Stock Insights

AIM:

The project aims to forecast the future values of the stock based on the past stock trends in order to help investors and businesses adopt an intelligent strategy.

METHODOLOGY:

The historical data of top performing PSX companies is used to train the neural network and the model is deployed to forecast future trends.
The project aims at forecasting the future Volume(strength indicator of market) and Open (the opening price of market) from the past trends.
The brief flow of the work is described as under;

- **GETTING DATA-SET:**

  Historical data of companies is scraped from the PSX (Pakistan Stock Exchange) Data Portal.
  The data is of the form:

	Date	Open	High	Low	Close	Volume
0	2018-01-01	67.0	67.69	67.0	67.20	1051500
1	2018-01-02	67.0	70.56	67.0	70.56	51500

- **FEATURE SELECTION:**

  Understanding the relationship among different variables of the data is the most crucial step. After a lot of work, following features are selected to train the model with;

1. For Volume-forecast : (all available variables) [**"Open" , "High" , "Low" , "Close" , "Volume"**]
2. For Open price forecast : **["Open" , "High" , "Low" , "Close"].**

- **TRAIN-TEST SPLIT:**

  The train-test split is of the ratio;

  Train-set : Test-set = 75 : 25

- **FEATURE SCALING:**

  In order to standardize the range of functionality of input data-set, Standard Scalar is used from the scikit learn tools.
  Two scalars were used, one for input data-set and one for output data-set.

- **RESHAPING DATA FOR LSTM MODEL:**

  While Implementing any LSTM, data must be in 3-D shape. To do so ,the input data is transformed into the shape of (total rows, time-step , no. of features)

- **MODEL TRAINING:**

Sequential model is used with inner LSTM layers.
Model is compiled with following hyper-parameters;

1. **For Volume forecast;**

<u>**Hyper-parameters:**</u>

Optimizer : Adam
Learning rate : 0.01
Epochs : 25
Batch-size : 256

<u>**Model Summary:**</u>

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 lstm (LSTM)                 (None, 30, 128)           68608

 lstm_1 (LSTM)               (None, 30, 64)            49408

 dropout (Dropout)           (None, 30, 64)            0

 lstm_2 (LSTM)               (None, 30, 32)            12416

 dropout_1 (Dropout)         (None, 30, 32)            0

 lstm_3 (LSTM)               (None, 10)                1720

 dropout_2 (Dropout)         (None, 10)                0

 dense (Dense)               (None, 1)                 11

=================================================================
Total params: 132,163
Trainable params: 132,163
Non-trainable params: 0
_________________________________________________________________
```

2. **For Open Forecast:**

<u>**Model Summary:**</u>

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 lstm (LSTM)                 (None, 14, 128)           68096

 lstm_1 (LSTM)               (None, 14, 64)            49408

 dropout (Dropout)           (None, 14, 64)            0

 lstm_2 (LSTM)               (None, 14, 32)            12416

 dropout_1 (Dropout)         (None, 14, 32)            0

 lstm_3 (LSTM)               (None, 10)                1720

 dropout_2 (Dropout)         (None, 10)                0

 dense_out (Dense)           (None, 1)                 11

=================================================================
Total params: 131,651
Trainable params: 131,651
Non-trainable params: 0
_________________________________________________________________
```

<u>**Hyper-parameters:**</u>

Optimizer : Adam
Learning rate : 0.001
Epochs : 25
Batch-size : 128

After compiling the model, with these hyper-parameters, the train set is fitted and the model is saved.

- **FUTURE FORECAST:**
  The model is then used to predict the future trends of open/volume variables.

MINIMIZING LOSS TO GET ACCURATE RESULTS:

The crucial stage, of any ML model, is to train the model. Many difficulties were faced while training the model against the data-set.
The two main problems were:

1. Model Overfitting
2. Unpredictable nature of stock data.

To overcome the problem of overfitting, two strategies were used;

1. Model Complexity
2. Regularization
3. Dropout

Model Complexity: By making the neural network more deep and complex, the model seems to get good results (along with other two strategies).
At first the model has

- Input layer
- one hidden layer
- Output layer

The model was updated to

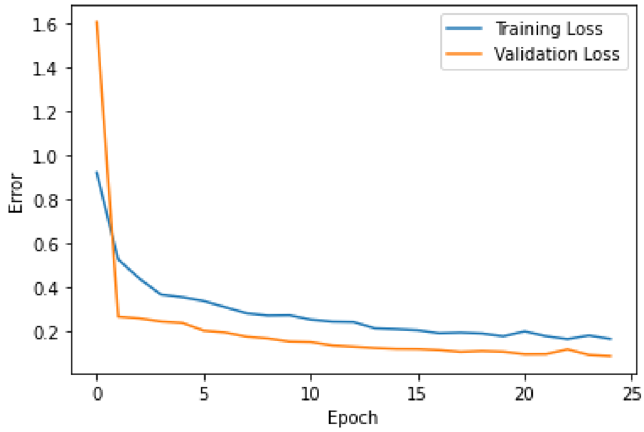
- Input layer
- three hidden layers
- Output layer

Regularization: Over-fitting in the model is greatly reduced by using "L2 regularization", as it adds a penalty to the loss function to punish higher weights of neurons. The lambda for regularization is also chosen after many trials.
For Volume-forecast model: L2-regularizer with lambda = 0.01 in the 1^{st} and 2^{nd} hidden layers.
For Open-forecast model: L2-regularizer with lambda = 0.001 in the 1^{st} and 2^{nd} hidden layers.
Dropout: Another technique deployed was adding Dropouts to the neural network. It helps in removing noise, by dropping the unnecessary neurons from the network layer. Dropouts were added with the rate of 0.2 after each hidden layer.

