Finding the Best Fitting Model for Forecasting of Tesla Stocks

-Eeshanya Joshi

BACKGROUND

The stock market is influenced by various factors, making accurate forecasting essential for investors.

This analysis focuses on the key differences between a few select models by comparing two models at a time when they are used to predict Tesla's stock price using the methodology given in Value of Information in the 'Mean-Square Case and its Application to the Analysis of Financial Time-Series Forecast' by Roman V. Belavkin et. al.

RESEARCH PAPER SUMMARY

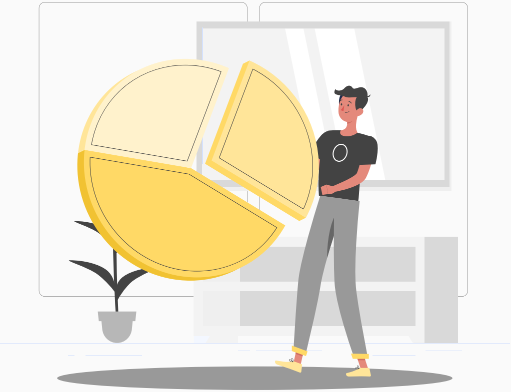
The paper discusses **Value of Information (VoI)** as a method to guide decisions in data analytics, particularly in selecting models and assessing the value of additional data.

Mutual Information: VoI measures the reduction of uncertainty about a response variable based on knowledge of predictors, helping to evaluate model effectiveness.

Theoretical Limits: VoI establishes upper and lower performance bounds based on mutual information, guiding analysts on the minimum information needed for desired outcomes.

Mathematical Components:

1. Shannon's Entropy
2. Conditional Entropy
3. Mutual Information
4. VoI Expression
5. Cost-Benefit Analysis



In the context of forecasting models, we shall be calculating **VoI** as the **difference in RMSE** between the models.

Dataset Used: TSLA.csv

Columns: Date, Open, High, Low, Close, Adj Close, Volume

Data Range: 29th June, 2010 to 24th March, 2022

DATASET OVERVIEW

Source: https://www.kaggle.com/datasets/varpit94/tesla-stock-data-updated-till-28jun2021

1	Date	Open	High	Low	Close	Adj Close	Volume
2	29-06-2010	3.8	5	3.508	4.778	4.778	93831500
3	30-06-2010	5.158	6.084	4.66	4.766	4.766	85935500
4	01-07-2010	5	5.184	4.054	4.392	4.392	41094000
5	02-07-2010	4.6	4.62	3.742	3.84	3.84	25699000
6	06-07-2010	4	4	3.166	3.222	3.222	34334500
7	07-07-2010	3.28	3.326	2.996	3.16	3.16	34608500
8	08-07-2010	3.228	3.504	3.114	3.492	3.492	38557000
9	09-07-2010	3.516	3.58	3.31	3.48	3.48	20253000
10	12-07-2010	3.59	3.614	3.4	3.41	3.41	11012500
11	13-07-2010	3.478	3.728	3.38	3.628	3.628	13400500
12	14-07-2010	3.588	4.03	3.552	3.968	3.968	20976000
13	15-07-2010	3.988	4.3	3.8	3.978	3.978	18699000
14	16-07-2010	4.14	4.26	4.01	4.128	4.128	13106500
15	19-07-2010	4.274	4.45	4.184	4.382	4.382	12432500
16	20-07-2010	4.37	4.37	4.01	4.06	4.06	9126500
17	21-07-2010	4.132	4.18	3.9	4.044	4.044	6262500
18	22-07-2010	4.1	4.25	4.074	4.2	4.2	4789000
19	23-07-2010	4.238	4.312	4.212	4.258	4.258	3268000
20	26-07-2010	4.3	4.3	4.06	4.19	4.19	4611000
21	27-07-2010	4.182	4.236	4.052	4.11	4.11	3098500
22	28-07-2010	4.11	4.18	4.102	4.144	4.144	2336000
23	29-07-2010	4.154	4.176	4	4.07	4.07	3080000
24	30-07-2010	4.04	4.088	3.91	3.988	3.988	2134500
25	02-08-2010	4.1	4.194	4.066	4.184	4.184	3590500
26	03-08-2010	4.2	4.39	4.164	4.39	4.39	6152500

MODELS

01
VAR

VAR (**Vector AutoRegression**) is a **multivariate** time series forecasting method that captures **linear interdependencies** among multiple time series, making it useful for analyzing relationships between interrelated time series

02
ARIMA

ARIMA or **AutoRegressive Integrated Moving Average** is a popular statistical method used for time series forecasting. It is best for **single, non-seasonal** time series data.

