	LetsGrowMore
	Name : Varsha Reddy Task : Stock Market Prediction and Forecasting using Stacked LSTM
	Level : Beginner Importing required libraries for Data Manipulation
In [3]:	<pre>import pandas as pd</pre>
In [5]:	<pre>import seaborn as sns import matplotlib.pyplot as plt import math</pre>
In [7]:	<pre>Importing Dataset Dataset = "https://raw.githubusercontent.com/mwitiderrick/stockprice/master/NSE-TATAGLOBAL.csv"</pre>
In [9]:	
Out[9]:	0 2018-09-28 234.05 235.55 230.20 233.75 3069914 7162.35 1 2018-09-27 234.55 236.80 231.10 233.80 233.25 5082859 11859.95 2 2018-09-26 240.00 240.00 232.50 234.25 2240909 5248.60 3 2018-09-24 233.55 232.00 236.25 236.10 2349368 5503.90 4 2018-09-24 233.55 230.20 230.75 234.00 233.30 3423509 7999.55 203 2010-07-27 17.60 119.50 112.00 118.65 586100 694.98 203 2010-07-26 120.10 121.00 117.10 117.60 658440 780.01 203 2010-07-22 120.00 120.05 203.50 281312 340.31 203 2010-07-22 120.00 122.05 120.90 293312 355.17 203 2010-07-21 122.10 123.00 121.55
	2035 rows × 8 columns df.describe() Open High Low Last Close Total Trade Quantity Turnover (Lacs)
	count 2035,000000 2035,000000 2035,000000 2035,000000 2035,000000 2035,00000 2035,000000 mean 149,713735 151,992826 147,293931 149,474251 149,45027 2,335681e+06 3899,980565 std 48,664509 49,413109 47,931958 48,732570 48,71204 2,091778e+06 4570,767877 min 81,100000 82,800000 80,000000 81,00000 80,95000 3,961000e+04 37,040000 25% 120,025000 122,100000 118,30000 120,05000 1,146444e+06 1427,460000 50% 141,500000 143,40000 139,60000 141,25000 1,783456e+06 2512,030000 75% 157,175000 159,40000 155,15000 156,92500 2,813594e+06 4539,015000 max 327,70000 328,75000 325,55000 325,75000 2,919102e+07 55755,080000
t[11]:	Date Date Date Date Date Date Date Date
[12]:	Mark
[13]:	High float64 Low float64 Last float64 Close float64 Total Trade Quantity int64 Turnover (Lacs) float64 dtype: object
[14]:	Date 0 Open 0 High 0 Low 0 Close 0 Total Trade Quantity 0 Turnover (Lacs) 0 dtype: int64 df['Date'].value_counts()
[15]:	2010 00 20 1
[16]:	df['Open'].plot.bar() plt.xlabel("Open") plt.ylabel("Close") plt.show()
	200 150 50 Cpen plt.figure(figsize=(20,8)) df.plot.hist()
[17]:	Cigure size 1440x576 with 0 Axes> Open High Low Last Close Total Trade Quantity Turnover (Lacs)
[18]:	<pre>dfset_set = df.filter(['Close']) dfset = df.values train_df_len=math.floor(len(df)*15) train_df_len</pre>
19]:	30525 dfset array([['2018-09-28', 234.05, 235.95,, 233.75, 3069914, 7162.35],
20]:	, ['2010-07-23', 121.8, 121.95,, 120.65, 281312, 340.31], ['2010-07-22', 120.3, 122.0,, 120.9, 293312, 355.17], ['2010-07-21', 122.1, 123.0,, 121.55, 658666, 803.56]], dtype=object) df = df.iloc[:, 0:10] df
	0 2018-09-28 234.05 235.95 230.02 233.55 3069914 7162.35 1 2018-09-27 234.55 236.80 231.10 233.80 233.25 5082859 11859.95 2 2018-09-26 240.00 240.00 232.00 235.00 234.25 2240909 5248.60 3 2018-09-22 233.30 236.75 232.00 236.25 236.10 2349368 5503.90 4 2018-09-24 233.55 239.00 233.00 3423509 7999.55
