Predicting Sugar Prices Using FBProphet

- This notebook intends to demonstrate a prediction method for a commodity (No 11 Sugar Contracts) using FB Prophet Library. We will use all data existent excluding the past 7 days to try to predict the next 14 days.
- The evaluation metric will be **MSE (Mean Squared Error)** of the predicted price versus the closing price for each of the 14 days.
- A high MSE tells us that the model is not very accurate because any big difference in predicting prices can mean profit/loss and also, in this case, can mean holding a position until the end of the contract and having the physical product delivered at your door (which is **NOT** a good thing)
- **Important:** This is **NOT financial advice**, only a programming exercise.

Importing libraries

In [10]:
```
import prophet
import yfinance as yf
import pandas as pd
import datetime as dt
from datetime import datetime
from prophet import Prophet
from sklearn.metrics import mean_squared_error
import warnings
warnings.filterwarnings('ignore')
```

Getting Sugar prices from Yahoo Finance library

In [11]:
```
sugar = yf.Ticker('SB=F')
```

In [12]:
```
sugar = sugar.history(period='max')
```

In [13]:
```
data = sugar.copy().reset_index()
```

Transforming Date from datetime to only date for better visualization and modeling

In [14]:
```
data['Date'] = data['Date'].dt.date
```

The columns that we are going to use to predict are only Date and the Close price.

In [15]:
```
data = data[['Date','Close']]
```

Prophet requires us to change the names of the date and the objective variable to ds and y, respectively

In [16]:
```
data.columns = ['ds','y']
```

Instantiating the model and creating the first predictions

In [17]:
```
prophet = Prophet(daily_seasonality=True)
prophet.fit(data)
```

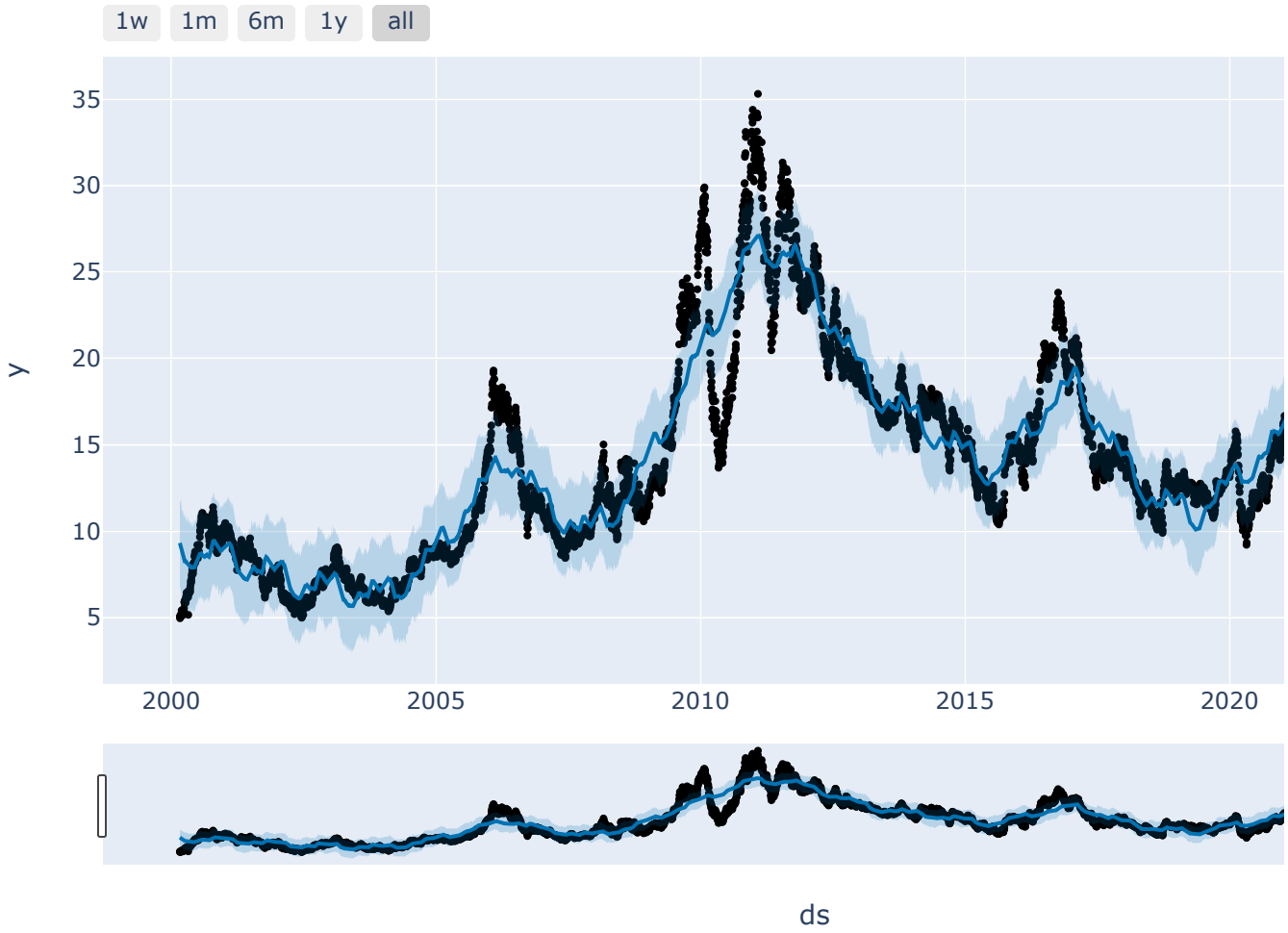
```
18:35:31 - cmdstanpy - INFO - Chain [1] start processing
18:35:32 - cmdstanpy - INFO - Chain [1] done processing
```