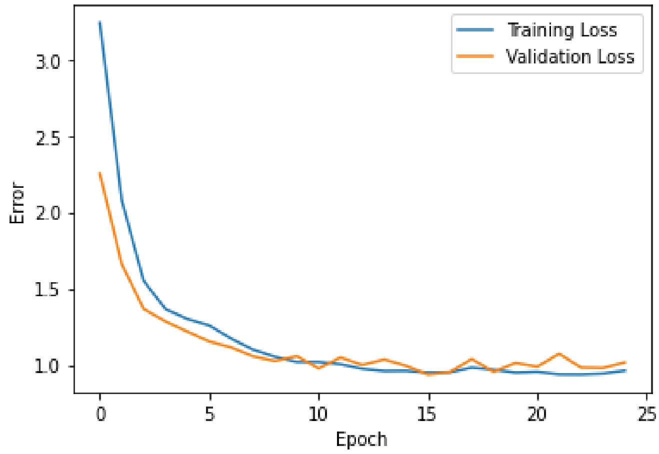
MODEL PERFORMANCE:

Loss Curves:

- **For Open-forecast:**



- **For Volume-forecast:**



Accuracy metrics:

To evaluate model performance, “Mean Squared Error” metric is used.
Test set is used to get the predictions of the model, and the output is compared against the actual values by “MeanSquaredError()” metric.

- **For Open-forecast model:**

```
]: pred = model.predict(Xtest)
```

```
]: mse = losses.MeanSquaredError()
```

```
]: mse(ytest , pred).numpy()
```

```
]: 0.6641352
```

- **For Volume-forecast model:**

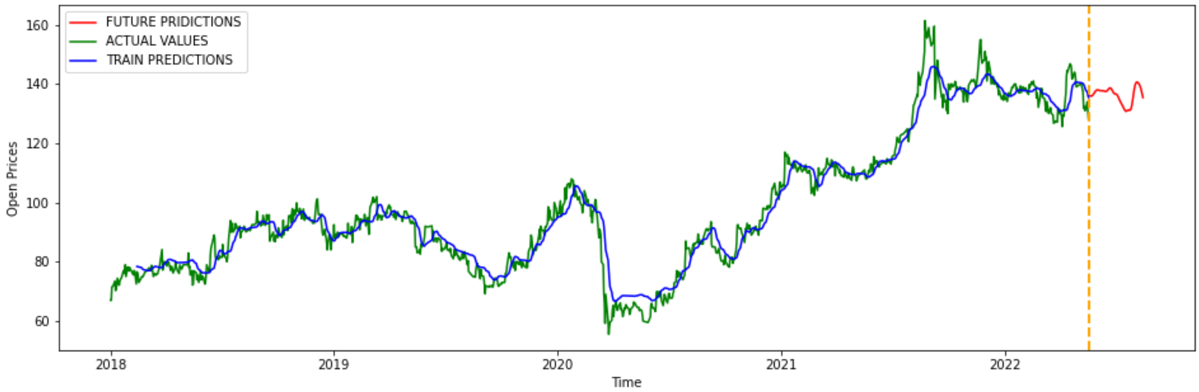
```
: mse = losses.MeanSquaredError()
```

```
: mse(ytrain , pred).numpy()
```

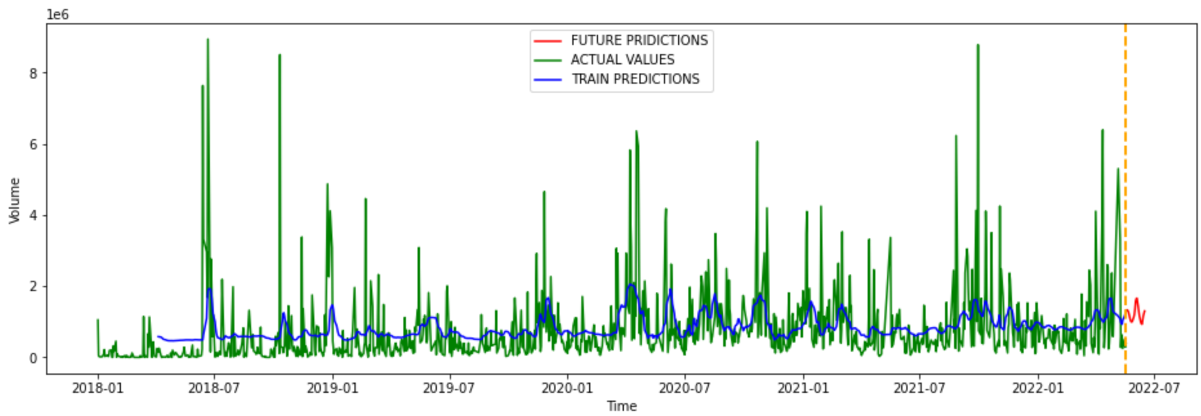
```
: 0.037807386
```

RESULTS VISUALIZATION:

For the open forecast model, the actual values, prediction over train set and future predictions by the model are as under;



Similarly for volume forecast:
(red line indicating future values of 30 days)



INFERENCE:

There are many fluctuations in stock trends and hence are very different to forecast. LSTM, as used in the project, can be adjusted with different hyper-parameters , number of layers, dropout values to get the precise results.
If the model is not good enough to predict the actual price/volume on a specific day, it is still precise enough to get an insight of the stock's direction.