BITCOIN: SENTIMENT PRICE & ADOPTION

COREY RECAI, ERNEST SUBAH, WHELAN POWER, RACHEL SOLOMON

MOTIVATION

- Bitcoin is a relatively new "asset class" which many believe to have significant potential both as an investment and form of digital currency.

- Given the buzz surrounding Bitcoin we wanted to investigate the relationship between ***Bitcoin mentions in the news media***, ***Bitcoin price*** and ***Bitcoin adoption rates*** and develop a machine learning model to validate these relationships.

- In order to achieve this we have used ***natural language processing*** and ***predictive analytics***.

STRATEGY – LAYERED APPROACH

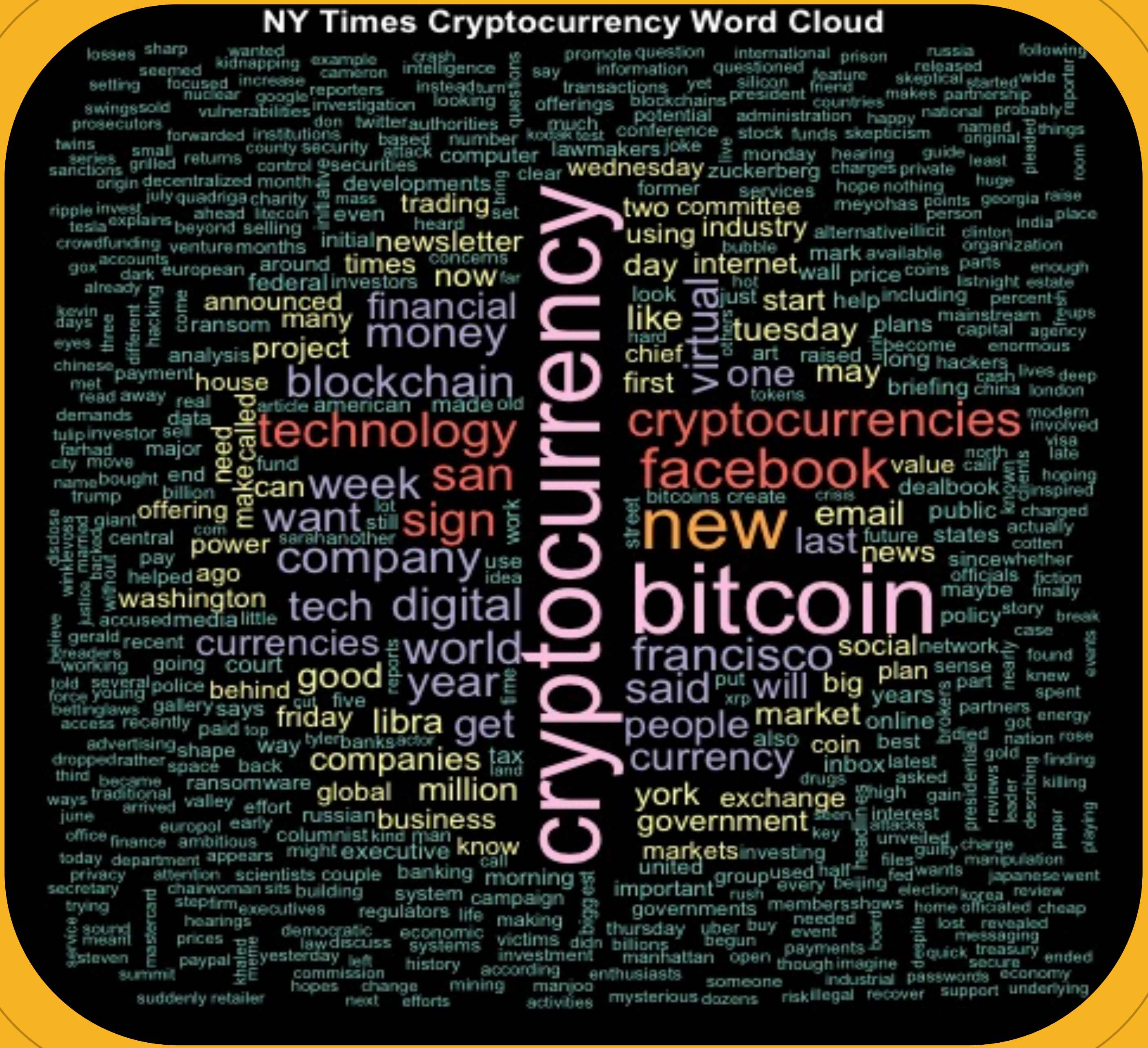
Natural Language Processing and **Predictive Analytics**.

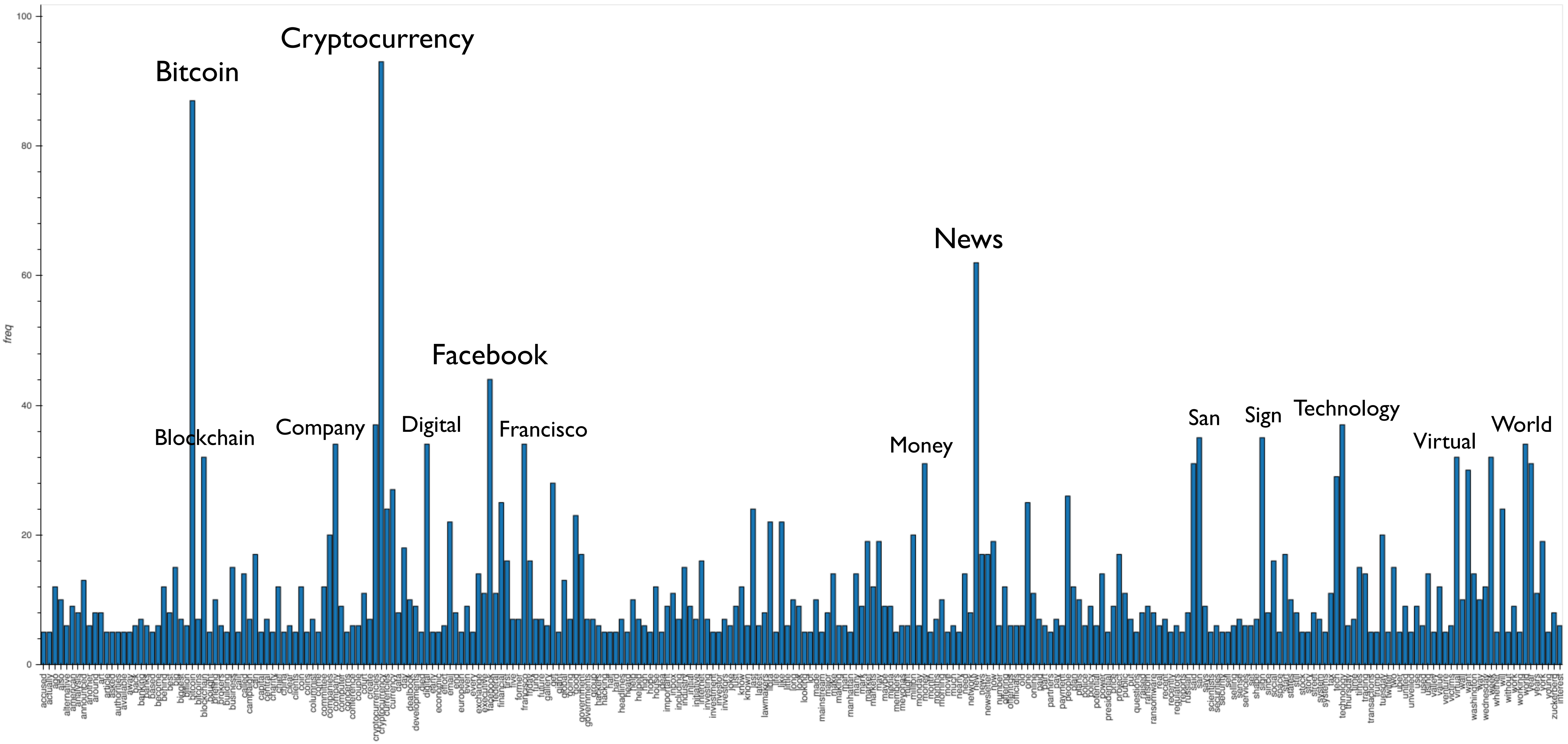
- Used the ***New York Times Articles API*** as a data source for Bitcoin news articles.
- Gathered Quandl data on ***Bitcoin price*** and ***Bitcoin adoption rate*** (using volume as a proxy).
- Used ***keyword analysis*** to analyze Bitcoin mentions in mainstream media.
- Incorporated ***sentiment analysis*** to determine whether statements made about Bitcoin were positive or negative.
- Evaluated and correlated data to determine the relationship between ***Bitcoin news sentiment***, ***Bitcoin price***, and ***Bitcoin adoption***.
- Created a ***classification report/confusion matrix*** based on Bitcoin adoption data.

