2022

Crypto-Prices Prediction

GROUP MEMBERS:
JEHANZEB TANVEER (SP19-BSE-104)
M. MAARIJ MALIK (SP19-BSE-095)

Introduction:

Topic: Cryptocurrency Price Prediction with Machine Learning

Aim: (What)

To predict the future unit prices of different cryptocurrencies using historical data of price rates.

Overview: (Why)

Predicting the price of cryptocurrencies is one of the popular case studies in the data science community. The prices of stocks and cryptocurrencies don't just depend on the number of people who buy or sell them. Today, the change in the prices of these investments also depends on the changes in the financial policies of the government regarding any cryptocurrency. The feelings of people towards a particular cryptocurrency or personality who directly or indirectly endorse a cryptocurrency also result in a huge buying and selling of a particular cryptocurrency, resulting in a change in prices.

In short, buying and selling result in a change in the price of any cryptocurrency, but buying and selling trends depend on many factors. Using machine learning for cryptocurrency price prediction can only work in situations where prices change due to historical prices that people see before buying and selling their cryptocurrency. So, in the section below, We will take you through how you can predict the bitcoin prices (which is one of the most popular cryptocurrencies) for the next 30 days.

Tools and Algorithms

- **yfinance API:** For live data extraction of latest cryptocurrency prices provided by Yahoo Finance.
- **AutoTS Python Library:** AutoTS is a time series analysis package for Python designed for rapidly deploying high-accuracy forecasts at scale. There are dozens of forecasting models usable in the sklearn style of .fit() and .predict(). These includes naive, statistical, machine learning, and deep learning models. Additionally, there are over 30 time series specific transforms usable in the sklearn style of .fit(), .transform() and .inverse_transform(). All of these functions directly on Pandas data-frames, without the need for conversion to proprietary objects.

  All models support forecasting multivariate (multiple time series) outputs and support probabilistic (upper/lower bound) forecasts. Using Regression techniques, most models can readily scale to tens and even hundreds of thousands of input series. Many models also support passing in user-defined exogenous regressors. These models are all designed for integration in an AutoML feature search which automatically finds the best models, preprocessing, and ensembling for a given dataset through genetic algorithms.

- **Regression**

  Regression is a technique for investigating the relationship between independent variables or features and a dependent variable or outcome. It's used as a method for predictive modelling in machine learning, in which an algorithm is used to predict continuous outcomes. In our Code, AutoTS uses the following of its custom regression models:

  1. Multivariate Regression
  2. Datepart Regression
  3. Window Regression

  Table:

Sr#	**Name**	**Purpose**
1	Datepart Regression	Datepart Regression is where X is simply the date features, and Y are the time series values for that date.
2	Multivariate Regression	Multivariate Regression is a technique that estimates a single regression model with more than one outcome variable.
3	Window Regression	Window Regression takes an n preceding data points as X to predict the future value or values of the series.

- **Naïve Probability Model**

  Naive Bayes is a kind of classifier which uses the Bayes Theorem. It predicts membership probabilities for each class such as the probability that given record or data point belongs to a particular class. The class with the highest probability is considered as the most likely class. In our Code, AutoTS uses the following of its custom Naive models:

  1. Constant Naive
  2. LastValue Naive
  3. Average Naïve
  4. Seasonal Naïve

Table:

Sr#	Name	Purpose
1	Constant Naive	For Constant naïve forecasts, we set a constant number of forecasts to be evaluated for the forecast.
2	Last-Value Naive	For Last-Value naïve forecasts, we simply set all forecasts to be the value of the last observation. This method works remarkably well for many economic and financial time series.
3	Average Naive	Average Naïve forecasting is the technique in which the average of previous data entries are used for the next period's forecast without predictions or adjusting the factors.
4	Seasonal Naive	A similar method is useful for highly seasonal data. In this case, we set each forecast to be equal to the last observed value from the same season (e.g., the same month of the previous year).

- **Motif Discovery**
  Time series motifs are pairs of individual time series, or subsequences of a longer time series, which are very similar to each other. As with their discrete analogues in computational biology, this similarity hints at structure which has been conserved for some reason and may therefore be of interest. In our Code, AutoTS uses the following of its custom Motif Discovery models:
  1. Multivariate Motif
  2. Univariate Motif
  3. Sectional Motifs

Table:

Sr#	Name	Purpose
1	Multivariate Motif	Multivariate Motif Discovery gives multiple outcomes based on the under-test data.
2	Univariate Motif	Univariate Motif Discovery finds a hidden pattern in a dataset that can be useful for different purposes, mostly business perspective.
3	Sectional Motif	As the name suggests, Sectional Motif Discovery takes specified sections of our dataset to make the model.

