Project Report

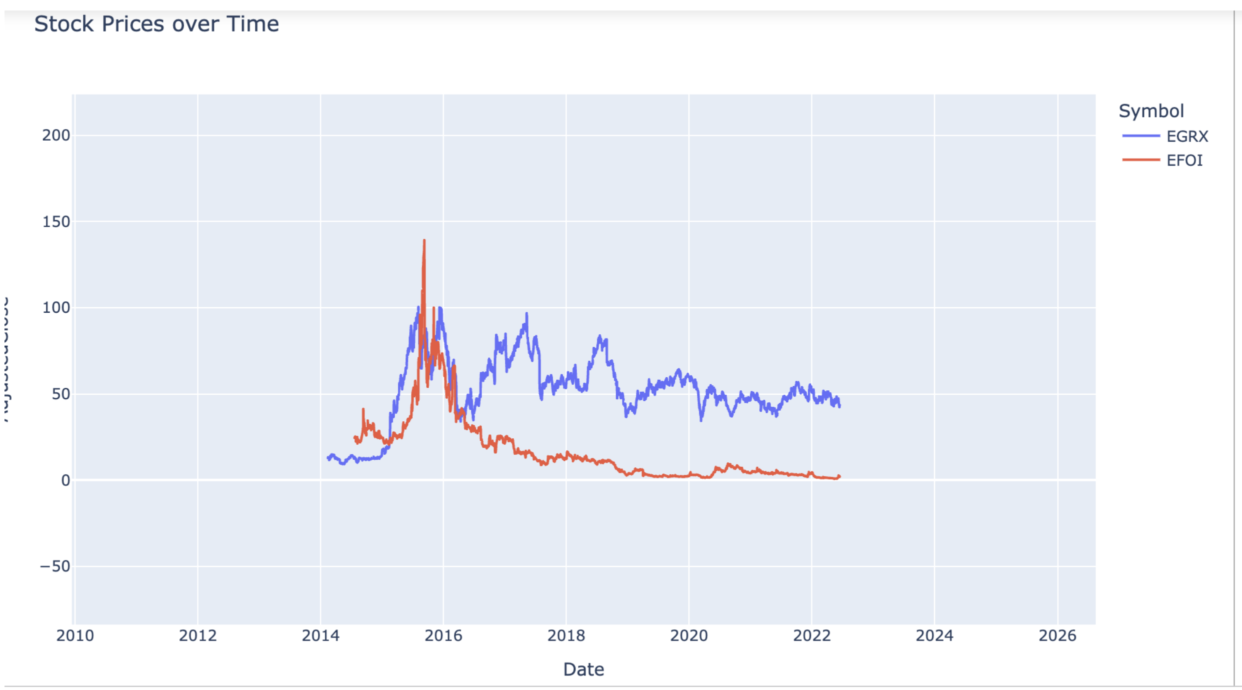
After the stock data is extracted and downloaded from Alpha Vantage API considering EFOI and EGRX companies, I merge the dataframes and following is the dataframe structure. For gathering the data, I have written a separate script alphavantage_service.py

10]:

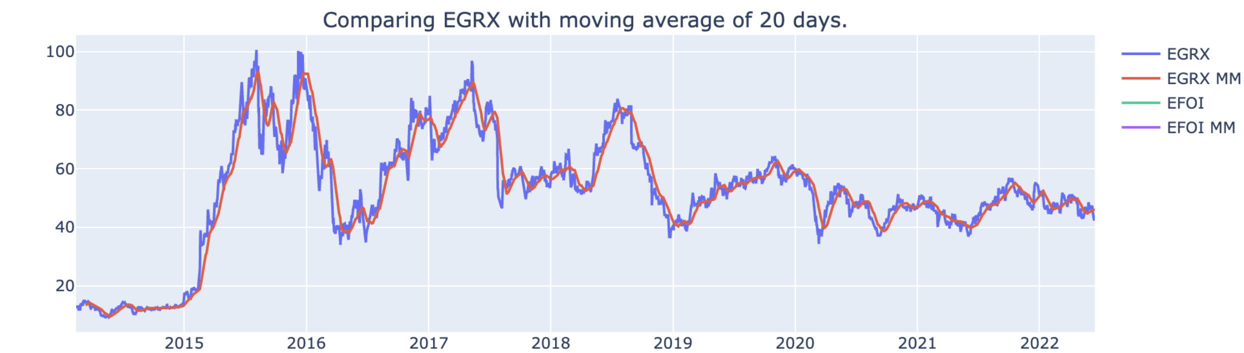
	Date	Symbol	Close	Open	Low	High	Volume	AdjustedClose	Change
0	2014-02-12	EGRX	12.83	15.5	12.75	16.44	5.94826e+06	12.83	-0.708409
1	2014-02-13	EGRX	13.22	12.51	12.47	13.48	487358	13.22	0.0303975
2	2014-02-14	EGRX	12.8	13.2	12.76	13.59	107162	12.8	-0.03177
3	2014-02-18	EGRX	12.94	13	12.6	13.07	81656	12.94	0.0109375
4	2014-02-19	EGRX	12.14	12.71	11.74	12.95	273287	12.14	-0.0618238
...	...	...	...	...	...	...	...	...	...
4086	2022-06-08	EFOI	2.7	2.98	2.3101	3.12	4.6515e+07	2.7	1.14286
4087	2022-06-09	EFOI	2.29	2.46	2.21	2.56	5.50007e+06	2.29	1.14019
4088	2022-06-10	EFOI	2.52	2.39	2.22	2.729	4.57433e+06	2.52	1.8404
4089	2022-06-13	EFOI	2.45	2.45	2.3062	2.54	4.83945e+06	2.45	1.75963
4090	2022-06-14	EFOI	2.04	2.51	1.96	2.6	2.32734e+06	2.04	-0.167347