PARAMETERS & DATA COLLECTION

- Article Source: ***New York Times Articles API***
- Time Period: ***2013-2020***
- Adoption Rate Proxy: ***Bitcoin Trading Volume***
- Bitcoin Price: ***Quandl Data***

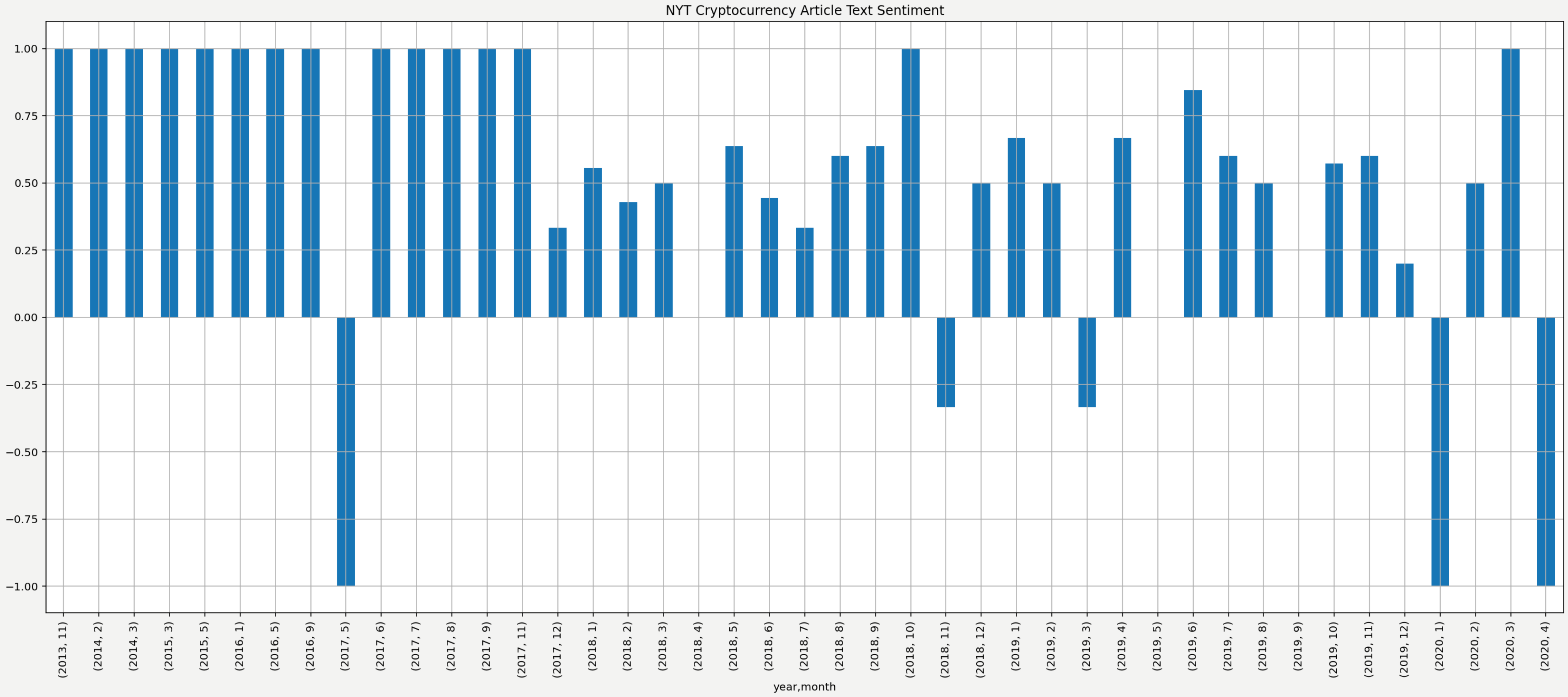
KEY WORD ANALYSIS



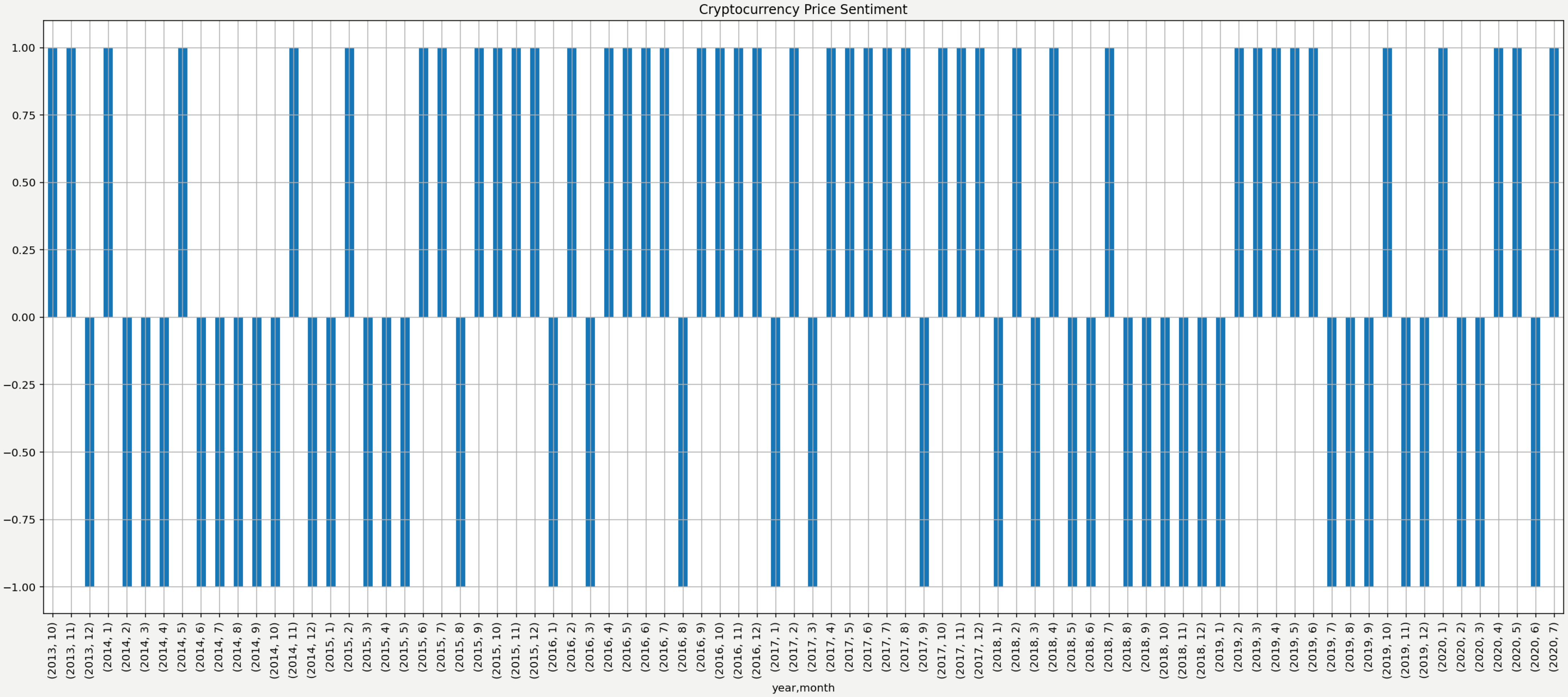


MOST FREQUENT WORDS

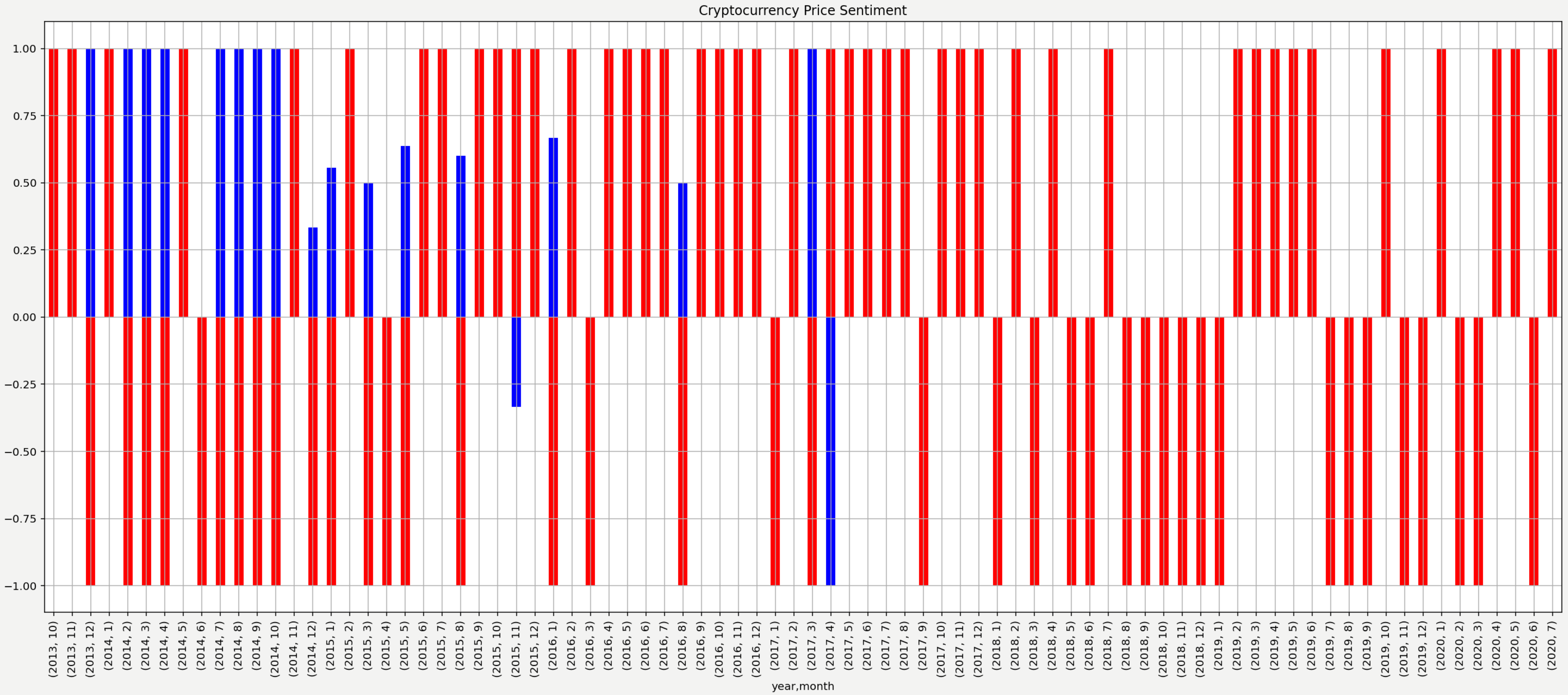
MONTHLY AVG ARTICLE SENTIMENT



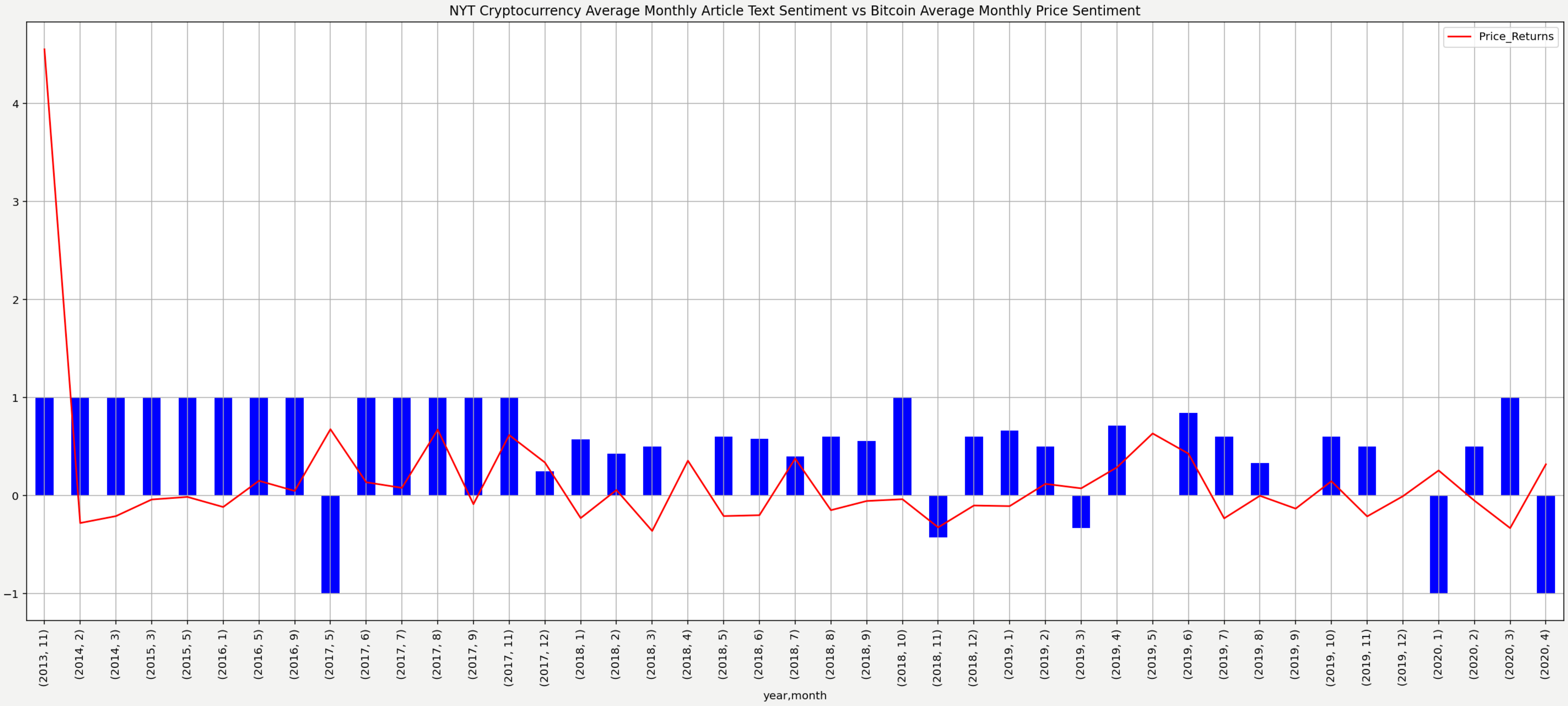
MONTHLY PRICE SENTIMENT



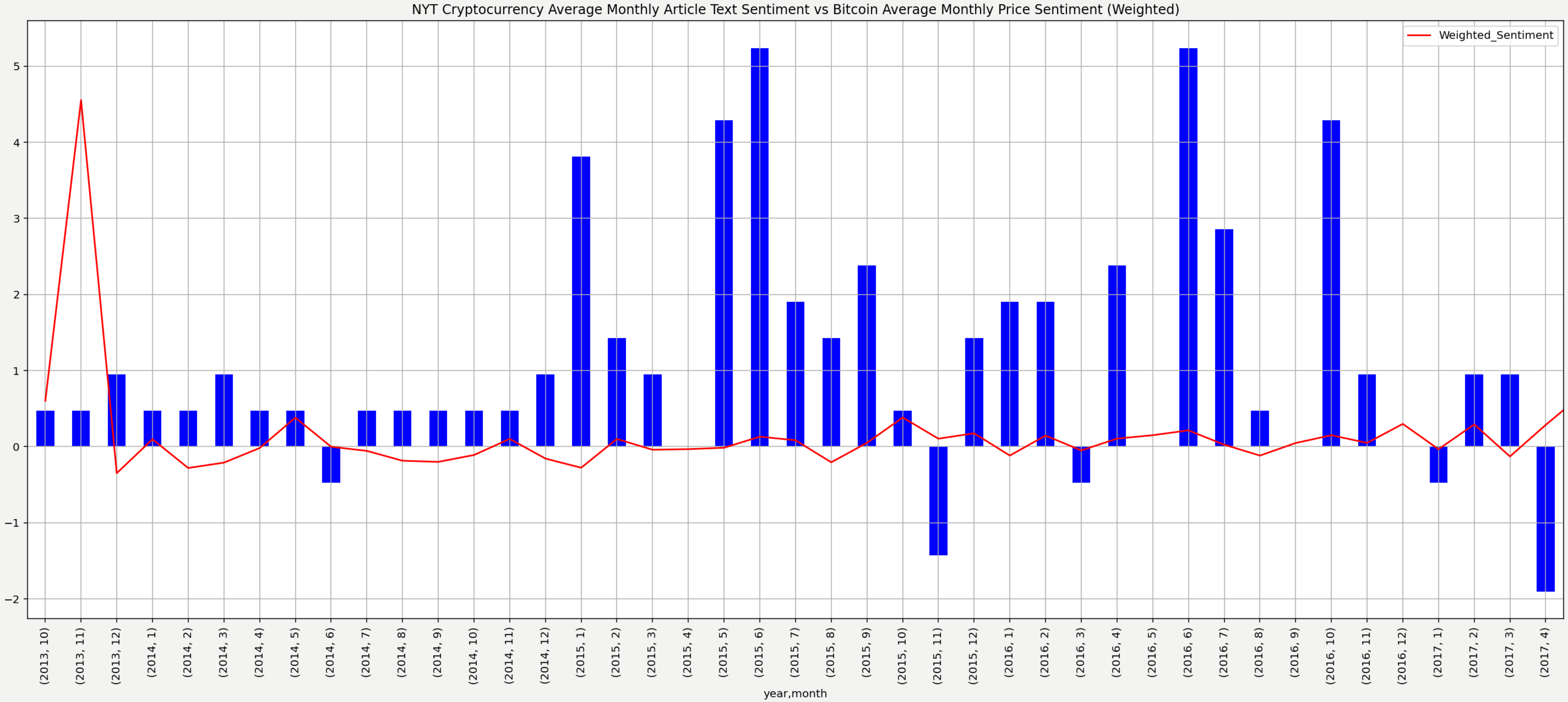
MONTHLY AVG PRICE V. MONTHLY AVG ARTICLE SENTIMENT



ARTICLE SENTIMENT V. PRICE SENTIMENT



ARTICLE SENTIMENT V. PRICE SENTIMENT (WEIGHTED)



STATISTICS & PREDICTIVE ANALYTICS