03
SARIMAX

SARIMAX is an **enhancement** for ARIMA using **seasonal** time series data, allowing for external variables.

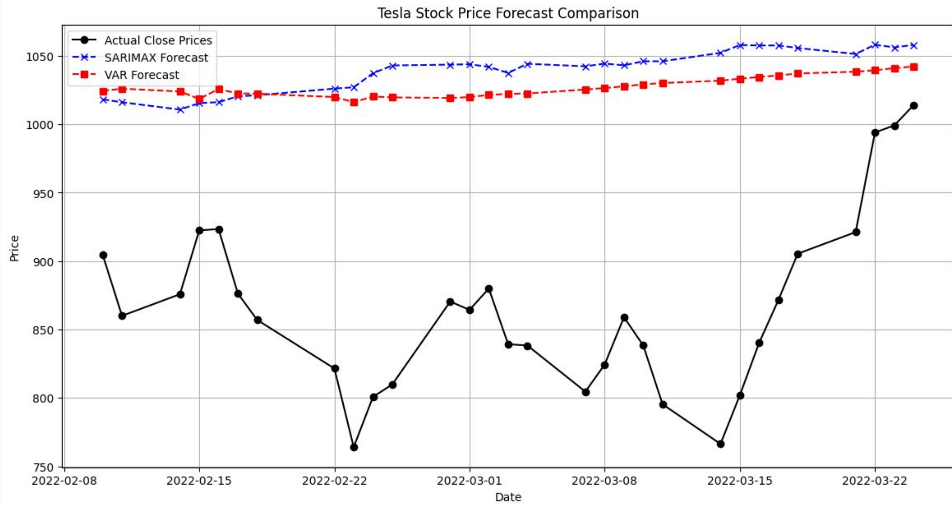
IMPLEMENTATION & RESULTS

```python
import pandas as pd
import numpy as np
from statsmodels.tsa.statespace.sarimax import SARIMAX
from statsmodels.tsa.api import VAR
from sklearn.metrics import mean_squared_error
import matplotlib.pyplot as plt

data = pd.read_csv('TSLA.csv')
data['Date'] = pd.to_datetime(data['Date'])
data.set_index('Date', inplace=True)

data = data[['Close', 'Volume']].dropna()

baseline_model = SARIMAX(data['Close'], order=(5, 1, 0), seasonal_order=(1, 1, 1, 12))
baseline_fit = baseline_model.fit()

baseline_forecast = baseline_fit.forecast(steps=30)

baseline_rmse = np.sqrt(mean_squared_error(data['Close'].iloc[-30:], baseline_forecast))

var_data = data[['Close', 'Volume']].dropna()

var_model = VAR(var_data)
var_fit = var_model.fit(maxlags=15, ic='aic')

var_forecast = var_fit.forecast(var_data.values[-var_fit.k_ar:], steps=30)

var_forecast_close = var_forecast[:, 0]

actual_close_prices = data['Close'].iloc[-30:].values
new_rmse = np.sqrt(mean_squared_error(actual_close_prices, var_forecast_close))

voi = baseline_rmse - new_rmse
print(f'SARIMAX RMSE: {baseline_rmse}')
print(f'VAR RMSE: {new_rmse}')
print(f'Value of Information (VoI): {voi}')
```

```python
plt.figure(figsize=(14, 7))
plt.plot(data.index[-30:], actual_close_prices, label='Actual Close Prices', color='black', marker='o')
plt.plot(data.index[-30:], baseline_forecast, label='SARIMAX Forecast', color='blue', linestyle='--', marker='x')
plt.plot(data.index[-30:], var_forecast_close, label='VAR Forecast', color='red', linestyle='--', marker='s')
plt.title('Tesla Stock Price Forecast Comparison')
plt.xlabel('Date')
plt.ylabel('Price')
plt.legend()
plt.grid()
plt.show()
```