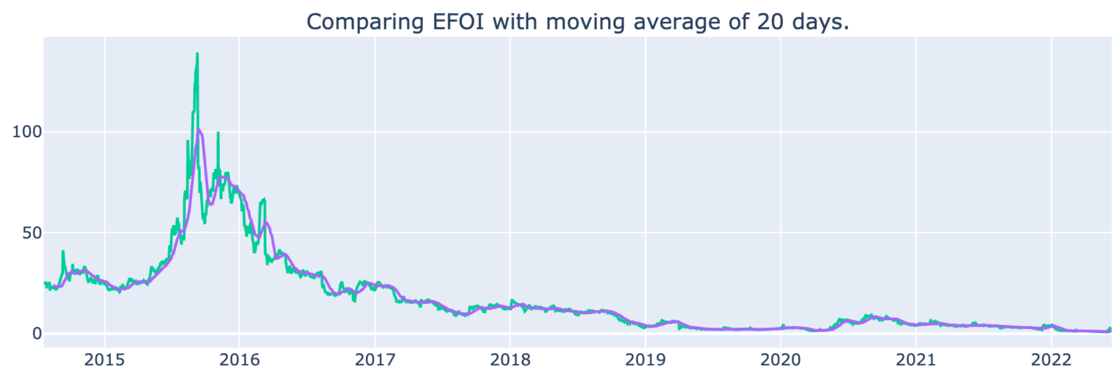
4091 rows × 9 columns

Stock prices over time



Simple Moving Average variations



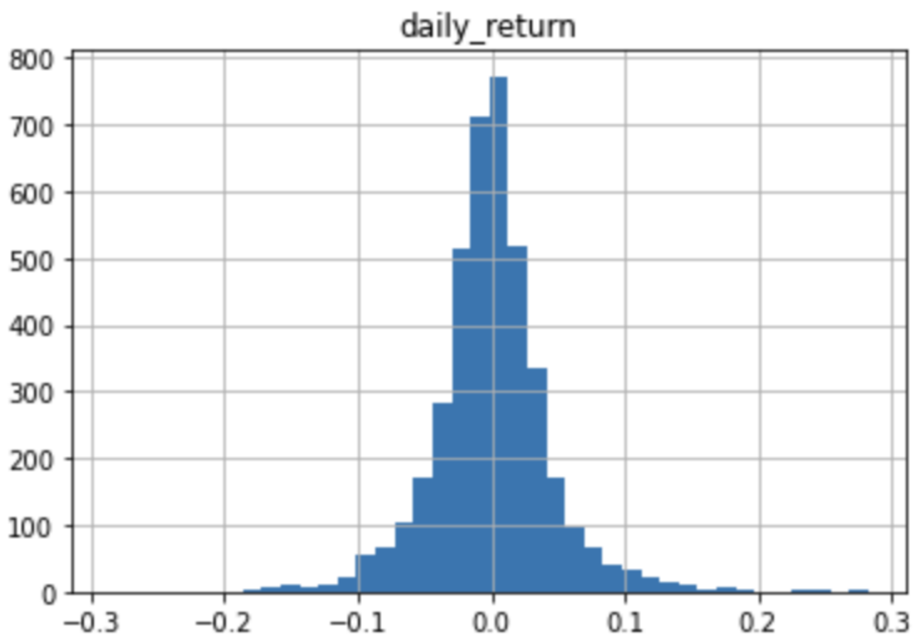


```
          Change
count  2101.000000
mean      0.034039
std       0.367559
min      -0.810404
25%      -0.021548
50%       0.001021
75%       0.023666
max       4.246269 EGRX
```

```
          Change
count  1992.000000
mean      0.590368
std       3.620128
min      -0.973425
25%      -0.036596
50%      -0.001446
75%       0.036154
max      91.427230 EFOI
```

The standard deviation for EFOI is higher (3.52) meaning the variance is higher!

