Machine Learning In Trading

B.tech Project
Department of Computer Science and Engineering

Faculty Advisor: Subhajit Sidhanta

Rashmi Sahu
B15CS032

November 25, 2018

Contents

Chapter 1

Introduction

The long term objective of this project is to use machine learning for solving problems in various fields like medicine, chemical engineering, finance, agriculture etc. As this application takes a source code file as an input so it can be used for any type of application. Here I have taken a finance problem to find future price of stock using most optimal Machine learning model.

As most of the people nowadays are interested to invest in stock markets so they need to make suitable trading decisions (whether to buy or sell the stocks) that can be done if one knows whether stock price will raise or goes down in future. Future stock price can be found by analysing and doing some mathematical calculations on historical stock data but takes a lot of time and human effort so to do this efficiently as well as effectively I have tried various machine learning algorithms and chose one with the maximum accuracy to train the model to predict future stock price.

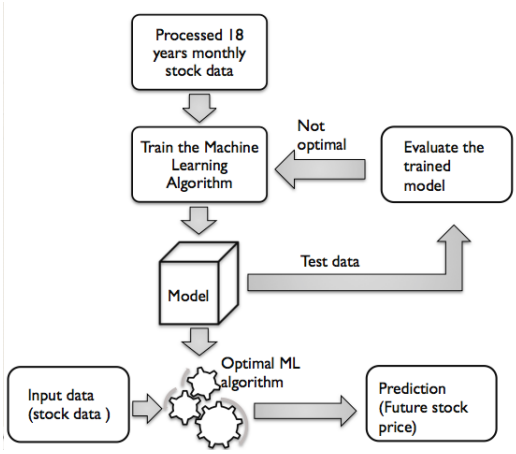


Figure 1.1 – Block Diagram

Chapter 2

Literature Survey

As there are thousands of stocks available for investors to invest in, they need to know the future price of stock for making right trading decisions. It is presumed that future stock price is related to stock data available online, thus technical analysis can be performed to extract processed information from this data. The subset of Machine learning algorithms that are to be tried depends on kind of data used. Here stock data contains numerical values so regression algorithms should be implemented as classification algorithms are used for catogarical problems. Regression models predicts numerical values whereas classification models predicts binary or multi-class outputs.
Scikit-learn python library is used as it is most popular tool being used nowadays for implementing various machine learning algorithms both supervised as well as unsupervised learning algorithms. Scikit-learns is based on numpy and Scipy that are basic Python libraries and can implement Machine Learning algorithms like K-nearest neighbour, Support vector machine, logistic regression, linear regression, Lasso regression just in few number of lines.
As this project includes 2-dimensional data stored in csv file so pandas python library is used that is most popular for data extraction and preparation as it provides high-level data structures like Data frames(two-dimensional). Pandas library is very simple to use as it has many in-built methods for filtering, combing and performing analysis on data.

Chapter 3

Methods Involved

The objective of this project is to build a web application that takes an application file that solves some definite problem belonging to various fields like finance, medicine, chemical etc as an input and to build and train the most optimal Machine Learning to learn the behaviour of that application. Later on that trained ML model can be used to predict the target value. Here I have taken a finance problem to find the future price of the stock that is required to make important trading decisions.**Django** framework has been used to build a web application.

3.1 Data Description

The dataset used is last 18 years yahoo finance data that contains basic information about stocks like open price, close price, high price, low price, adjusted price, volume of stocks for certain range fo time.

3.2 Data Preprocessing

It is presumption that future stock price is related to basic information of stocks available online. So, technical analysis need to be performed using mathematics and statistics to extract processed information from raw data. Following features are generated from raw data: 1.Daily return
2.Moving Average
3.Percentage change in price
4.Percentage change in high and low price
5.Average of close and high price
6.Average of close and open price 6.Execution time to calculate these features for each set of data
7.Balance volume of stock
8.Future Price