This section explores the relationships between Bitcoin Price/Price Returns, Adoption Rate and perceived sentiment (Positive & Negative) as classification values.

Statistical Analysis:

- Explored the relationship between Bitcoin Price Returns and Bitcoin Adoption Rate
- Normal Gaussian Curves for Bitcoin Price Returns and Bitcoin Adoption Rate
- Volatility of Returns and Adoption Rate
- Correlation Coefficient

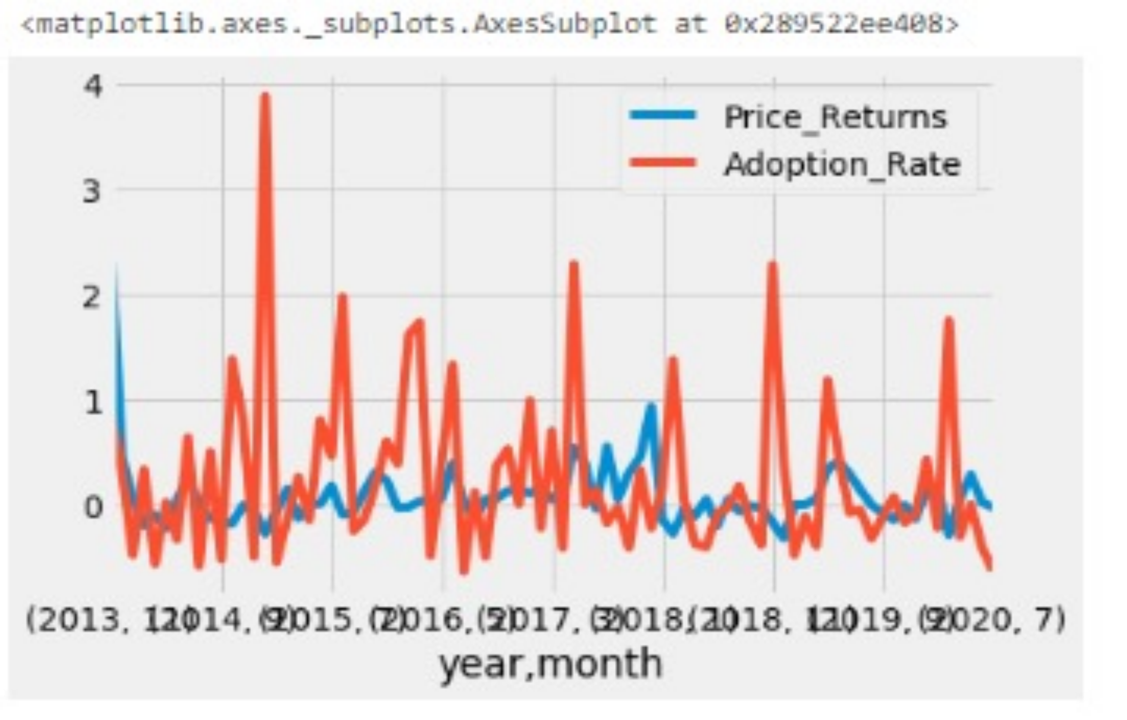
Predictive Analytics / Logistic Regression:

- Explored the Relationship between Bitcoin Price Return and Sentiments developed from the Adoption Rate (outcomes = + for up, - for down)
- Confusion Matrix and Classification Report

STATISTICAL ANALYSIS: *DESCRIPTIVE STATS*

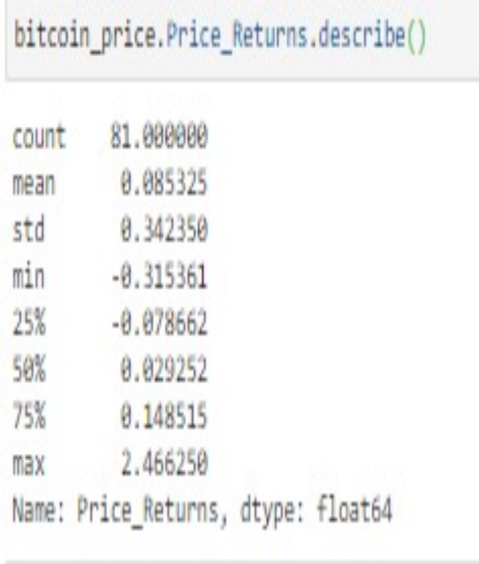
Bitcoin Adoption Rate showed more volatility (80%) when compared to Price Returns (34%) over the same period of time.