SARIMAX VS VAR

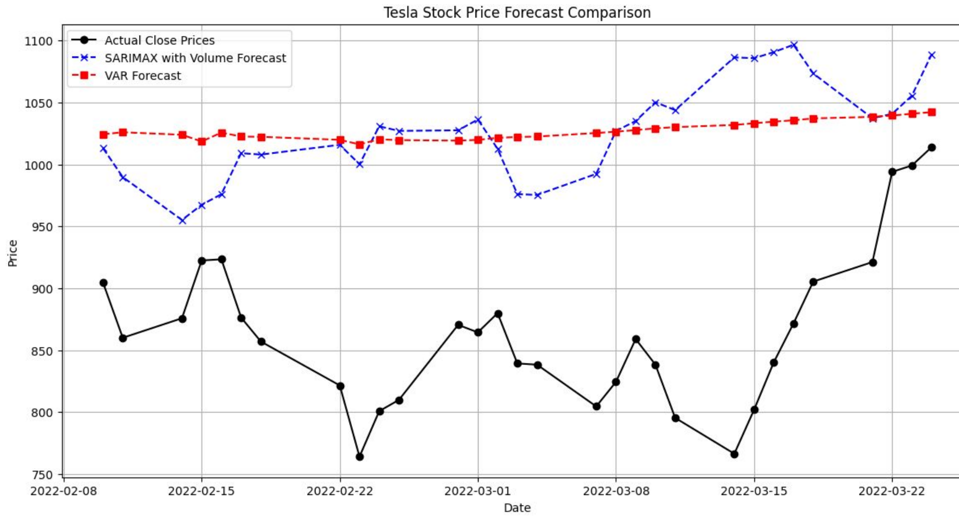
```
SARIMAX RMSE: 185.6826690881751
VAR RMSE: 172.71313731220633
Value of Information (VoI): 12.969531775968761
```



```python
import pandas as pd
import numpy as np
from statsmodels.tsa.statespace.sarimax import SARIMAX
from statsmodels.tsa.api import VAR
from sklearn.metrics import mean_squared_error
import matplotlib.pyplot as plt

data = pd.read_csv('TSLA.csv')
data['Date'] = pd.to_datetime(data['Date'])
data.set_index('Date', inplace=True)

data = data[['Close', 'Volume']].dropna()

baseline_model = SARIMAX(data['Close'], exog=data['Volume'], order=(5, 1, 0), seasonal_order=(1, 1, 1, 12))
baseline_fit = baseline_model.fit()

baseline_forecast = baseline_fit.forecast(steps=30, exog=data['Volume'].iloc[-30:])

baseline_rmse = np.sqrt(mean_squared_error(data['Close'].iloc[-30:], baseline_forecast))

var_data = data[['Close', 'Volume']].dropna()

var_model = VAR(var_data)
var_fit = var_model.fit(maxlags=15, ic='aic')

var_forecast = var_fit.forecast(var_data.values[-var_fit.k_ar:], steps=30)

var_forecast_close = var_forecast[:, 0]

actual_close_prices = data['Close'].iloc[-30:].values
new_rmse = np.sqrt(mean_squared_error(actual_close_prices, var_forecast_close))

voi = baseline_rmse - new_rmse
print(f'SARIMAX with Volume RMSE: {baseline_rmse}')
print(f'VAR RMSE: {new_rmse}')
print(f'Value of Information (VoI): {voi}')

plt.figure(figsize=(14, 7))
plt.plot(data.index[-30:], actual_close_prices, label='Actual Close Prices', color='black', marker='o')
plt.plot(data.index[-30:], baseline_forecast, label='SARIMAX with Volume Forecast', color='blue', linestyle='--', marker='x')
plt.plot(data.index[-30:], var_forecast_close, label='VAR Forecast', color='red', linestyle='--', marker='s')
plt.title('Tesla Stock Price Forecast Comparison')
plt.xlabel('Date')
plt.ylabel('Price')
plt.legend()
plt.grid()
plt.show()
```

SARIMAX WITH VOLUME VS VAR

```
SARIMAX with Volume RMSE: 177.57489256711304
VAR RMSE: 172.71313731220633
Value of Information (VoI): 4.861755254906711
```

