

# Impact of Covid-19 on the Stock Market

The goal of this project was to highlight the impact of Covid-19 on the stock market, and make predictions on future stock prices.

## LINEAR REGRESSION

```
[ ] np.exp(test[['returns','strat_LR']].sum())
returns      1.377365
strat_LR    1.052226
dtype: float64
```

```
▶ (test['direction_LR'].diff() != 0).value_counts()
False     157
True     137
Name: direction_LR, dtype: int64
```

```
▶ p1 = np.exp(test[['returns','strat_LR']].cumsum())
p1.plot()
plt.show()
```

## LOGISTICAL REGRESSION

```
[43] x_train1 = df[df.Date.dt.year < 2021][['const', 'Lag 1', 'Lag 2']]
y_train1 = df[df.Date.dt.year < 2021]['Direction']
x_test1 = df[df.Date.dt.year == 2021][['const', 'Lag 1', 'Lag 2']]
y_test1 = df[df.Date.dt.year == 2021]['Direction']
```

```
▶ model1 = sm.Logit(y_train1, x_train1)
result1 = model1.fit()
prediction1 = result1.predict(x_test1)
confusion_matrix(y_test1, prediction1)
```

```
Optimization terminated successfully.
Current function value: 0.683249
Iterations 4
```

	Predicted	Down	Up
Actual	Down	10	108
	Up	8	126

```
(10+126)/len(x_test1)
```

```
0.5396825396825397
```

```
[35] x_train = df[df.Date.dt.year < 2021][['const', 'Lag 1', 'Lag 2', 'Lag 3', 'Lag 4', 'Lag 5', 'Volume']]
y_train = df[df.Date.dt.year < 2021]['Direction']
x_test = df[df.Date.dt.year == 2021][['const', 'Lag 1', 'Lag 2', 'Lag 3', 'Lag 4', 'Lag 5', 'Volume']]
y_test = df[df.Date.dt.year == 2021]['Direction']
```

```
[36] model = sm.Logit(y_train, x_train)
```

```
[37] result = model.fit()
Optimization terminated successfully.
Current function value: 0.680976
Iterations 4
```

```
[38] prediction = result.predict(x_test)
```

```
▶ confusion_matrix(y_test, prediction)
```

	Predicted	Down	Up
Actual	Down	13	105
	Up	9	125

```
[42] (13+125)/len(x_test)
```

```
0.5476190476190477
```

## LONG-SHORT-TERM-MEMORY MODEL

```
#Predicted vs True Adj Close Value - LSTM
plt.plot(y_test, label='True Value')
plt.plot(y_pred, label='LSTM Value')
plt.title("Prediction by LSTM")
plt.xlabel('Time Scale')
plt.ylabel('Scaled USD')
plt.legend()
plt.show()
```