Price Returns vs Adoption Rate (11/2013 – 6/2020)



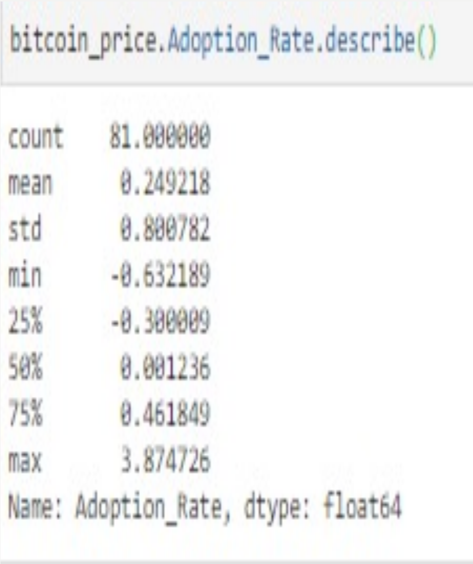
```
<matplotlib.axes._subplots.AxesSubplot at 0x289522ee408>
```

Price Returns



```
bitcoin_price.Price_Returns.describe()

count    81.000000
mean      0.085325
std       0.342350
min      -0.315361
25%      -0.078662
50%       0.029252
75%       0.148515
max       2.466250
Name: Price_Returns, dtype: float64
```

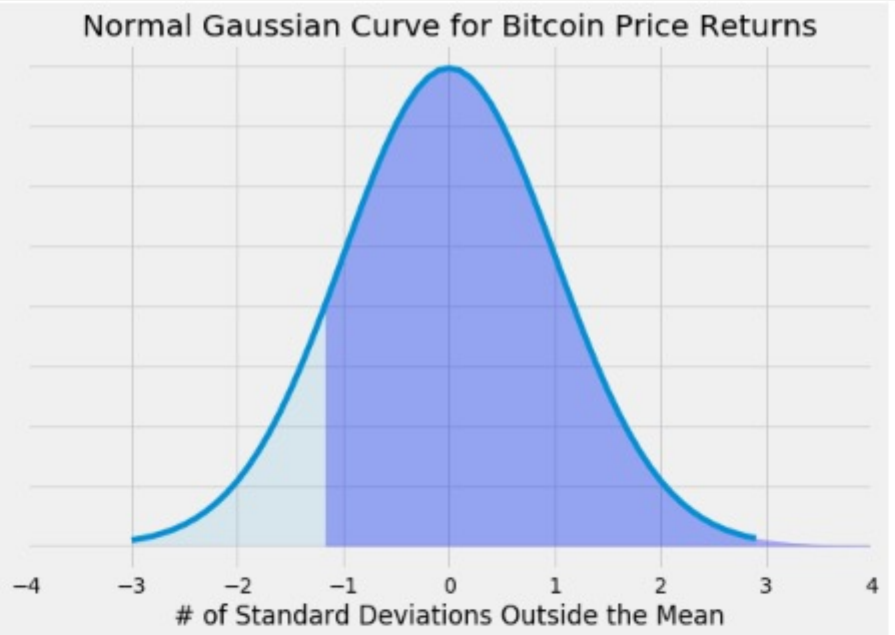
Adoption Rate



```
bitcoin_price.Adoption_Rate.describe()

count    81.000000
mean      0.249218
std       0.800782
min      -0.632189
25%      -0.300009
50%       0.001236
75%       0.461849
max       3.874726
Name: Adoption_Rate, dtype: float64
```