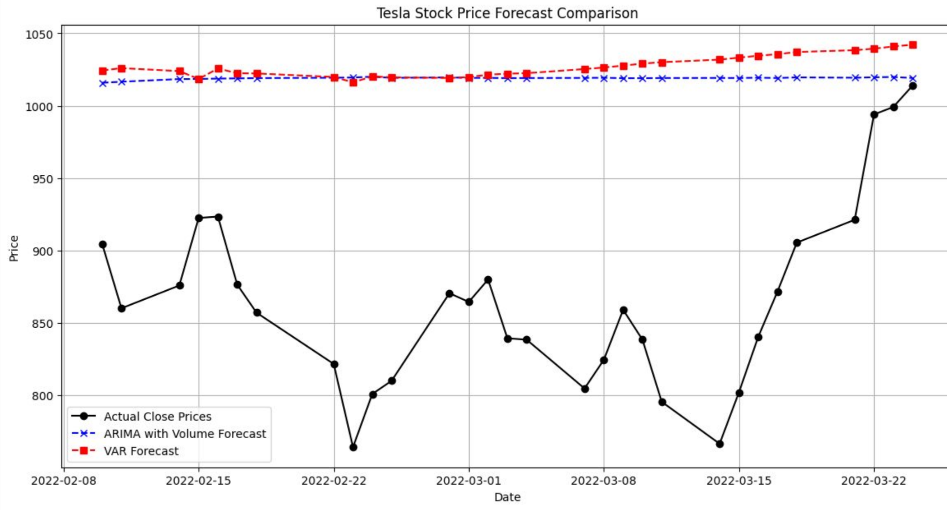


```python
import pandas as pd
import numpy as np
from statsmodels.tsa.arima.model import ARIMA
from statsmodels.tsa.api import VAR
from sklearn.metrics import mean_squared_error
import matplotlib.pyplot as plt

data = pd.read_csv('TSLA.csv')
data['Date'] = pd.to_datetime(data['Date'])
data.set_index('Date', inplace=True)

data = data[['Close', 'Volume']].dropna()

baseline_model = ARIMA(data['Close'], exog=data['Volume'], order=(5, 1, 0))
baseline_fit = baseline_model.fit()

baseline_forecast = baseline_fit.forecast(steps=30, exog=data['Volume'].iloc[-30:])

baseline_rmse = np.sqrt(mean_squared_error(data['Close'].iloc[-30:], baseline_forecast))

var_data = data[['Close', 'Volume']].dropna()

var_model = VAR(var_data)
var_fit = var_model.fit(maxlags=15, ic='aic')

var_forecast = var_fit.forecast(var_data.values[-var_fit.k_ar:], steps=30)

var_forecast_close = var_forecast[:, 0]

actual_close_prices = data['Close'].iloc[-30:].values
new_rmse = np.sqrt(mean_squared_error(actual_close_prices, var_forecast_close))

voi = baseline_rmse - new_rmse
print(f'ARIMA with Volume RMSE: {baseline_rmse}')
print(f'VAR RMSE: {new_rmse}')
print(f'Value of Information (VoI): {voi}')
```

```python
plt.figure(figsize=(14, 7))
plt.plot(data.index[-30:], actual_close_prices, label='Actual Close Prices', color='black', marker='o')
plt.plot(data.index[-30:], baseline_forecast, label='ARIMA with Volume Forecast', color='blue', linestyle='--', marker='x')
plt.plot(data.index[-30:], var_forecast_close, label='VAR Forecast', color='red', linestyle='--', marker='s')
plt.title('Tesla Stock Price Forecast Comparison')
plt.xlabel('Date')
plt.ylabel('Price')
plt.legend()
plt.grid()
plt.show()
```

ARIMA WITH VOLUME VS VAR

```
ARIMA with Volume RMSE: 166.3734912331474
VAR RMSE: 172.71313731220633
Value of Information (VoI): -6.3396460790589
```



CONCLUSION

The Analysis Demonstrates that the ARIMA model has the most Value of Information when applied for the prediction of daily closing stock value (specifically for Tesla stock).

The VAR and ARIMA models probably outperform the SARIMAX models because of the absence of seasonality in the dataset.

RESEARCH PAPER SOURCE

https://arxiv.org/abs/2410.01831

REFERENCES

https://zerotomastery.io/blog/arima-sarima-sarimax-explained/

https://en.wikipedia.org/wiki/Vector_autoregression

GITHUB LINK

https://github.com/Code-Ph0enix/timeseries/tree/main