	Open	High	Low	Close	Adj. Close	Volume
0	51.091900	51.091900	38.487400	46.038898	20.643429	42251934
1	51.091900	51.091900	38.487400	46.038898	20.643429	42251934
2	46.319599	55.780102	45.196701	48.284698	21.650429	105045235
3	48.032101	70.181297	47.947800	65.970398	29.580530	150395215
4	66.279198	76.862503	59.682098	64.847504	29.077034	130242309
5	63.724602	72.904297	61.366501	68.805702	30.851858	120818230
6	67.654701	69.058296	52.271000	58.222401	26.106396	71507492
7	57.969700	58.727699	41.912300	49.660301	22.267231	31925754
8	49.969101	72.427002	49.211102	61.563000	27.604294	175134215
9	60.917301	86.182602	59.148800	78.153801	37.543209	185833404
10	78.546898	97.916901	77.143204	77.957298	37.448822	164460186
11	78.013496	98.141502	73.072701	93.228798	44.784882	131931046
12	94.828903	95.727203	75.655403	82.168198	39.471638	88437024
13	82.364700	96.569397	77.788902	83.347298	40.038048	144579799
14	83.235001	87.951103	73.156898	79.389000	38.136574	74927192
15	80.511902	86.463303	77.480103	79.922401	38.392811	52997767
16	83.094597	103.811996	77.817001	78.238098	37.583710	99981855
17	78.097702	84.357903	67.542397	67.682800	32.513203	50388683
18	69.058296	71.219902	50.025200	64.651001	31.056793	67391494
19	65.549301	70.630402	58.446899	60.692699	29.155315	74593738
20	60.692699	70.125099	55.022099	68.609200	35.071209	63398917
21	69.619797	79.445198	66.195000	66.503799	33.994980	74148220
22	67.317902	81.803299	61.759499	66.223000	33.851437	103475579
23	66.475700	67.879303	58.390800	61.198101	31.282848	47999761
24	61.619099	64.819397	52.635899	52.944698	27.063929	27268333
25	52.523701	58.896099	47.751301	54.236099	27.724056	39470690
26	55.162498	70.125099	54.320301	69.114502	35.329510	100207918
27	68.637299	74.616699	65.970398	73.044601	37.338463	96371076
28	73.381500	78.939903	69.282898	77.115196	39.419247	115666566
29	76.413399	92.386597	74.672897	86.744003	44.341248	187824236
..	...	...	...	...	...	...
212	332.505005	338.841003	296.539001	318.546997	300.112701	145066940
213	320.023987	331.838013	282.295990	306.734009	288.983307	139079794
214	311.069000	361.467987	297.348999	338.269012	318.693390	167586387
215	339.316986	375.997986	337.506989	352.608002	340.067902	143176393
216	353.751007	389.287994	338.506989	356.562012	343.881317	135214551
217	359.085999	411.487000	358.515015	386.286987	372.549133	141027377
218	385.714996	420.632996	348.987000	395.433014	381.369934	143175098