The daily return is normally distributed

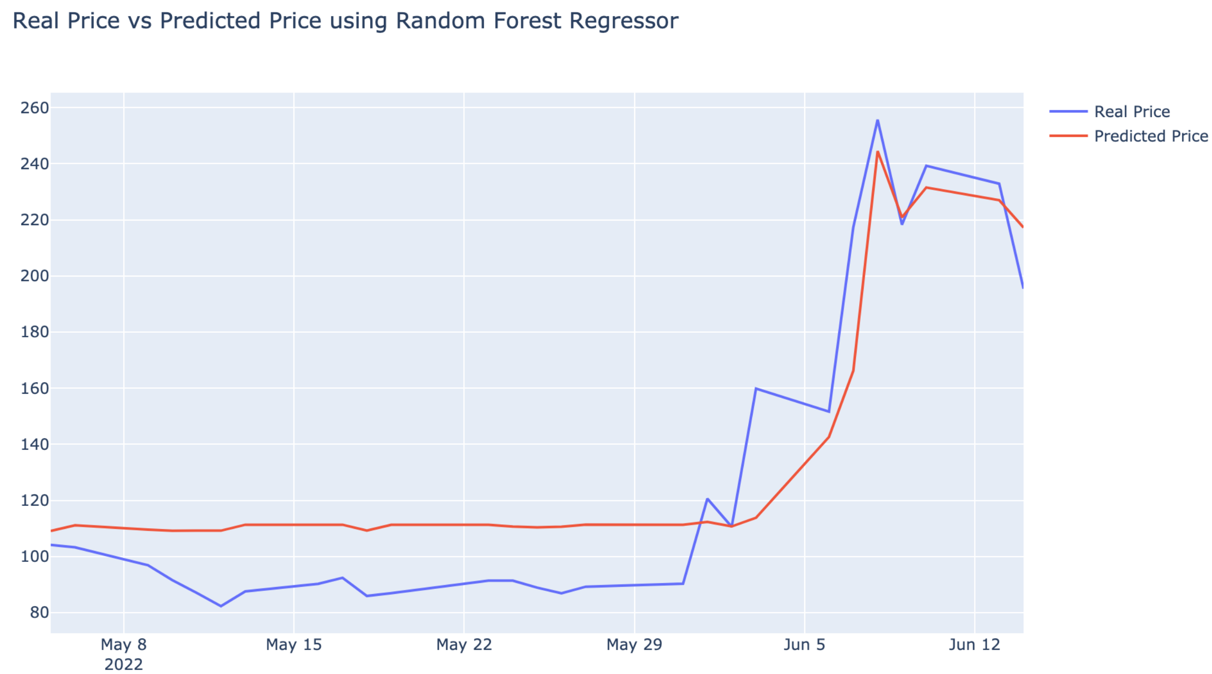


From the above histogram we see that daily return is normally distributed

After creating the daily_returns feature,based on the adjusted closed price, it was possible to plot the relation between expected Return and Risk. When the Risk and Volatile graph was visualized, it showed that EGRX Eagle Pharma stocks had lesser risk and more returns

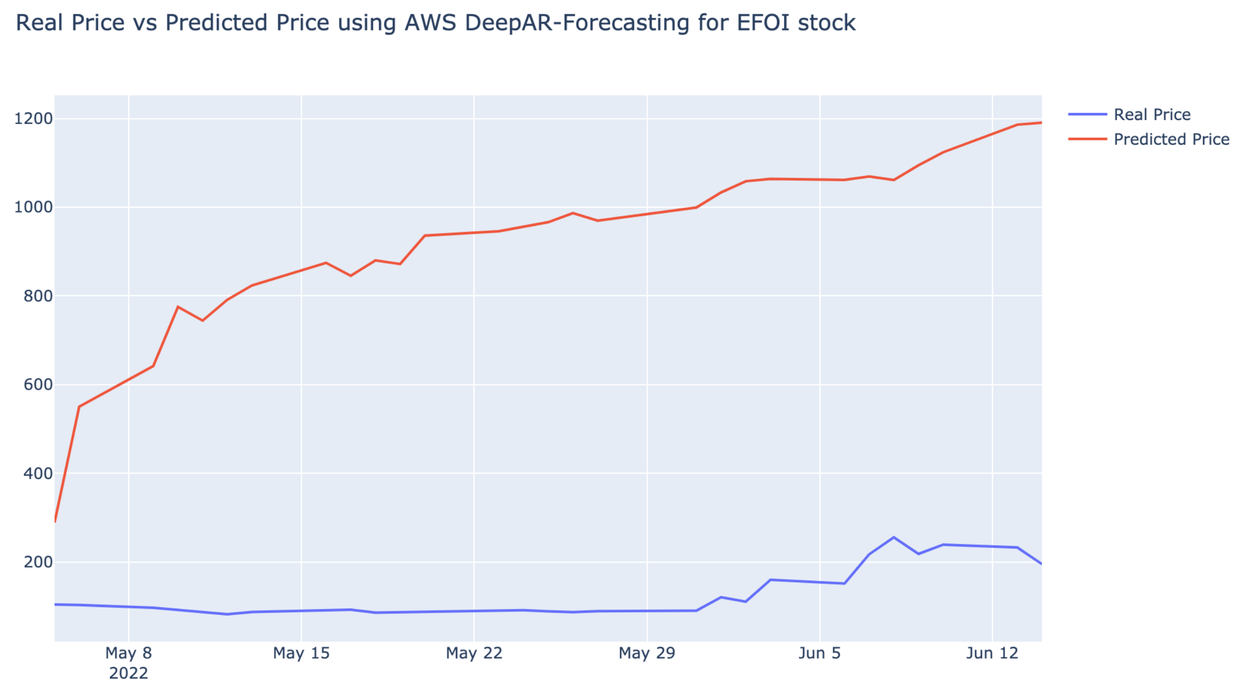
Choice of Algorithm and Techniques
For the feature engineering, I created few more features such as diff, lag and rolling average