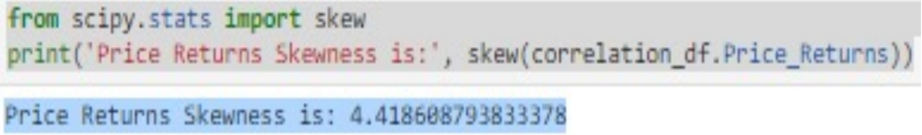
STATISTICAL ANALYSIS: *GAUSSIAN CURVE*

Bitcoin Price Returns



Skewness (*from scipy.stats import skew*)

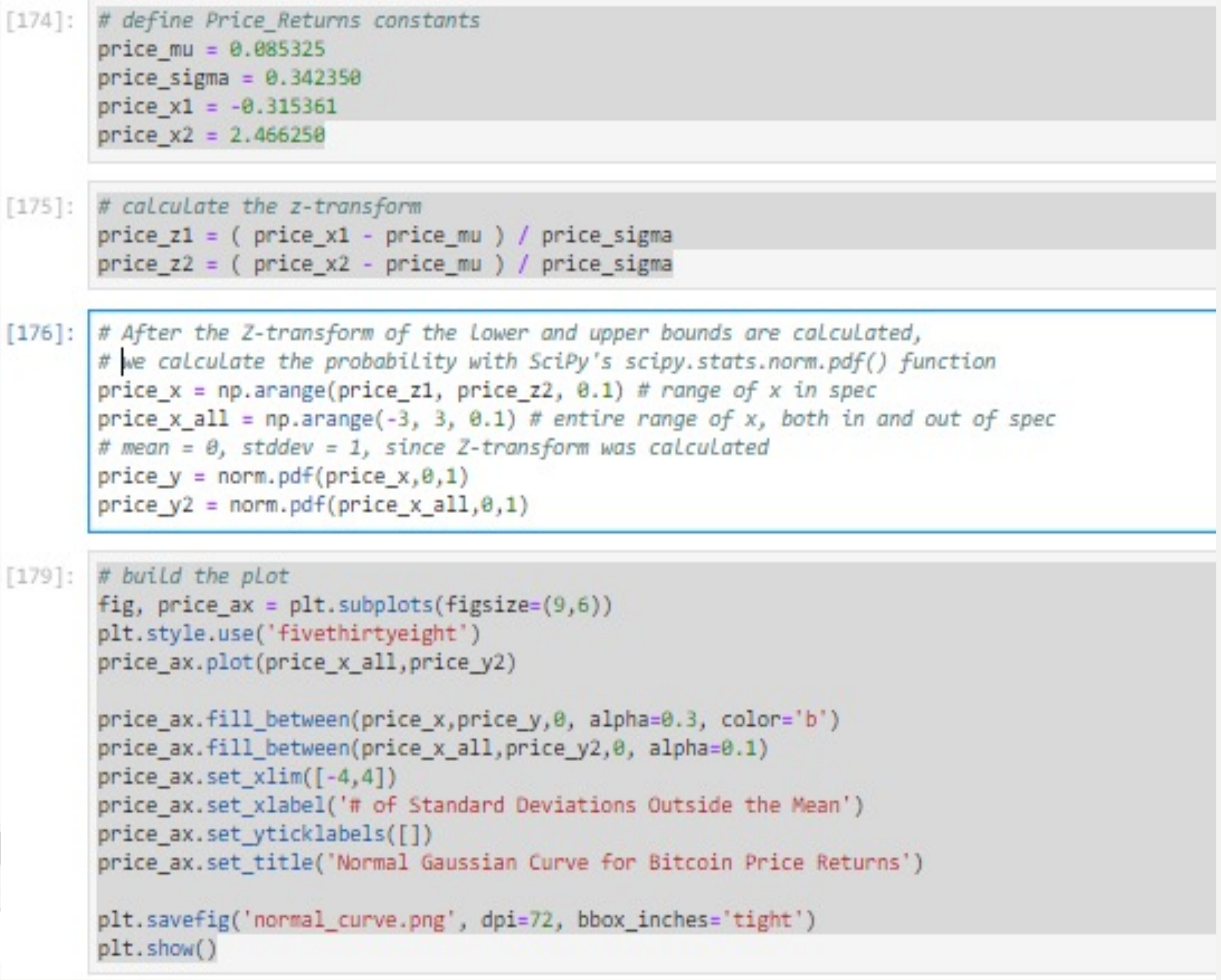
- The data is not symmetrical along the x axis
- The data is positively skewed (> 0)
- Price Returns Skewness (4.42) > Adoption Rate Skewness (1.85)



```
from scipy.stats import skew
print('Price Returns Skewness is:', skew(correlation_df.Price_Returns))
```

```
Price Returns Skewness is: 4.418608793833378
```

Code



```
[174]: # define Price_Returns constants
       price_mu = 0.085325
       price_sigma = 0.342350
       price_x1 = -0.315361
       price_x2 = 2.466250
```

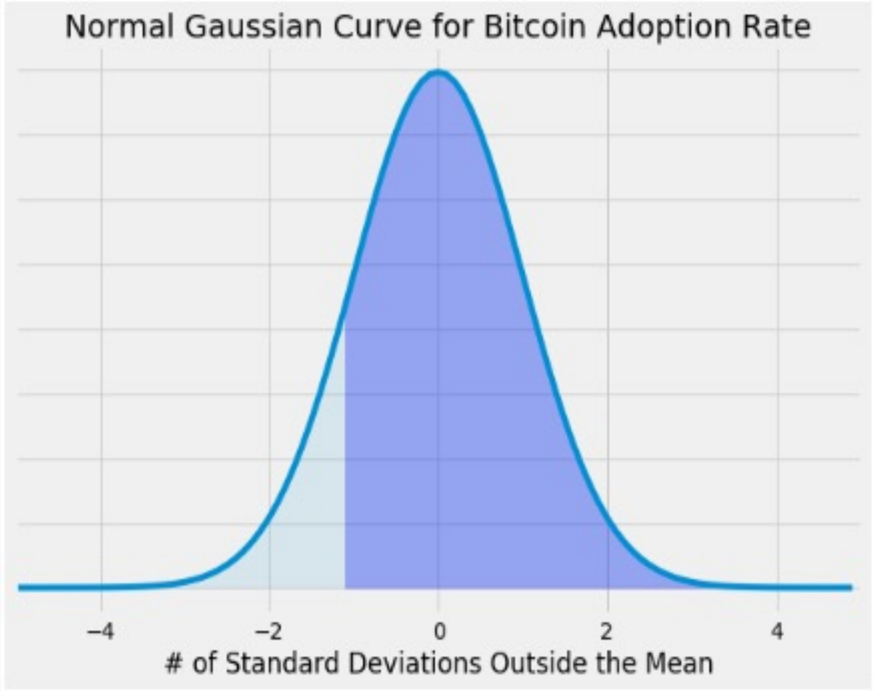
```
[175]: # calculate the z-transform
       price_z1 = ( price_x1 - price_mu ) / price_sigma
       price_z2 = ( price_x2 - price_mu ) / price_sigma
```

```
[176]: # After the Z-transform of the lower and upper bounds are calculated,
       # we calculate the probability with SciPy's scipy.stats.norm.pdf() function
       price_x = np.arange(price_z1, price_z2, 0.1) # range of x in spec
       price_x_all = np.arange(-3, 3, 0.1) # entire range of x, both in and out of spec
       # mean = 0, stddev = 1, since Z-transform was calculated
       price_y = norm.pdf(price_x,0,1)
       price_y2 = norm.pdf(price_x_all,0,1)
```

```
[179]: # build the plot
       fig, price_ax = plt.subplots(figsize=(9,6))
       plt.style.use('fivethirtyeight')
       price_ax.plot(price_x_all,price_y2)

       price_ax.fill_between(price_x,price_y,0, alpha=0.3, color='b')
       price_ax.fill_between(price_x_all,price_y2,0, alpha=0.1)
       price_ax.set_xlim([-4,4])
       price_ax.set_xlabel('# of Standard Deviations Outside the Mean')
       price_ax.set_yticklabels([])
       price_ax.set_title('Normal Gaussian Curve for Bitcoin Price Returns')

       plt.savefig('normal_curve.png', dpi=72, bbox_inches='tight')
       plt.show()
```

