CS210 PROJECT REPORT

Tunahan Arslan

29210

Datasets:

I used two datasets for this project,

First one is the 2019-2024 US stock market dataset from:

https://www.kaggle.com/datasets/saketk511/2019-2024-us-stock-market-data

Second one is the Covid-19 global dataset from:

https://www.kaggle.com/datasets/josephassaker/covid19-global-dataset

The purpose of this project is to investigate if there have been a correlation between the cryptocurrency bull-season and the covid-19 pandemic in the world specifically in the 2020-2022 time period. For this purpose, we are going to focus primarly on the Bitcoin data as well as the average daily new covid cases reported in the US. The reason why we focus on Bitcoin is because it is the most clear indicator of the state of the cryptocurrency market throughout history. And the reason why we consider only the US data on covid case numbers is because US was the most capable country in terms of testing therefore providing a very clear data closest to the real case numbers compared to every other country. The average daily covid case numbers in the US is a clear indicator of the course of the pandemic in the world in general.

The implementation

First we read our first dataset and store it as df1. We want to focus on the timespan between the end of 2022 and the beginning of 2020 so we extract that part of the data and print the first 5 rows.

	Unnamed: 0	Date	Natural_Gas_Price	Natural_Gas_Vol.	Crude_oil_Price	Crude_oil_Vol.	Copper_Price	Copper_Vol.	Bitcoin_Price	Bitcoin_Vol.	...	Berkshire_Price	Berkshire_Vol.	Netflix_Price	Netflix_Vol.	Amazon_Price	Amazon_Vol.
273	273	2022-12-30	4.475	62280.0	80.47	NaN	3.8130	10.0	16,607.20	192760.0	...	4,68,711	3360.0	294.88	7570000.0	84.00	62400000.0
274	274	2022-12-29	4.559	78440.0	78.61	NaN	3.8265	40.0	16,636.40	181470.0	...	4,68,725	3040.0	291.12	9560000.0	84.18	55000000.0
275	275	2022-12-28	4.709	1720.0	78.60	NaN	3.8425	42610.0	16,546.20	217960.0	...	4,59,800	3030.0	276.88	5930000.0	81.82	58230000.0
276	276	2022-12-27	5.282	41150.0	79.77	NaN	3.8405	48120.0	16,706.10	192180.0	...	4,61,955	4470.0	284.17	5690000.0	83.04	57280000.0
277	277	2022-12-23	4.980	87380.0	79.34	NaN	3.8090	38560.0	16,779.10	184120.0	...	4,63,400	2800.0	294.96	4250000.0	85.25	57430000.0

5 rows × 39 columns

As said earlier, we are going to focus on Bitcoin on our first dataset. So we want to check if there is any null value among the Bitcoin_Price values to get rid of outliers. Our code gives 0 for the sum of null values.

We use describe() method to further investigate our Bitcoin_Price values.

```
count          740
unique         739
top       21,365.20
freq             2
Name: Bitcoin_Price, dtype: object
```

Next thing is to read the second dataset and do the same operations. We read and store the second dataset as df2. We print the first 5 rows to get a closer look at the dataset.

	date	country	cumulative_total_cases	daily_new_cases	active_cases	cumulative_total_deaths	daily_new_deaths
0	2020-2-15	Afghanistan	0.0	NaN	0.0	0.0	NaN
1	2020-2-16	Afghanistan	0.0	NaN	0.0	0.0	NaN
2	2020-2-17	Afghanistan	0.0	NaN	0.0	0.0	NaN

	date	country	cumulative_total_cases	daily_new_cases	active_cases	cumulative_total_deaths	daily_new_deaths
3	2020-2-18	Afghanistan	0.0	NaN	0.0	0.0	NaN
4	2020-2-19	Afghanistan	0.0	NaN	0.0	0.0	NaN

We again use the describe() method on this data.

	cumulative_total_cases	daily_new_cases	active_cases	cumulative_total_deaths	daily_new_deaths
count	1.847870e+05	174329.000000	1.667470e+05	1.782270e+05	157850.000000
mean	7.251089e+05	2987.633285	6.239283e+04	1.388600e+04	39.831834
std	3.681471e+06	17803.232663	3.955641e+05	6.049521e+04	181.102770
min	0.000000e+00	-322.000000	-1.432100e+04	0.000000e+00	-39.000000
25%	1.099000e+03	0.000000	6.000000e+01	2.400000e+01	0.000000
50%	1.775600e+04	58.000000	1.386000e+03	3.040000e+02	1.000000
75%	2.238085e+05	728.000000	1.462050e+04	4.111000e+03	12.000000
max	8.420947e+07	909610.000000	1.793543e+07	1.026646e+06	5093.000000

As we can see this data includes all the countries in the world and not indicative as a whole in terms of the effect of the covid 19 pandemic. We need to narrow it down to a more useful data. As we said earlier we want to only use the data from the US for that purpose.