Advantages of Time Series:

The advantages of time series analysis are high accuracy and simplicity. Memories are fragile and prone to error. You may think that your sales peak before Christmas and hit their bottom in February… but do they really?

The simplest and, in most cases, the most effective form of time series analysis is to simply plot the data on a line chart. With this step, there will no longer be any doubts as to whether or not sales truly peak before Christmas and dip in February.

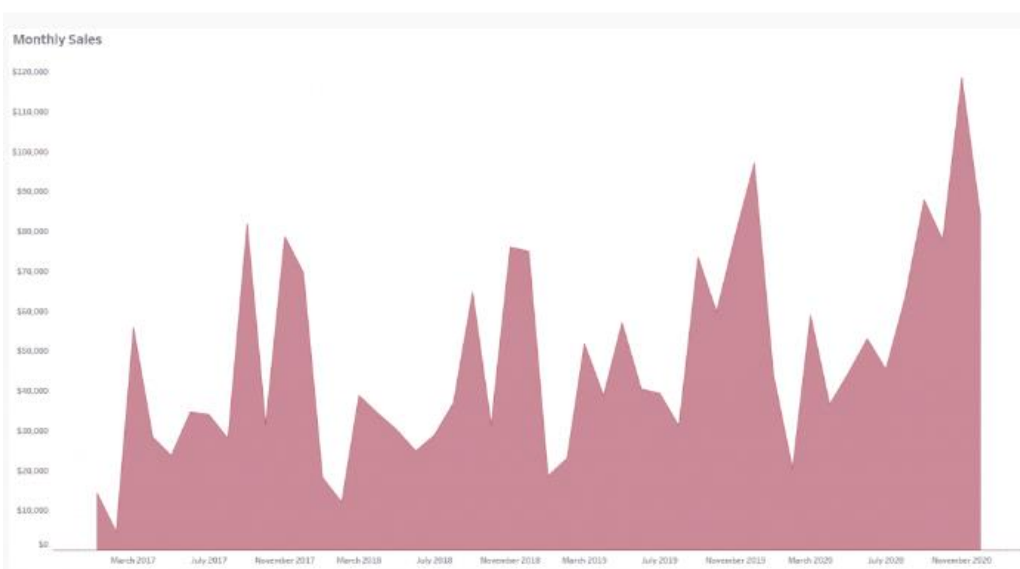

Figure 1 - Finding hidden patterns through Time-Series Analysis

Now that we have plotted our sales data, it becomes immediately clear what patterns we have. Sales are trending upwards year-over-year, and seem to follow a regularly yearly pattern. The months of January and February see the lowest sales figures and there is a major spike in November and December.

A major benefit of time series analysis is that it can be the basis to forecast data. This is because time series analysis — by its very nature — uncovers patterns in data, which can then be used to predict future data points.

For example, autocorrelation patterns and seasonality measures can be used to predict when a certain data point can be expected. Further, stationarity measures can be used to estimate what the value of that data point will be.

Really, it’s the forecasting aspect of time series analysis that makes it so popular in business applications. Analyzing and understanding past data is all good and well, but it’s being able to predict the future that helps to make optimal business decisions.

Time series analysis is a really handy tool as it gives accurate predictions for future values, but it also requires more skill than regression analysis since you need to adapt your model according to the historical data.

Another benefit of time series analysis is that it can help to clean data. This makes it possible to find the true “signal” in a data set, by filtering out the noise. This can mean removing outliers or applying various averages so as to gain an overall perspective of the meaning of the data.

Of course, cleaning data is a prominent part of almost any kind of data analysis. The true benefit of time series analysis is that it is accomplished with little extra effort.

Disadvantages of Time Series:

Time series analysis is useful for short-term forecasting, but it could sometimes lead to wrong predictions. This is because it requires historical data in order to construct the models, which means that if some significant changes occurred over time, then those changes will not be included within the forecasted periods. One more thing that can happen is related to outliers and error propagation. If there are a lot of outliers inside your model and they are not properly handled by the time series technique you have used, then errors will be propagated all throughout your forecasts. Another disadvantage of time series analysis comes from its sensitivity to changing trends over time since most statistical methods require stationary data as a preliminary condition.

Another disadvantage of time series analysis is that it doesn't give the exact project value, rather than a probability distribution on possible future outcomes. This means that we can forecast how probable specific values will be reached but not what they're exactly going to be.