Out[17]:
```
<prophet.forecaster.Prophet at 0x1e5220b1ad0>
```

Predicting the Next 7 days with all the available data.

In [18]:
```
future_dates = prophet.make_future_dataframe(periods=7)
predictions = prophet.predict(future_dates)
```

In [19]:
```
from prophet.plot import plot_plotly
plot_plotly(prophet,predictions)
```



In order to test the validity of the model, we will exclude the last 7 days of the database and generate the predictions, then we will compare with the actual values to calculate the errors.

In [20]:
```
unknown_data = data.iloc[-7:]
```

In [21]:
```
data = data.iloc[:-7]
```

In [22]:
```
new_prophet = Prophet(daily_seasonality=True)
new_prophet.fit(data)
```

```
18:35:34 - cmdstanpy - INFO - Chain [1] start processing
18:35:36 - cmdstanpy - INFO - Chain [1] done processing
```

Out[22]:
```
<prophet.forecaster.Prophet at 0x1e523a9e910>
```

In [23]:
```
future_dates = (new_prophet.make_future_dataframe(periods=14))
predictions = new_prophet.predict(future_dates)
```

In [33]:
```
import matplotlib.pyplot as plt
plt.figure(figsize=(12,8))
pred = predictions[predictions['ds'].isin(unknown_data['ds'])]
pred_final = pred[['ds','yhat']]
pred_final.columns = ['Date','Predicted Price']
data_final = data.copy()
data_final.columns = ['Date','Actual Price']
sugar_2 = sugar.reset_index()[['Date','Close']]
sugar_2['Date'] = sugar_2['Date'].dt.date
sugar_2['Date'] = pd.to_datetime(sugar_2['Date'])
```