To accomplish this objective we write the proper code to extract only the USA information from df2. We take only the data from the time interval 2020-2022 on this dataset as well. We name our new dataset as us_data and print the first five rows to see our dataframe.

	date	country	cumulative_total_cases	daily_new_cases	active_cases	cumulative_total_deaths	daily_new_deaths
176587	2020-02-15	USA	15.0	NaN	12.0	0.0	NaN
176588	2020-02-16	USA	15.0	0.0	12.0	0.0	NaN
176589	2020-02-17	USA	15.0	0.0	12.0	0.0	NaN
176590	2020-02-18	USA	15.0	0.0	12.0	0.0	NaN
176591	2020-02-19	USA	15.0	0.0	12.0	0.0	NaN

Using the describe() method this time on us_data,

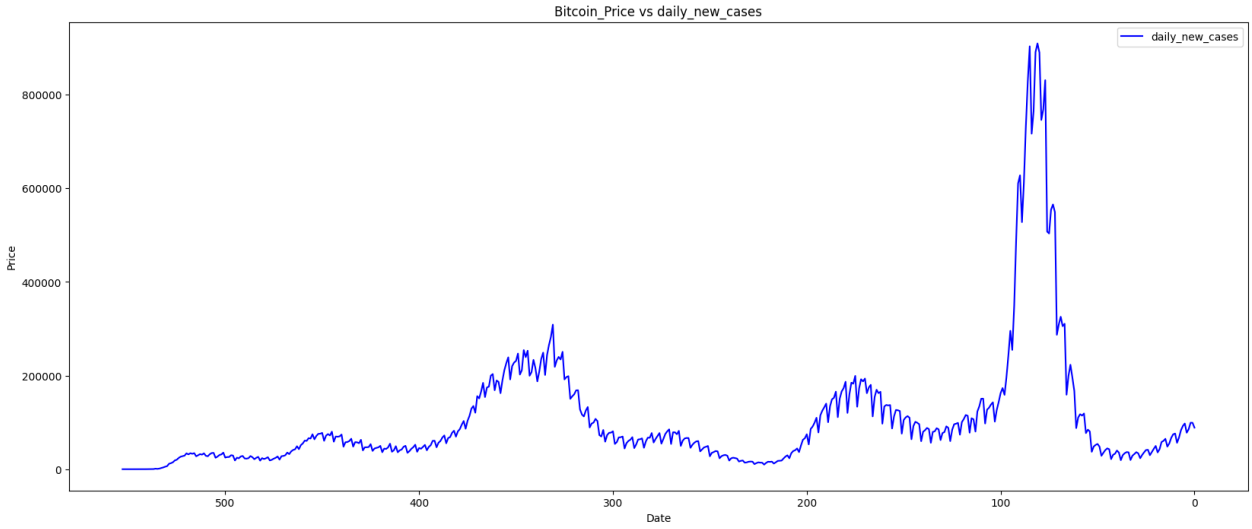
	date	cumulative_total_cases	daily_new_cases	active_cases	cumulative_total_deaths	daily_new_deaths
count	820	8.200000e+02	819.000000	8.200000e+02	8.200000e+02	815.000000
mean	2021-03-30 12:00:00.000000256	3.161365e+07	102819.851038	3.256485e+06	5.023898e+05	1259.688344
min	2020-02-15 00:00:00	1.500000e+01	0.000000	1.000000e+01	0.000000e+00	0.000000
25%	2020-09-06 18:00:00	6.667717e+06	33101.500000	1.192466e+06	1.975215e+05	589.000000

	date	cumulative_total_cases	daily_new_cases	active_cases	cumulative_total_deaths	daily_new_deaths
50%	2021-03-30 12:00:00	3.135999e+07	61092.000000	2.243084e+06	5.730690e+05	1022.000000
75%	2021-10-21 06:00:00	4.659445e+07	119769.000000	4.198612e+06	7.651068e+05	1742.000000
max	2022-05-14 00:00:00	8.420947e+07	909610.000000	1.793543e+07	1.026646e+06	4352.000000
std	NaN	2.615952e+07	135073.542354	3.399781e+06	3.201851e+05	900.680962