Dataset:

Live data extraction at every execution of code from “Yahoo Finance” using yfinance API.

EDA(Exploratory Data Analysis)

Figure 2 - Ethereum Price Analysis

Figure 3 - Bitcoin Price Analysis

Experimental Setup:

Predicting the future prices of cryptocurrency is based on the problem of Time series analysis. The AutoTS library in Python is one of the best libraries for time series analysis. So here We will be using the AutoTS library to predict the bitcoin prices for the next 30 days:

Using yfinance API to extract data of Cryptocurrency prices.

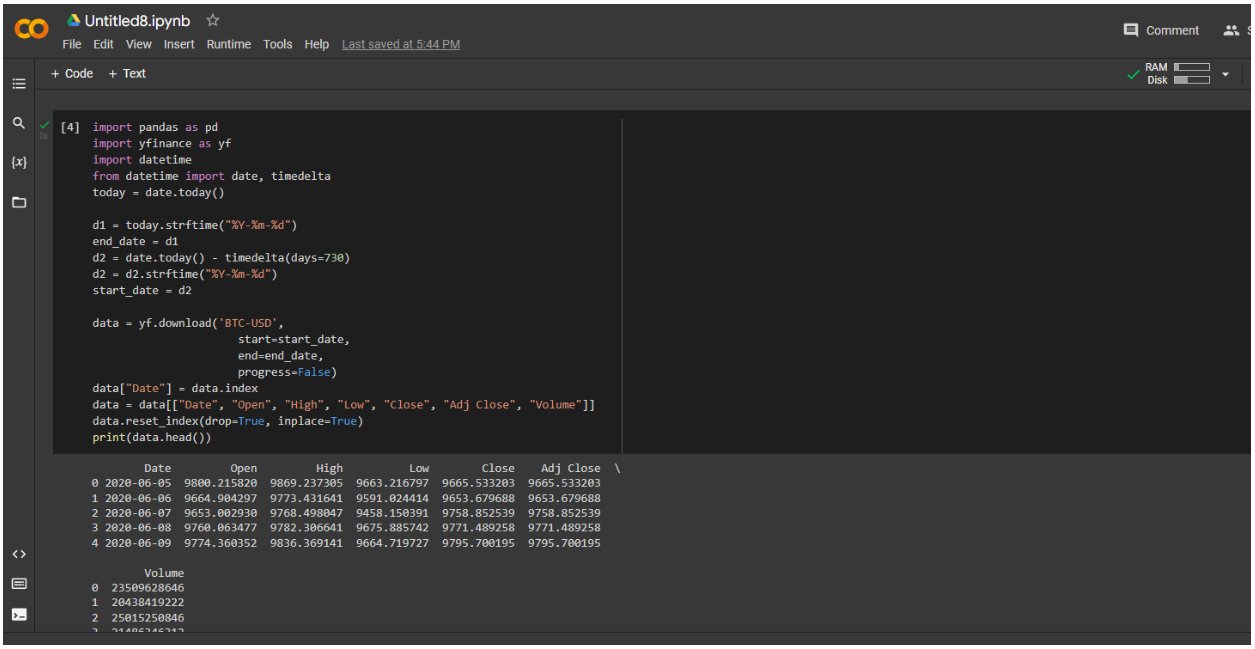

Figure 4 - Data Extraction using yFinance API

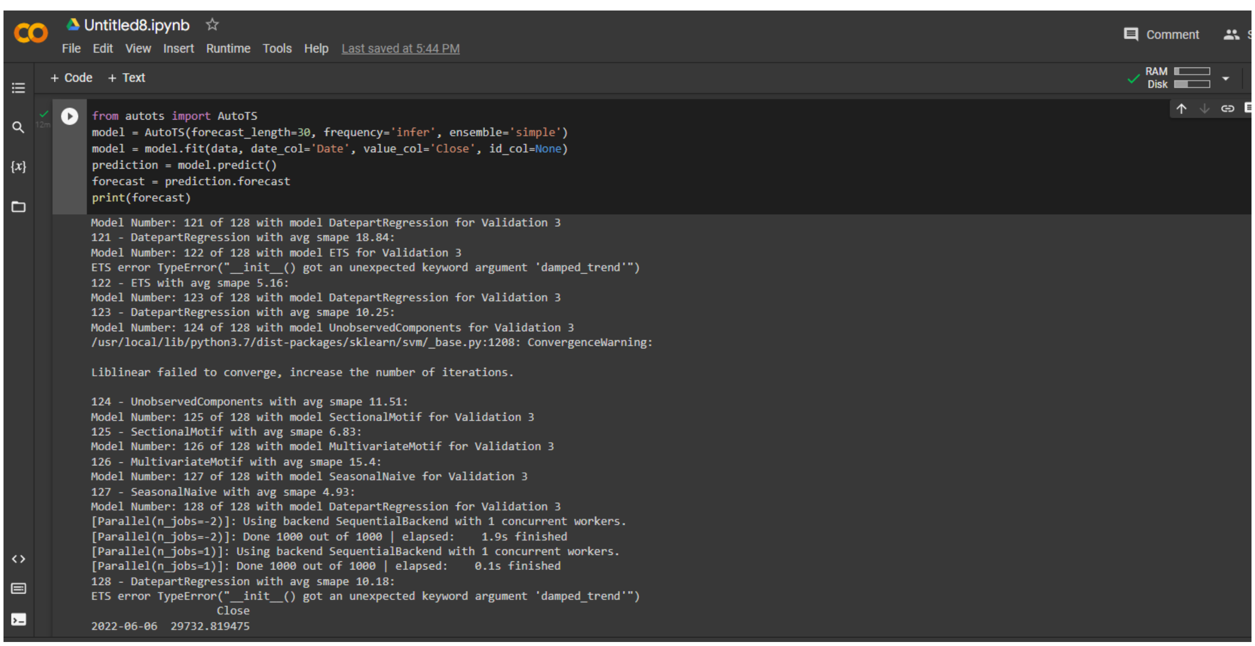

Figure 5 - Using AutoTS to predict future crypto-prices of 30 days.

```
[8] pip install autots

Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
Collecting autots
  Downloading AutoTS-0.4.1-py3-none-any.whl (470 kB)
     |████████████████████████████████| 470 kB 20.0 MB/s
Requirement already satisfied: statsmodels>=0.10.* in /usr/local/lib/python3.7/dist-packages (from autots) (0.10.2)
Requirement already satisfied: numpy>=1.14.6 in /usr/local/lib/python3.7/dist-packages (from autots) (1.21.6)
Requirement already satisfied: pandas>=0.25.* in /usr/local/lib/python3.7/dist-packages (from autots) (1.3.5)
Requirement already satisfied: scikit-learn>=0.20.* in /usr/local/lib/python3.7/dist-packages (from autots) (1.0.2)
Requirement already satisfied: pytz>=2017.3 in /usr/local/lib/python3.7/dist-packages (from pandas>=0.25.*->autots) (2022.1)
Requirement already satisfied: python-dateutil>=2.7.3 in /usr/local/lib/python3.7/dist-packages (from pandas>=0.25.*->autots) (2.8.2)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.7/dist-packages (from python-dateutil>=2.7.3->pandas>=0.25.*->autots) (1.15.0)