Random Forest Regressor: A simple algorithm that used the above mentioned 3 features to create a baseline model. n_estimators= 1000 means 100 decision trees

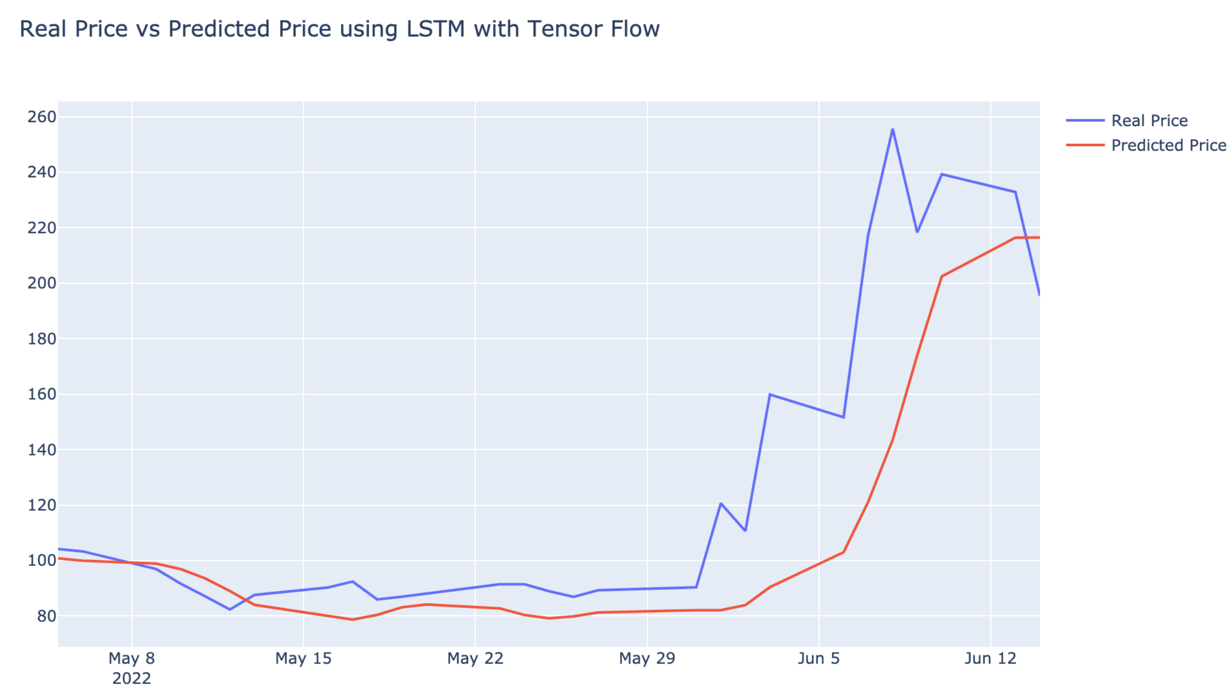


AWS DeepAR : A forecasting algorithm from AWS.We don't have to do feature engineering process as it uses scalar time series data. The DeepAR creates feature time series based on frequency of the target time series, example day of the month, day of the year. One can see the prediction deviation from the real price.

Since the DeepAR demands input in JSON format, couple of methods were created to save the train and test data in json format and uploaded to S3



LSTM: It is indeed a worth to try on time series data as we know from many examples that LSTM performance is promising on time series data. On Stocks data, it performed quite well



For Evaluations, I have used MAE and RMSE

	RMSE	MAE
Random Forest	21.650	18.500
DeepAR	812.370	794.722
LSTM	35.988	22.542

Conclusion

The Random Forest performed very well without any hyperparameters, The next closer performance was by LSTM. The DeepAR didn’t perform that well. The further steps would be to tune LSTM and also experiment with Convolutional Neural Networks.
Furthermore, one can build REST APIs like an web interface application to get predictions and moving averages visualisations