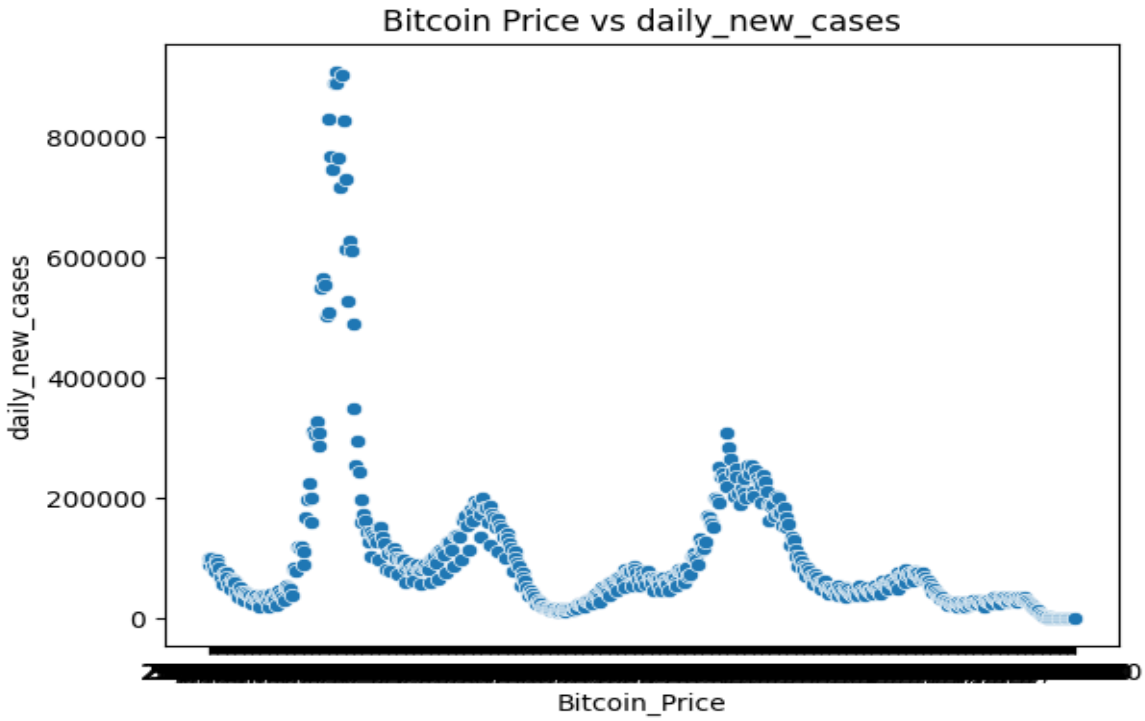
Now we need to merge our df1 with us_data to perform operations on them together. We use the proper code and name our new data frame merged_data.

From the us_data we only want to focus on daily_new_cases since it is the best indicator of the general state of the pandemic; whether it is accelarating or slowing down.

We plot a line plot using our merged_data to see how bitcoin_price and daily_new_cases changed according to each other.



Also a scatter plot for the same purpose.



Now we can move on to hypothesis testing.

Null Hypothesis: There is a significant correlation between Bitcoin_Price and daily_new_cases in the two year period between 2020-2022.

Alternative Hypothesis: There is no significant correlation between Bitcoin_Price and daily_new_cases in the two year period between 2020-2022.

Level of Significance = 0.05

If the p value from our test gives out a number lower than our significance level, that means there is actually a strong chance that our null hypothesis is true. Otherwise the alternative hypothesis should hold.

We get the following output from our pearsonr test.

```
Pearson correlation coefficient: 0.2123159029573104
P-value: 4.5673528364596627e-07
There is a significant correlation between Bitcoin_Price and
daily_new_cases.
```

Now we are going to create a linear regression model for our hypothesis.