Requirement already satisfied: joblib>=0.11 in /usr/local/lib/python3.7/dist-packages (from scikit-learn>=0.20.*->autots) (1.1.0)
Requirement already satisfied: threadpoolctl>=2.0.0 in /usr/local/lib/python3.7/dist-packages (from scikit-learn>=0.20.*->autots) (3.1.0)
Requirement already satisfied: scipy>=1.1.0 in /usr/local/lib/python3.7/dist-packages (from scikit-learn>=0.20.*->autots) (1.4.1)
Requirement already satisfied: patsy>=0.4.0 in /usr/local/lib/python3.7/dist-packages (from statsmodels>=0.10.*->autots) (0.5.2)
Installing collected packages: autots
Successfully installed autots-0.4.1
```

Figure 6 - Installing AutoTS Library.

Forecasting Results:

Bitcoin

1- For 30 days

Untitled8.ipynb

File Edit View Insert Runtime Tools Help Last saved at 5:44 PM

+ Code + Text

```
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done 1000 out of 1000 | elapsed:    0.1s finished
128 - DatepartRegression with avg smape 10.18:
ETS error TypeError("__init__() got an unexpected keyword argument 'damped_trend'")
                   Close
2022-06-06  29732.819475
2022-06-07  29732.819475
2022-06-08  29732.819475
2022-06-09  29732.819475
2022-06-10  29732.819475
2022-06-11  29732.819475
2022-06-12  29732.819475
2022-06-13  29732.819475
2022-06-14  29732.819475
2022-06-15  29732.819475
2022-06-16  29732.819475
2022-06-17  29732.819475
2022-06-18  29732.819475
2022-06-19  29732.819475
2022-06-20  29732.819475
2022-06-21  29732.819475
2022-06-22  29732.819475
2022-06-23  29732.819475
2022-06-24  29732.819475
2022-06-25  29732.819475
2022-06-26  29732.819475
2022-06-27  29732.819475
2022-06-28  29732.819475
2022-06-29  29732.819475
2022-06-30  29732.819475
2022-07-01  29732.819475
2022-07-02  29732.819475
2022-07-03  29732.819475
2022-07-04  29732.819475
2022-07-05  29732.819475
```