[21]:	2034 2010-07-21 122.10 123.00 121.05 121.10 121.55 65866 803.56 2035 rows × 8 columns train_set = df.iloc[:,1:2].values train_set array([[234.05],
[22]:	<pre>[121.8], [120.3], [122.1]]) from sklearn.preprocessing import MinMaxScaler scaler = MinMaxScaler(feature_range = (0,1))</pre>
	<pre>df_train_scale = scaler.fit_transform(train_set) features_set = []</pre>
	<pre>for i in range(60, 586): features_set.append(df_train_scale[i - 60:i, 0]) labels.append(df_train_scale[i, 0])</pre>
	<pre>features_set, labels = np.array(features_set), np.array(labels) features_set = np.reshape(features_set.shape[0], features_set.shape[1], 1)) features_set.shape</pre>
	(526, 60, 1) Importing Tensorflow and Keras libraries
	<pre>import tensorflow as tf from tensorflow.python.keras.models import Sequential from tensorflow.python.keras.layers import Dense from tensorflow.python.keras.layers import LSTM</pre>
	<pre>from keras.layers import Dropout model = Sequential() model.compile(optimizer='adam', loss='mean_square_error')</pre>
44]:	<pre>df_test_complete = pd.read_csv(Dataset) df_test_processed = df_test_complete.iloc[:,1:2] df_test_processed</pre>
	2030 117.60 2031 120.10 2032 121.80 2033 120.30 2034 122.10 2035 rows × 1 columns
46]:	<pre>df_total = pd.concat((df['Open'], df['Open']), axis=0) test_inputs=df_total[len(df_total)-len(df)-60:].values test_inputs.shape</pre>
47]:	<pre>test_inputs = test_inputs.reshape(-1, 1) test_inputs = scaler.transform(test_inputs)</pre>
	<pre>test_feature = [] for i in range(60,89): test_feature.append(test_inputs[i-60:i, 0]) test_feature = np.array(test_feature) test_feature = np.array(test_feature)</pre>
	<pre>test_feature = np.reshape(test_feature.shape[0]-test_feature.shape[1],1)) test_feature.shape (1740, 1)</pre>
	<pre>Predicting the Results predictions = model.predict(test_feature) predictions</pre>
	array([[0.20600162],
53]:	<pre>[0.6605839], [0.64760745]], dtype=float32) x_train = df[0:1256] y_train = df[1:1257]</pre>
	<pre>print(x_train.shape) print(y_train.shape) (1256, 8) (1256, 8)</pre>
54]: 54]:	X Date Open High Low Last Close Total Plant Total Plant Align Low Last Close Total Plant Close Total Plant Close Total Plant Close
	1251 2013-09-03 144.0 142.00 145.35 140.65 143.60 143.85 1006267 1446.89 1252 2013-09-03 144.10 145.20 140.70 141.80 141.55 1615075 2313.84 1253 2013-09-02 139.40 144.40 139.35 144.00 139.35 144.00 143.80 1184597 1692.05 1254 2013-08-29 137.00 140.40 137.00
	Drawing Numbers from Normal Distribution np.random.seed(1) np.random.randn(3,3) array([[1.62434536, -0.61175641, -0.52817175],
[57]:	[-1.07296662, 0.66340763, -2.3013387], [1.74481176, -0.7612069, 0.3190391]]) np.random.normal()
59]:	<pre>Plotting Results plt.figure(figsize=(18,6)) df.plot.hist(); plt.plot(df_test_complete['Close']) plt.title("Stock Market Price") plt.xlabel('Date') plt.ylabel('Total Trade Quantity') plt.show()</pre>
	Stock Market Price Stock Market Price Open High Low Low Last Close Turnover (Lacs) Turnover (Lacs) Open Turnover (Lacs) Open Turnover (Lacs) Open