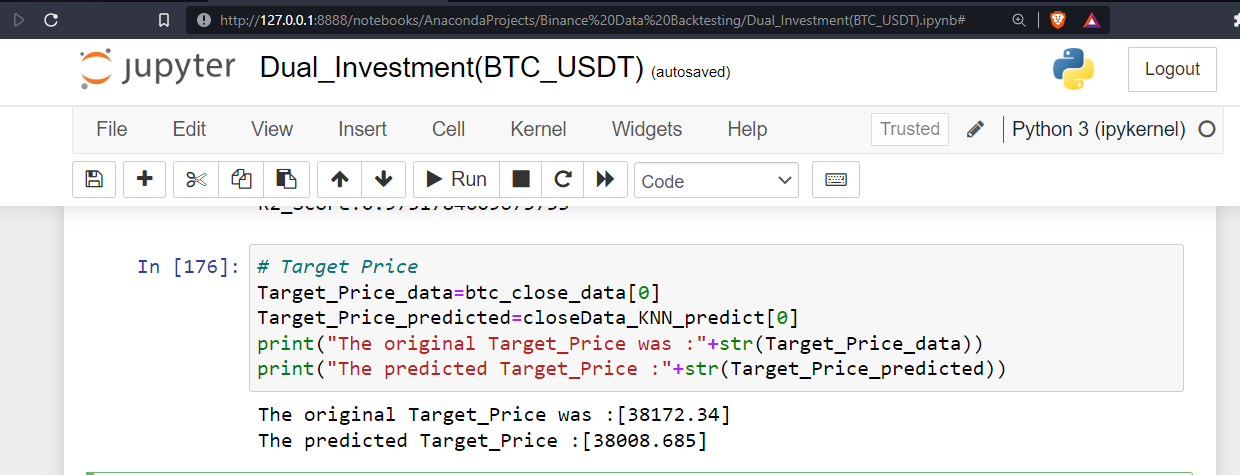
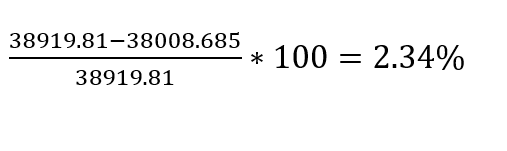


Figure showing the Model predicted value using K-Nearest Neighbour Model.

The % Error in this case is 

We then make an inference for the Dual Investment

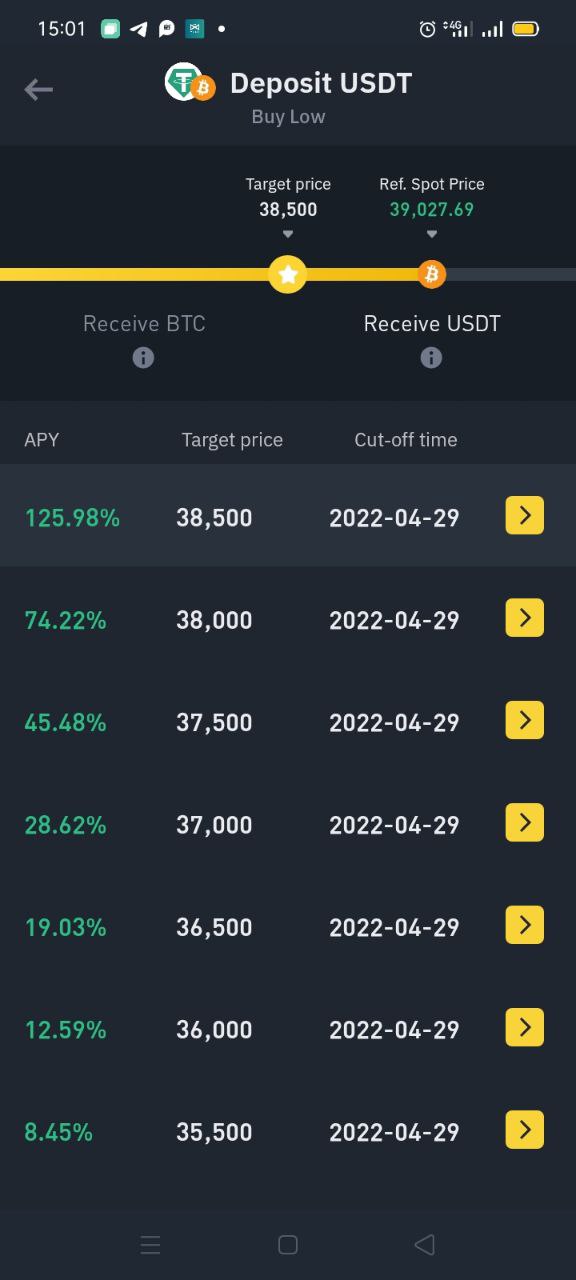


Figure showing the available Dual Investment for the BTC/USDT pairs.

From the figure we can see the target prices flunctuate between 35,500 and 38,500 for the different dates. By assuming all factors remain constant in the market, then our predicted value of 38008.685 is between the range and guarantees a Return of Investment.

In conclusion, to enjoy the full benefits of Binance Investing, create your account here. Also recall my disclaimer any information given by this post is meant for educational purpose only before investing also do your personal research. But obviously, I would appreciate if you quote me as a reference and follow for more of this content.

# REFERENCES

Ask Python: <https://www.askpython.com/python/examples/python-predict-function>

My Great learning <https://www.mygreatlearning.com/blog/mean-square-error-explained/>

# APPENDIX

The code is available from this link:

<https://github.com/ElectronicsDr/Crypto-API-SP-/blob/main/Bot%20Trading%20Assessment/Dual_Investment(BTC_USDT).ipynb>