Figure 3.1 – Raw data

	Open	High	Low	Close	...	Avg	CH_Avg	execution_time	Moving Avg
0	51.091900	51.091900	38.487400	46.038898	...	48.565399	1.032487	84.074974	44.789650
1	51.091900	51.091900	38.487400	46.038898	...	48.565399	0.523813	63.396215	44.789650
2	46.319599	55.780102	45.196701	48.284698	...	47.302149	0.036471	56.870937	50.488402
3	48.032101	70.181297	47.947800	65.970398	...	57.001250	-1.171998	56.727648	59.064549
4	66.279198	76.862503	59.682098	64.847504	...	65.563351	-1.440982	56.669950	68.272301
5	63.724602	72.904297	61.366501	68.805702	...	66.265152	-0.720360	56.579828	67.135399
6	67.654701	69.058296	52.271000	58.222401	...	62.938551	1.318331	57.897806	60.664648
7	57.969700	58.727699	41.912300	49.660301	...	53.815001	-0.093221	61.055183	50.320000
8	49.969101	72.427002	49.211102	61.563000	...	55.766051	-1.091810	65.764666	60.819052
9	60.917301	86.182602	59.148800	78.153801	...	69.535551	-0.750127	59.823275	72.665701
10	78.546898	97.916901	77.143204	77.957298	...	78.252098	2.140104	59.483051	87.530052
11	78.013496	98.141502	73.072701	93.228798	...	85.621147	-0.846241	56.727171	85.607101
12	94.828903	95.727203	75.655403	82.168198	...	88.498550	0.710429	56.362152	85.691303
13	82.364700	96.569397	77.788902	83.347298	...	82.855999	0.995080	58.825016	87.179149
14	83.235001	87.951103	73.156898	79.389000	...	81.312000	-0.376959	58.572054	80.554001
15	80.511902	86.463303	77.480103	79.922401	...	80.217151	2.063685	61.259985	81.971703
16	83.094597	103.811996	77.817001	78.238098	...	80.666348	-1.552540	56.093931	90.814498
17	78.097702	84.357903	67.542397	67.682800	...	72.890251	-0.366354	56.182861	75.950150
18	69.058296	71.219902	50.025200	64.651001	...	66.854648	-0.090972	56.217909	60.622551
19	65.549301	70.630402	58.446899	60.692699	...	63.121000	-1.252660	56.101084	64.538651
20	60.692699	70.125099	55.022099	68.609200	...	64.650949	-1.025202	56.214094	62.573599
21	69.619797	79.445198	66.195000	66.503799	...	68.061798	0.199132	56.365013	72.820099
22	67.317902	81.803299	61.759499	66.223000	...	66.770451	-1.468913	56.703806	71.781399
23	66.475700	67.879303	58.390800	61.198101	...	63.836900	-0.069484	56.272984	63.135052
24	61.619099	64.819397	52.635899	52.944698	...	57.281898	-1.646923	56.052923	58.727648
25	52.523701	58.896099	47.751301	54.236099	...	53.379900	1.150452	56.619644	53.323700
26	55.162498	70.125099	54.320301	69.114502	...	62.138500	-0.834718	56.355000	62.222700
27	68.637299	74.616699	65.970398	73.044601	...	70.840950	0.445626	56.107998	70.293548
28	73.381500	78.939903	69.282898	77.115196	...	75.248348	-0.424805	63.647032	74.111401
29	76.413399	92.386597	74.672897	86.744003	...	81.578701	0.489791	59.668779	83.529747
..	...	...	...	...	...	...	...	...	...
212	332.505005	338.841003	296.539001	318.546997	...	325.526001	0.789800	56.200981	317.690002
213	320.023987	331.838013	282.295990	306.734009	...	313.378998	-0.450276	56.296825	307.067002
214	311.069000	361.467987	297.348999	338.269012	...	324.669006	0.358140	56.156158	329.408493
215	339.316986	375.997986	337.506989	352.608002	...	345.962494	-0.483189	60.006857	356.752488
216	353.751007	389.287994	338.506989	356.562012	...	355.156509	-0.194915	62.642336	363.897492
217	359.085999	411.487000	358.515015	386.286987	...	372.686493	-1.173927	58.423758	385.001008
218	385.714996	420.632996	348.987000	395.433014	...	390.574005	0.409652	58.510065	384.809998
219	397.625000	414.773987	359.515015	372.757996	...	385.191498	1.290626	58.555126	387.144501
220	374.854004	454.217010	370.519012	441.165009	...	408.009507	1.128862	58.464050	412.368011
221	442.117004	472.462006	434.114014	459.885986	...	451.001495	1.860949	58.636904	453.288010
222	462.076996	484.704987	440.355011	459.885986	...	460.981491	1.206175	56.998014	462.529999
223	463.029999	482.417999	421.157013	428.016998	...	445.523499	0.237690	56.139946	451.787506
224	428.016998	496.041992	407.770996	480.846008	...	454.431503	0.736363	58.673143	451.906494

Figure 3.2 – Processed data

3.3 Training various ML models

80% of stock data is used as training data and 20% is used as test data. Following Machine Learning algorithms have been trained on processed data and are tested on test data to chose the one with maximum accuracy:

1.Linear Regression
2.Lasso Regression
3.Polynomial Regression
4.Ridge Regression
5.K-nearest neighbour
6.Support Vector Machine
7.Logistic Regression

The ML algorithm with maximum accuracy is used to build a trained model that can be further used to predict future stock price.

Chapter 4

Results

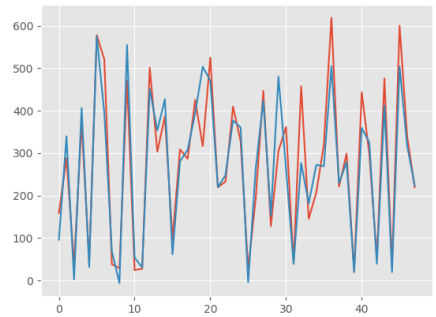
4.1 Model Selection

The accuracy of the Machine Learning model is based on the number of correct predictions made by it.