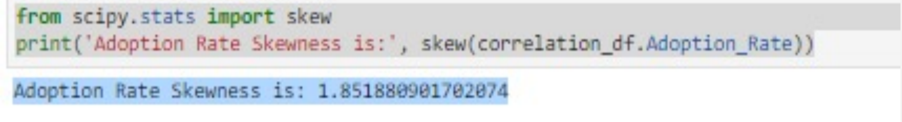
STATISTICAL ANALYSIS: *GAUSSIAN CURVE*

Bitcoin Adoption Rate



Skewness (*from scipy.stats import skew*)

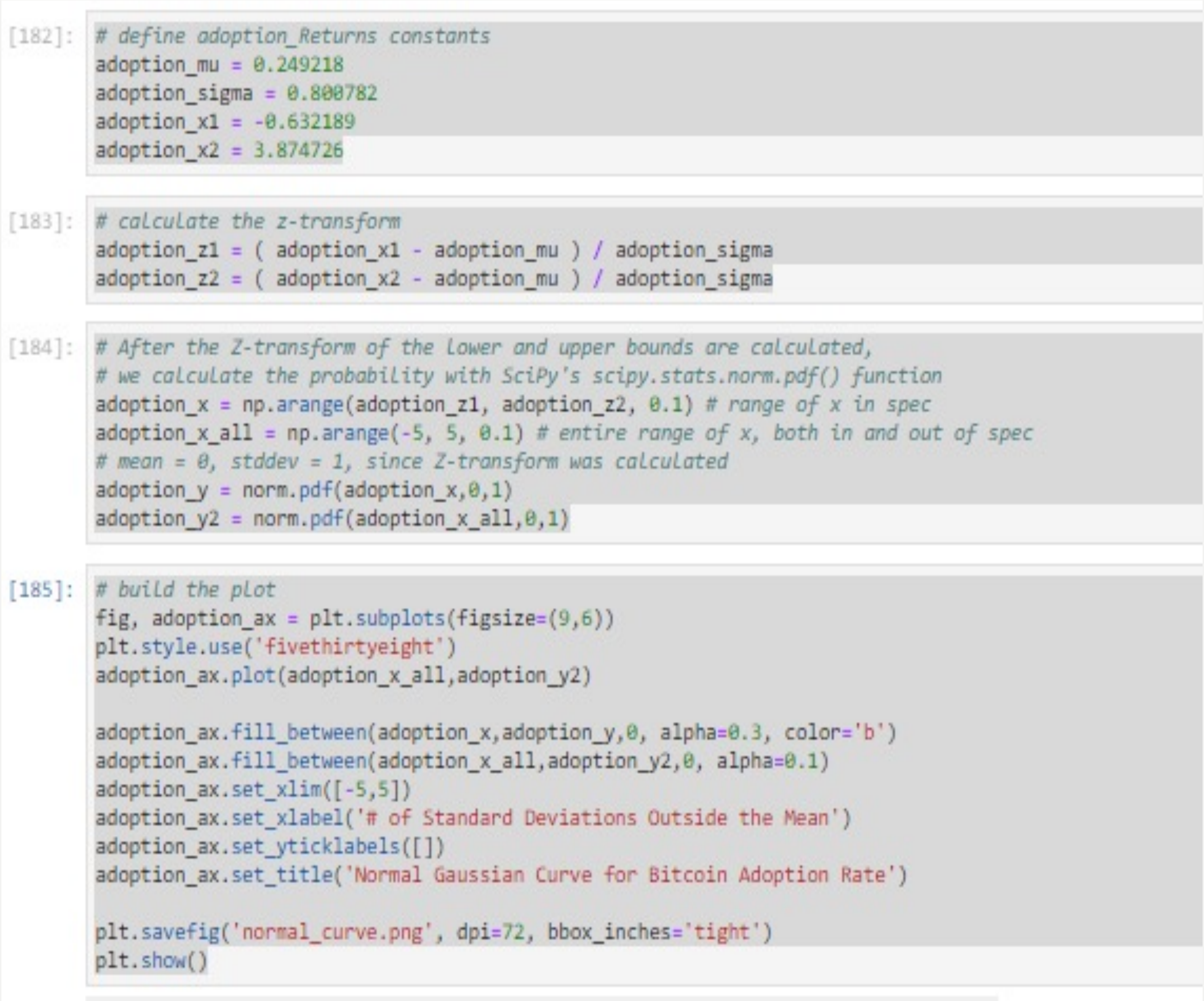
- The data is not symmetrical along the x axis
- The data is positively skewed (> 0)
- Price Returns Skewness (4.42) > Adoption Rate Skewness (1.85)



```
from scipy.stats import skew
print('Adoption Rate Skewness is:', skew(correlation_df.Adoption_Rate))
```

```
Adoption Rate Skewness is: 1.851880901702074
```

Code



```
[182]: # define adoption_Returns constants
       adoption_mu = 0.249218
       adoption_sigma = 0.800782
       adoption_x1 = -0.632189
       adoption_x2 = 3.874726
```

```
[183]: # calculate the z-transform
       adoption_z1 = ( adoption_x1 - adoption_mu ) / adoption_sigma
       adoption_z2 = ( adoption_x2 - adoption_mu ) / adoption_sigma
```

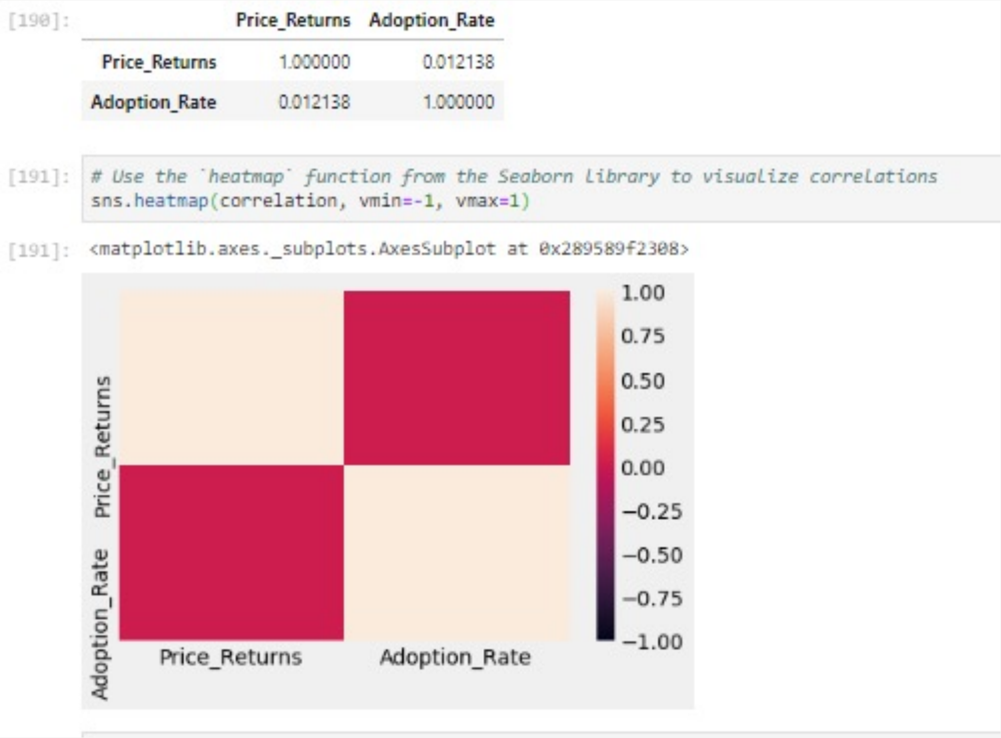
```
[184]: # After the Z-transform of the lower and upper bounds are calculated,
       # we calculate the probability with SciPy's scipy.stats.norm.pdf() function
       adoption_x = np.arange(adoption_z1, adoption_z2, 0.1) # range of x in spec
       adoption_x_all = np.arange(-5, 5, 0.1) # entire range of x, both in and out of spec
       # mean = 0, stddev = 1, since Z-transform was calculated
       adoption_y = norm.pdf(adoption_x,0,1)
       adoption_y2 = norm.pdf(adoption_x_all,0,1)
```

```
[185]: # build the plot
       fig, adoption_ax = plt.subplots(figsize=(9,6))
       plt.style.use('fivethirtyeight')
       adoption_ax.plot(adoption_x_all,adoption_y2)

       adoption_ax.fill_between(adoption_x,adoption_y,0, alpha=0.3, color='b')
       adoption_ax.fill_between(adoption_x_all,adoption_y2,0, alpha=0.1)
       adoption_ax.set_xlim([-5,5])
       adoption_ax.set_xlabel('# of Standard Deviations Outside the Mean')
       adoption_ax.set_yticklabels([])
       adoption_ax.set_title('Normal Gaussian Curve for Bitcoin Adoption Rate')

       plt.savefig('normal_curve.png', dpi=72, bbox_inches='tight')
       plt.show()
```