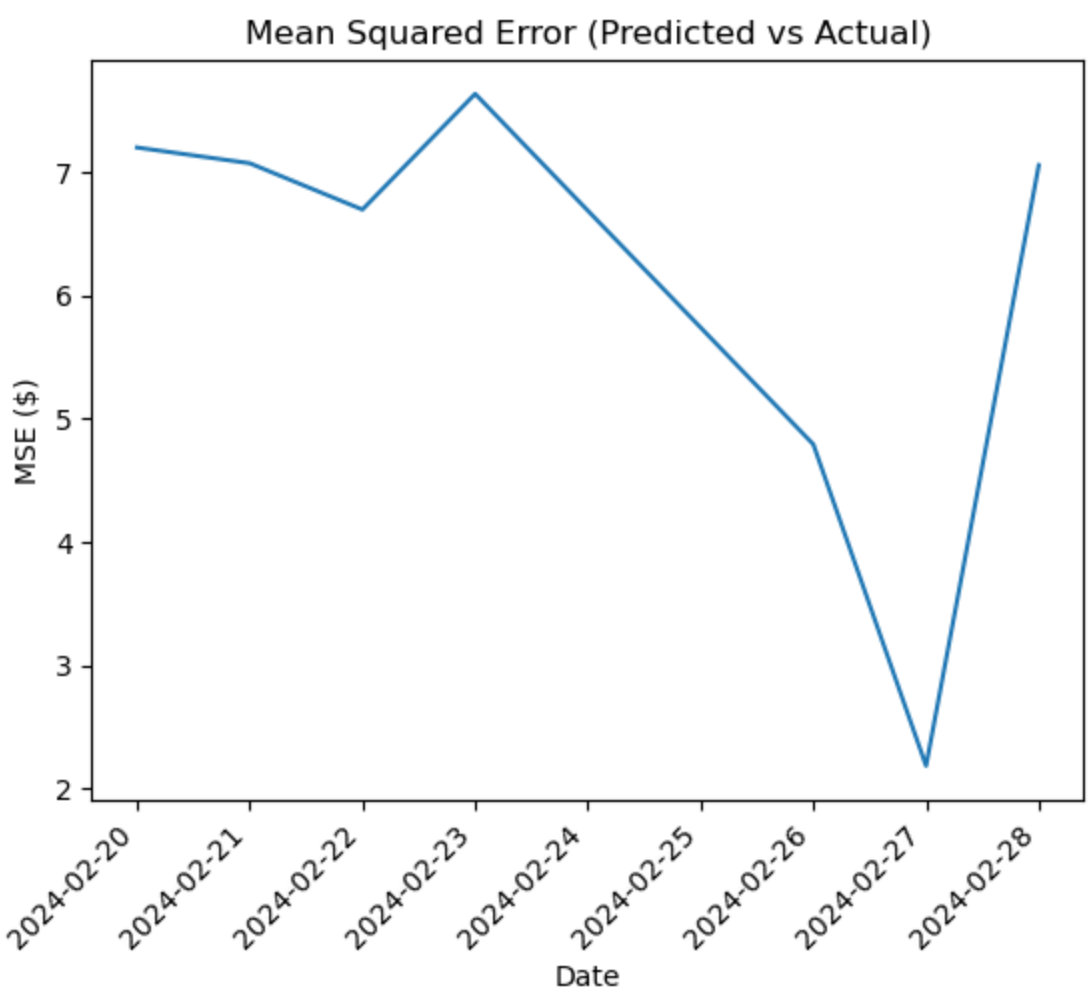
```
<Figure size 1200x800 with 0 Axes>
```

After running the models and making the predictions, let's calculate the errors and MSE

In [34]:
```
df_final = pred_final.merge(sugar_2,on='Date',how='inner')
df_final['error'] = df_final['Predicted Price'] - df_final['Close']
df_final['MSE'] = df_final.apply(lambda row: mean_squared_error([row['Close']], [row['Pr
mean_actual = df_final['Close'].mean()
df_final['MSE Percentage'] = (df_final['MSE'] / (mean_actual**2)) * 100
```