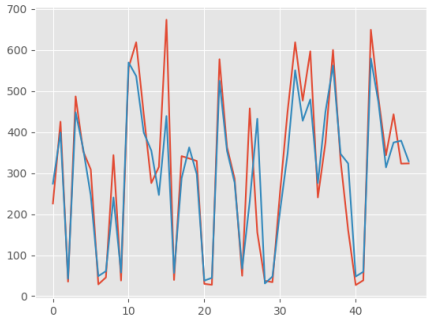
ML algorithm	Accuracy	Root mean squared error
Linear Regression	87.78%	58.92
Polynomial Regression	78.33%	78.47
Ridge Regression	87.59 %	59.36
Lasso Regression	88.25%	57.76
K-nearest neighbour	0.00%	246.97
Support Vector Machine	0.00%	203.54
Logistic Regression	0.00%	220.15

Figure 4.1 – Model Selection:Comparison of results

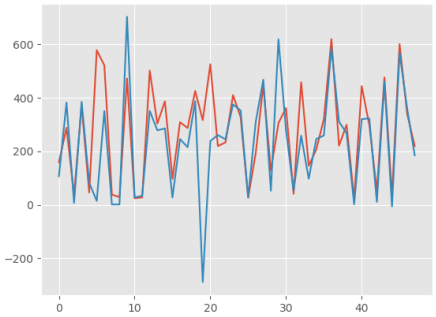
4.2 Plotting Predicted value and actual value of future stock price



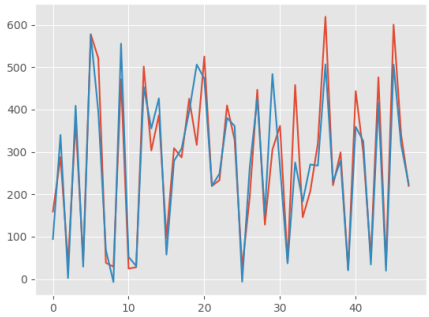
(a) Lasso Regression



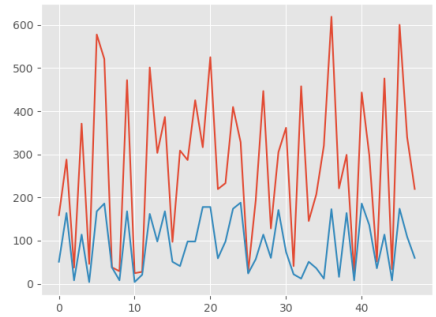
(b) Linear Regression



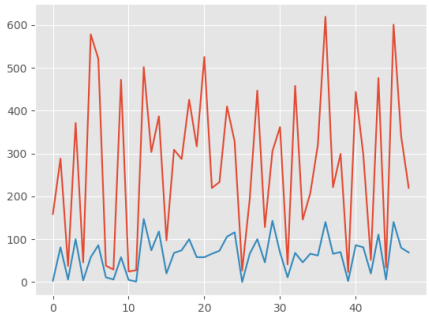
(c) Polynomial Regression



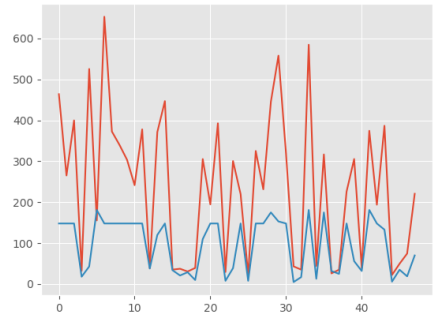
(d) Ridge Regression



(e) Logistic Regression



(f) K-nearest neighbour



(g) Support Vector Machine

Figure 4.2 – Blue line:shows predicted stock price whereas Red line:shows actual stock price

Chapter 5

Future Work

1. This is a generic application that can be used to optimize the task belonging to various fields like medicine, chemical, agriculture etc in which people have no knowledge of Machine learning so it would be easy for them if they has some trained model that do that task for them.

2. For example, in chemical engineering Machine Learning can be used for compound classification as it can be a time-consuming, tedious and error-prone task for a chemist to identify thousands of compounds manually. Thus this task can be automate by learning a ML model for each class of compounds.

3.The execution time to get the results is reduced to a great extent using trained Machine learning model.

Chapter 6

References

1. Application of Machine Learning Techniques to Trading.
https://medium.com/auquan/https-medium-com-auquan-machine-learning-techniques-trading-b7120cee4f05

2. Course on Machine Learning in Trading.
https://classroom.udacity.com/courses/ud501

3. Vatsal H.Shah. Machine Learning Techniques for Stock prediction.Foundations of Machine Learning | Spring 2007, Courant Institute of Mathematical Science, New York University