STATISTICAL ANALYSIS: *CORRELATION*

- Our analysis showed a very low, almost no, correlation between Bitcoin Price Returns and Adoption Rate
- Correlation was calculated at 0.012138
- The scatter plot shows a lot of clustering of the data around zero which is reflective of the correlation coefficient

Correlation



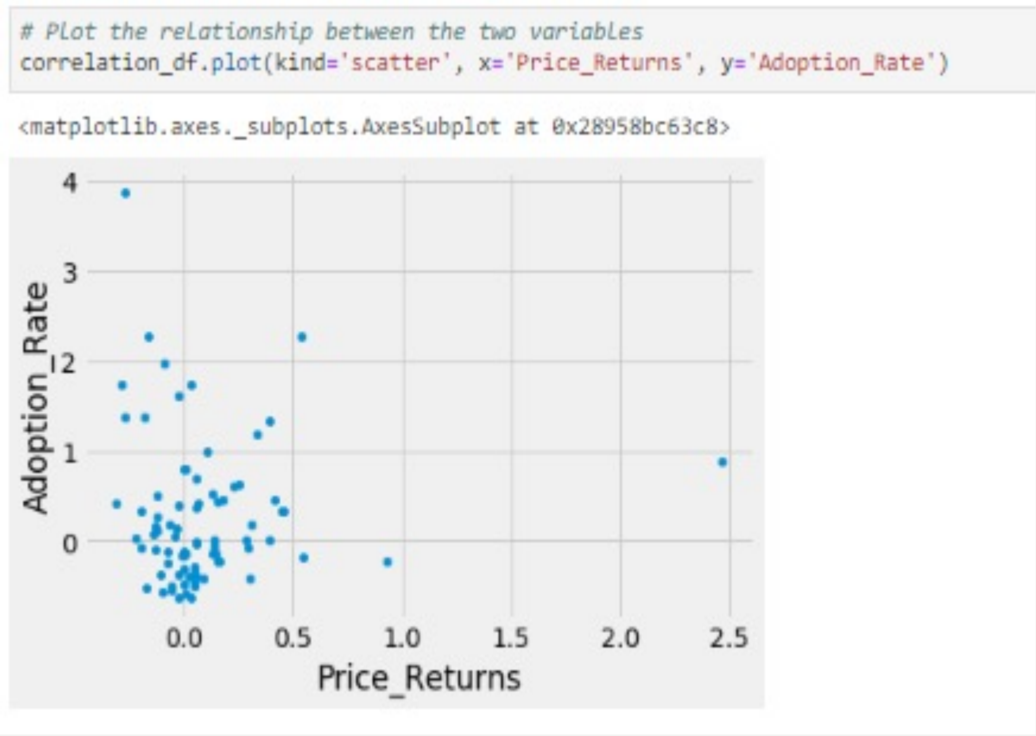
[190]:

	Price_Returns	Adoption_Rate
Price_Returns	1.000000	0.012138
Adoption_Rate	0.012138	1.000000

```
[191]: # Use the `heatmap` function from the Seaborn library to visualize correlations
       sns.heatmap(correlation, vmin=-1, vmax=1)
```

```
[191]: <matplotlib.axes._subplots.AxesSubplot at 0x289589f2308>
```

Clustering



```
# Plot the relationship between the two variables
correlation_df.plot(kind='scatter', x='Price_Returns', y='Adoption_Rate')
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x28958bc63c8>
```

PREDICTIVE ANALYTICS: *LOGISTIC REGRESSION*

- Explored the Relationship between Bitcoin Price Return and Sentiments developed from the Adoption Rate (Target Class = + for up, - for down)

- Developed two target classes of outcome (Positive and Negative) to reflect the increase (decrease) in quantity of bitcoin being adopted as an investment asset

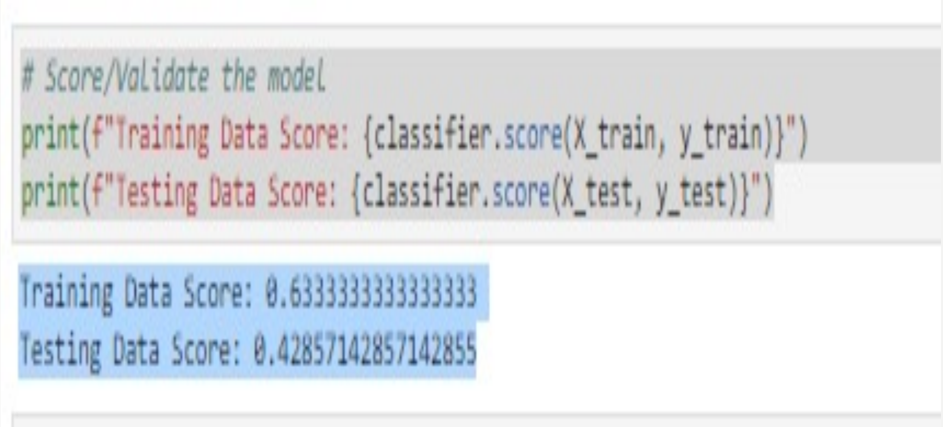
- Used Logistic Regression to create a classification model