Figure 7 – Bitcoin Price Forecasting for next 30 days

2- For 7 days

	Close
2022-06-14	23069.782900
2022-06-15	22829.939786
2022-06-16	22590.485736
2022-06-17	22351.420717
2022-06-18	22112.744692
2022-06-19	21874.457622
2022-06-20	21636.559463

Figure 8 - Bitcoin Price Forecasting for next 7 days

3- For 1 day

	Close
2022-06-14	20790.574147

Figure 9 - Bitcoin Price Forecasting for next day

Ethereum:

1- For 30 days

Crypto price predection.ipynb

File Edit View Insert Runtime Tools Help Last edited on 5 June

\+ Code + Text

```
ETS error TypeError("__init__() got an unexpected keyword argument 'damped_trend'")
128 - ETS with avg smape 8.0:
                  Close
2022-06-06  1747.842331
2022-06-07  1681.786779
2022-06-08  1706.492861
2022-06-09  1735.532767
2022-06-10  1707.410517
2022-06-11  1736.769707
2022-06-12  1762.425364
2022-06-13  1732.442289
2022-06-14  1721.506572
2022-06-15  1745.713146
2022-06-16  1759.735785
2022-06-17  1717.599280
2022-06-18  1786.526361
2022-06-19  1765.701681
2022-06-20  1774.732171
2022-06-21  1787.599448
2022-06-22  1808.482746
2022-06-23  1884.944209
2022-06-24  1941.781004
2022-06-25  1953.166727
2022-06-26  1968.094429
2022-06-27  1920.120068
2022-06-28  1969.931656
2022-06-29  1937.971231
2022-06-30  2031.245758
2022-07-01  1984.710775
2022-07-02  2021.555007
2022-07-03  2025.880152
2022-07-04  1991.903217
2022-07-05  1989.768144
```

Figure 10 - Ethereum Price Forecasting for next 30 days

2- For 7 days

	Close
2022-06-14	1390.890228
2022-06-15	1376.361674
2022-06-16	1361.803988
2022-06-17	1347.217168
2022-06-18	1332.601214
2022-06-19	1317.956128
2022-06-20	1303.281908

Figure 11 - Ethereum Price Forecasting for next 7 days

3- For 1 day

```
                Close
2022-06-14  1184.6952
```

Figure 12 - Ethereum Price Forecasting for next day

Model Cross-Validation Results:

For BTC

```
-------------------------------------------
Model Cross Validation Results:
-------------------------------------------
    MAE (Mean Absolute Error = 4624.27
    MSE (Mean Squared Error = 37836615.48
    MAPE (Mean Absolute Percent Error) = 24%
    RMSE (Root Mean Squared Error) = 6151.1475
    Normalized RMSE (MinMax) = 12%
    Normalized RMSE (as Std Dev of Actuals)= 36%
Time Taken = 14 seconds
  End of Prophet Fit
```

Figure 13 - Model Cross-Validation Results for Bitcoin

For ETH

```
-------------------------------------------
Model Cross Validation Results:
-------------------------------------------
    MAE (Mean Absolute Error = 317.74
    MSE (Mean Squared Error = 182393.92
    MAPE (Mean Absolute Percent Error) = 34%
    RMSE (Root Mean Squared Error) = 427.0760
    Normalized RMSE (MinMax) = 23%
    Normalized RMSE (as Std Dev of Actuals)= 72%
Time Taken = 18 seconds
  End of Prophet Fit
```

Figure 14 - Model Cross-Validation Results for Ethereum

References

Yahoo Finance: https://finance.yahoo.com/cryptocurrencies

AutoTS Documentation: https://pypi.org/project/AutoTS/

Model Info Links:

Naive Models: https://otexts.com/fpp3/simple-methods.html

Regression Models: https://winedarksea.github.io/AutoTS/build/html/source/tutorial.html#id10

Motif Discovery Models: https://www.researchgate.net/publication/264716280_Time_series_motif_discovery_Dimensions_and_applications