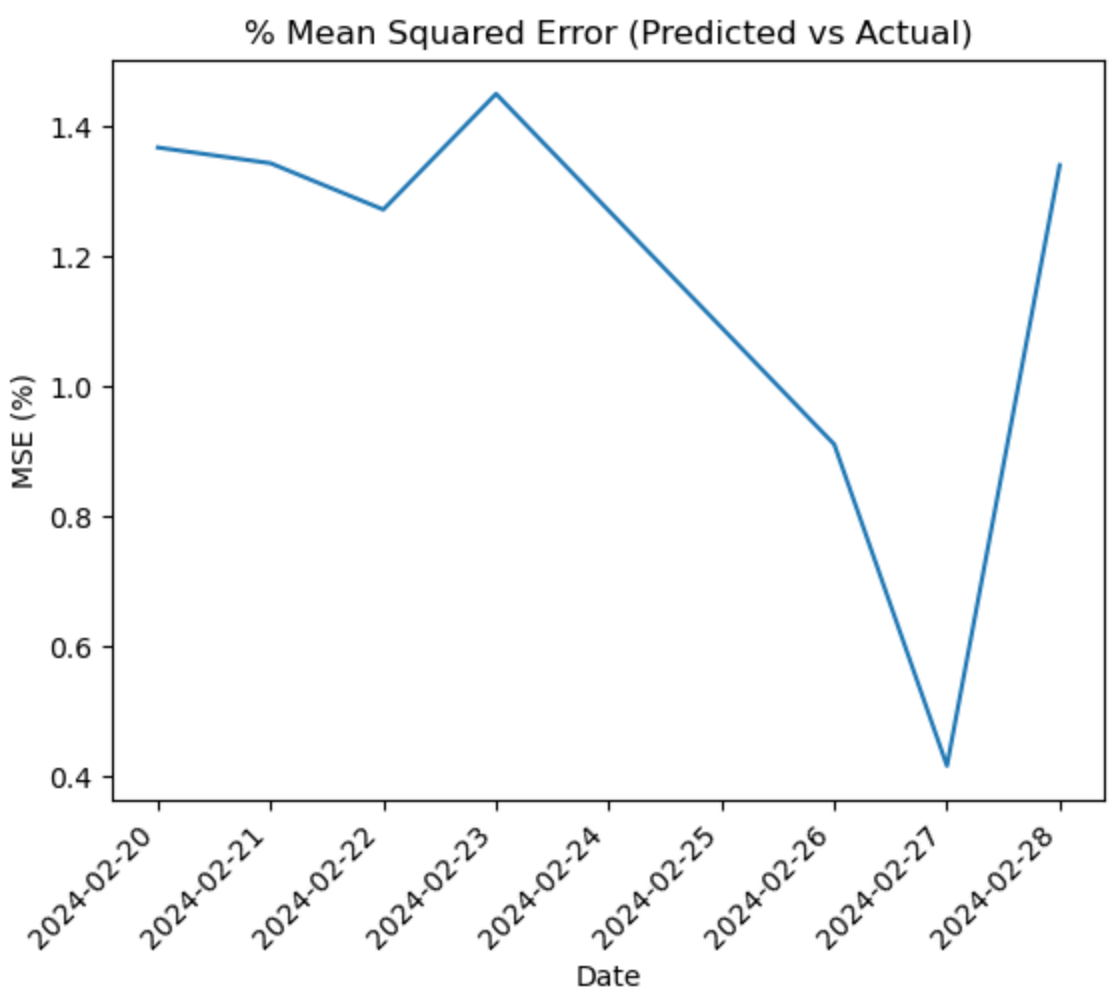
In [39]:
```
plt.plot('Date','MSE',label='MSE',data=df_final)
plt.xticks(rotation=45, ha='right')
plt.xlabel('Date')
plt.ylabel('MSE ($)')
plt.title('Mean Squared Error (Predicted vs Actual)')
```

Out[39]:
```
Text(0.5, 1.0, 'Mean Squared Error (Predicted vs Actual)')
```



In [40]:
```
plt.plot('Date','MSE Percentage',label='MSE Percentage',data=df_final)
plt.xticks(rotation=45, ha='right')
plt.xlabel('Date')
plt.ylabel('MSE (%)')
plt.title('% Mean Squared Error (Predicted vs Actual)')
```

Out[40]:
```
Text(0.5, 1.0, '% Mean Squared Error (Predicted vs Actual)')
```



Conclusion

- Although with room for improvement, this model is predicting the price for Sugar contracts with a good level of accuracy. As we can see, the MSE (%) for 14 days varies between 0.4% and 1.5%.
- The MSE, in this case, is highly correlated with the number of days that we are trying to predict since the price varies with high frequency. After testing 7, 14, 28, 56, and 365 days, the optimal scenario was 14 days which presented a good level of accuracy and a low error.
- Other factors can be taken into account to produce better predictions such as sugar cane production, and sugar mix average (published biweekly by the regulators) that will give a better understanding in terms of the future availability of this product.
- Lastly, this was only a quick introduction to predictive modeling and results. In less than 15 minutes we can produce mensurable predictions and by tunning the correct variables we can improve the results