- Confusion Matrix and Classification Report

PREDICTIVE ANALYTICS: *LOGISTIC REGRESSION*

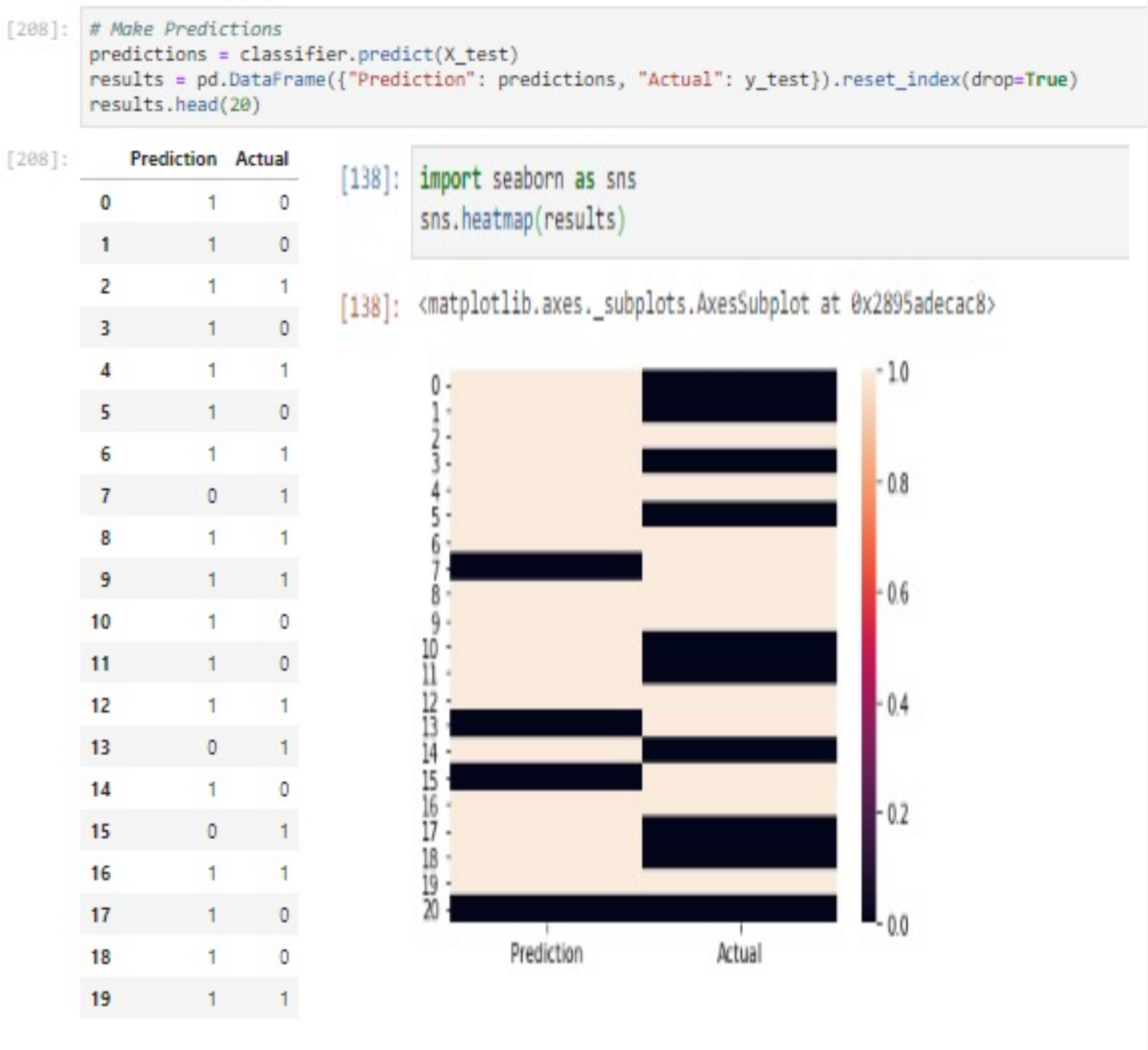
Training and Testing Dataset

- y = Adoption Sentiment
  - y>=0 → Positive (outcome = +1)
  - y<0 → Negative (outcome = -1)
- X = Closing Price and pct_change



```
# Score/Validate the model
print(f"Training Data Score: {classifier.score(X_train, y_train)}")
print(f"Testing Data Score: {classifier.score(X_test, y_test)}")
```

```
Training Data Score: 0.6333333333333333
Testing Data Score: 0.42857142857142855
```



```
[208]: # Make Predictions
       predictions = classifier.predict(X_test)
       results = pd.DataFrame({"Prediction": predictions, "Actual": y_test}).reset_index(drop=True)
       results.head(20)
```

[208]:

	Prediction	Actual
0	1	0
1	1	0
2	1	1
3	1	0
4	1	1
5	1	0
6	1	1
7	0	1
8	1	1
9	1	1
10	1	0
11	1	0
12	1	1
13	0	1
14	1	0
15	0	1
16	1	1
17	1	0
18	1	0
19	1	1

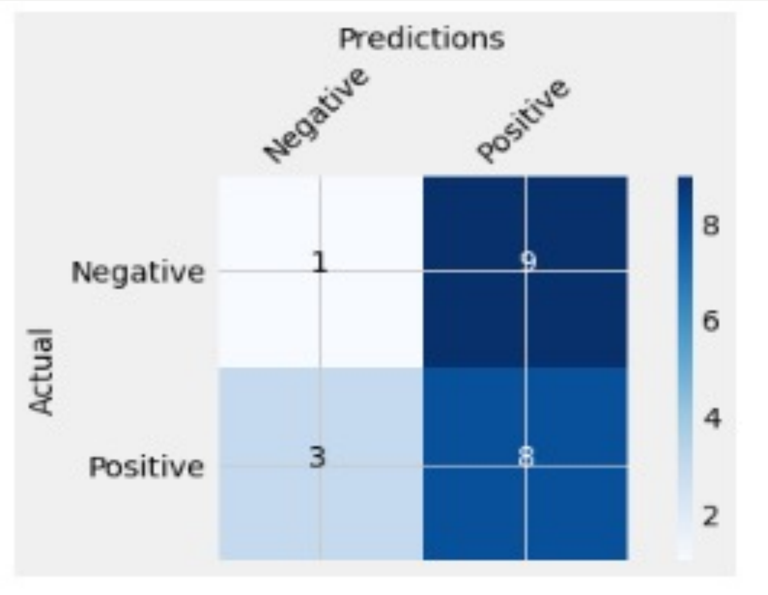
```
[138]: import seaborn as sns
       sns.heatmap(results)
```

```
[138]: <matplotlib.axes._subplots.AxesSubplot at 0x2895adecac8>
```

PREDICTIVE ANALYTICS: *LOGISTIC REGRESSION*

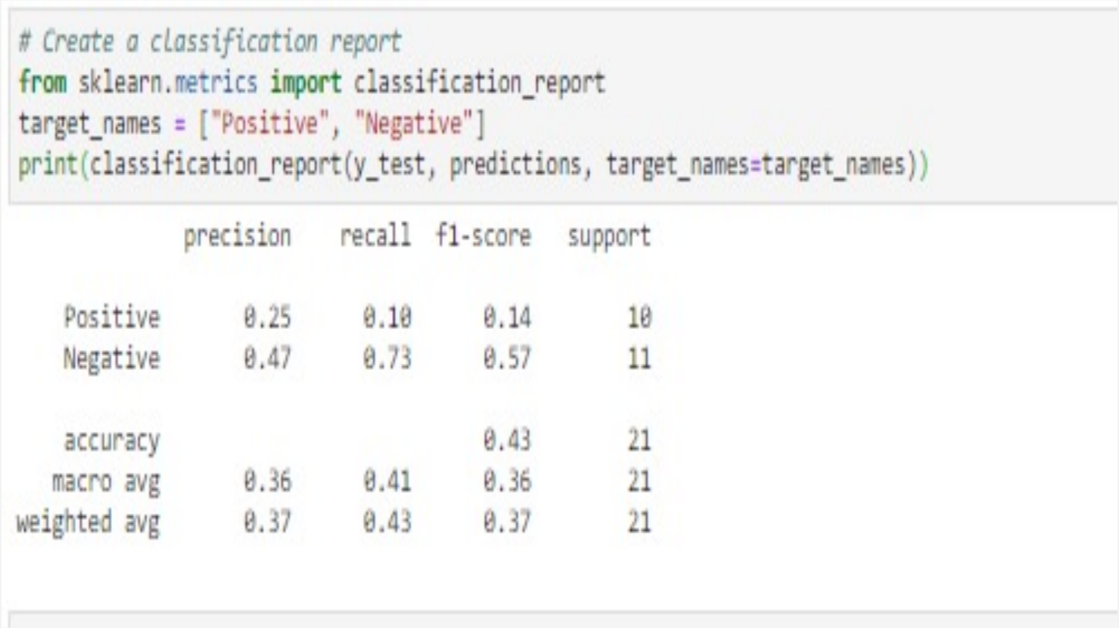
Confusion Matrix

- N = 21 = TP + TN + FP + FN = 1+8+9+3 = 21
- TP = 8 (Actual=Predicted=Positive)
- TN = 1 (Actual=Predicted=Negative)
- FP/Type I Error = 9 (Predicted=+ve, Actual=-ve)
- FN/Type II Error = 3 (Predictive=-ve, Actual=+ve)



Classification Report

- Precision = TP/(TP+FP) = 8/(8+9)=8/17=47%
- Recall = TP/(TP+FN) = 8/(8+3) = 8/11 = 73%
- F1 Score = 2[(Precision*Recall)/(Precision+Recall) = 2[(0.47*0.73)/(0.47+0.73)] = 2*(0.342/1.198) = 57.14%



```
# Create a classification report
from sklearn.metrics import classification_report
target_names = ["Positive", "Negative"]
print(classification_report(y_test, predictions, target_names=target_names))
```

```
              precision    recall  f1-score   support

    Positive       0.25      0.10      0.14        10
    Negative       0.47      0.73      0.57        11

    accuracy                           0.43        21
   macro avg       0.36      0.41      0.36        21
weighted avg       0.37      0.43      0.37        21
```

CONCLUSIONS

- Based on our analysis there is not a strong relationship between, ***Bitcoin sentiment in the news media***, ***Bitcoin price*** and overall ***Bitcoin adoption***.
  - High adoption rate and a significant media coverage do not indicate a trading opportunity.
- While Bitcoin price is volatile (SD=34%), Bitcoin adoption is more volatile (SD=80%)
- Logistic Regression Model did not show a strong relationship between adoption rate and sentiment.

FORWARD THOUGHTS

- Pull in data from additional sources for a more well-rounded data sentiment.
  - NY Times has a generally positive bias towards Bitcoin.

- Explore variables that potentially have stronger correlations to Bitcoin sentiment or price.