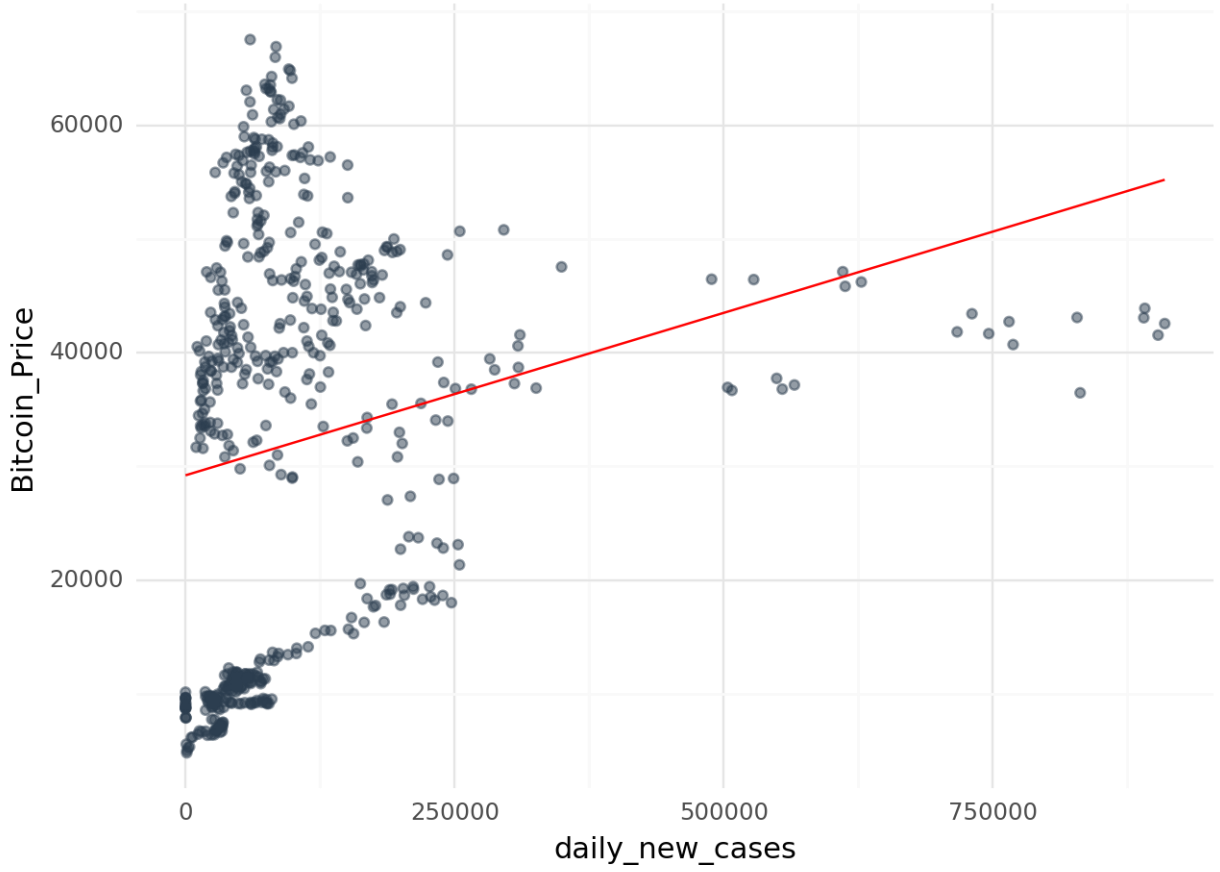
LinearRegression

```
LinearRegression()
```

We print coef and intercept values and an array of predictions for daily_new_cases. And the r2 score for daily_new_cases

```
0.045078042648577954
```

We print a new merged_data with the fitted values added according to our linear regression model and plot a graphic to depict a trend line between our two variables.



Machine Learning Models

To create machine learning models we have used Decision Tree regressor and Random Forest regressor models.

The decision tree regressor model gives the following mse and r-squared values.

```
Mean Squared Error: 83184.84981982007
R-squared: 0.9997312821187037
```

The random forest regressor model gives the following mse and r-squared values.

```
Mean Squared Error: 32124.69855540489
R-squared: 0.9998962253228588
```

MSE measures the average squared difference between the actual values (y_true) and the predicted values (y_pred) by the model. Lower MSE values indicate better model performance, with a perfect model having an MSE of 0. R-squared on the other hand is a statistical measure that represents the proportion of the variance in the dependent variable (target) that is explained by the independent variables (features) in the model. An r-squared value that is close to 1 explains the variability of the response data around its mean better compared to a value that is close to 0.

Since we know that a lower mse and higher r-squared value indicate better model performance, we can say that our random forest model performs better than the decision tree model considering it gives a lower MSE and a slightly higher r-squared value.