**Modeling and predicting stock market using machine learning**

**Chapter- 1**

**INTRODUCTION**

**1.1 Introduction: PYTHON WITH MACHINE LEARNING**

Machine learning is an application of artificial intelligence (AI) that provides systems the ability to automatically learn and improve from experience without being explicitly programmed. **Machine learning focuses on the development of computer programs** that can access data and use it learn for themselves.

Machine learning usually refers to the changes in systems that perform tasks associated with artificial intelligence (AI). Such tasks involve recognition, diagnosis, planning, robot control, prediction, etc.

**1.2 Methods:**

**1.Supervised machine learning algorithms** can apply what has been learned in the past to new data using labeled examples to predict future events. Starting from the analysis of a known training dataset, the learning algorithm produces an inferred function to make predictions about the output values. The system is able to provide targets for any new input after sufficient training. The learning algorithm can also compare its output with the correct, intended output and find errors in order to modify the model accordingly.

**2. Unsupervised machine learning algorithms**are used when the information used to train is neither classified nor labeled. Unsupervised learning studies how systems can infer a function to describe a hidden structure from unlabeled data. The system doesn’t figure out the right output, but it explores the data and can draw inferences from datasets to describe hidden structures from unlabeled data.

**3. Semi-supervised machine learning algorithms** fall somewhere in between supervised and unsupervised learning, since they use both labelled and unlabelled data for training – typically a small amount of labelled data and a large amount of unlabelled data. The systems that use this method are able to considerably improve learning accuracy. Usually, semi-supervised learning is chosen when the acquired labelled data requires skilled and relevant resources in order to train it / learn from it. Otherwise, acquiring unlabelled data generally doesn’t require additional resources.

**4. Reinforcement machine learning algorithms**is a learning method that interacts with its environment by producing actions and discovers errors or rewards. Trial and error search and delayed reward are the most relevant characteristics of reinforcement learning. This method allows machines and software agents to automatically determine the ideal behavior within a specific context in order to maximize its performance. Simple reward feedback is required for the agent to learn which action is best; this is known as the reinforcement signal.

Machine learning enables analysis of massive quantities of data. While it generally delivers faster, more accurate results in order to identify profitable opportunities or dangerous risks, it may also require additional time and resources to train it properly. Combining machine learning with AI and cognitive technologies can make it even more effective in processing large volumes of information.

**1.2 Our Project:**

This project about analysis and processing the previous stock data,then making the model from the previously known data, and then predicting the values through that model for the possible future values.

Data scientists use many different kinds of machine learning algorithms to discover patterns in big data that lead to actionable insights.

We have research on different model based on different algorithms like svm, single linear regression, multiple linear regression etc. implementing to our dataset to predict values and use our python knowledge to implement it in the simple way.

Then we will compare accuracy of each algorithm and then we can easily find out the which model giving us the most appropriate predicting value.

**Chapter 2**

**Basic of Modeling and predicting stock market using machine learning.**

**2.1 Overview:**

Machine learning, more specifically the field of predictive modeling is primarily concerned with minimizing the error of a model or making the most accurate predictions possible, at the expense of explainability. In applied machine learning we will borrow, reuse and steal algorithms from many different fields, including statistics and use them towards these ends.

To build a model we first need to focus on different algorithm then we are going to use in this project.

**2.2. Regression:**

Regression is the process of finding a model or function for distinguishing the data into continuous real values instead of using classes.

Mathematically, with a regression problem, one is trying to find the function approximation with the minimum error deviation.

In regression, the data numeric dependency is predicted to distinguish it. The main goal of regression algorithms is predicting, the discrete or a continues value.

The representation is a linear equation that combines a specific set of input values (x) the solution to which is the predicted output for that set of input values (y). As such, both the input values (x) and the output value are numeric.

The linear equation assigns one scale factor to each input value or column, called a coefficient and represented by the capital Greek letter Beta (B). One additional coefficient is also added, giving the line an additional degree of freedom (e.g. moving up and down on a two-dimensional plot) and is often called the intercept or the bias coefficient.

For example, in a simple regression problem (a single x and a single y), the form of the model would be:

y = B0 + B1\*x

In higher dimensions when we have more than one input (x), the line is called a plane or a hyper-plane. The representation therefore is the form of the equation and the specific values used for the coefficients.

**2.2.1**

**Steps 1:**

**Simple Linear Regression**

With simple linear regression when we have a single input, we can use statistics to estimate the coefficients.

This requires that you calculate statistical properties from the data such as means, standard deviations, correlations and covariance. All of the data must be available to traverse and calculate statistic

### Step2:

### Ordinary Least Squares

When we have more than one input we can use Ordinary Least Squares to estimate the values of the coefficients.

The Ordinary Least Squares procedure seeks to minimize the sum of the squared residuals. This means that given a regression line through the data we calculate the distance from each data point to the regression line, square it, and sum all of the squared errors together. This is the quantity that ordinary least squares seeks to minimize.

**Step 3:**

**Gradient Descent**

This operation is called Gradient Descent and works by starting with random values for each coefficient.

When there are one or more inputs you can use a process of optimizing the values of the coefficients by iteratively minimizing the error of the model on your training data. The sum of the squared errors are calculated for each pair of input and output values. A learning rate is used as a scale factor and the coefficients are updated in the direction towards minimizing the error. The process is repeated until a minimum sum squared error is achieved or no further improvement is possible.

**Step 4:**

### Regularization

There are extensions of the training of the linear model called regularization methods. These seek to both minimize the sum of the squared error of the model on the training data (using ordinary least squares) but also to reduce the complexity of the model

**2.3 Support vector machine:**

A support vector machine (SVM) is machine learning algorithm that analyzes data for classification and regression analysis. SVM is a supervised learning method that looks at data and sorts it into one of two categories. An SVM outputs a map of the sorted data with the margins between the two as far apart as possible. SVMs are used in text categorization, image classification, handwriting recognition and in the sciences.

A support vector machine is also known as a support vector network (SVN).

**2.4 polynomial linear regression:**

polynomial regression, a special case of multiple linear regression that adds terms with degrees greater than one to the model. The real-world curvilinear relationship is captured when you transform the training data by adding polynomial terms, which are then fit in the same manner as in multiple linear regression.

**The equation of Quadratic Equation or polynomial of degree 2 is :**

https://cdn-images-1.medium.com/max/600/1*Bg1Zx6fOeL0R8Z9G98A-SA.gif

**Similarly a Equation of degree 3 :**

https://cdn-images-1.medium.com/max/600/1*A2UDxVqLmv4KkV4yHkZ5EQ.gif

**Polynomial Degree n Would be like:**

**https://cdn-images-1.medium.com/max/600/1*PWT0R1EnTIaLSNcWWnItWA.gif**

**2.5 radial and sigmoidal function.**

**Radial:** A radial basis function (RBF) is a real-valued function whose value depends only on the distance from the origin, so that or alternatively on the distance from some other point called a center, so that Any that satisfies the property is a radial function.

Types: Gaussian, multiquadratic, inverse quadratic, inverse multiquadratic.

**Sigmoidal function:**

The sigmoid function is used mostly used in machine learning problem since we need to scale the data in some given specific range with a threshold.

using the sigmoid fuction

1/(1+ex)

will adjust all your data points between 0 and 1. If you want to adjust it to 2 or a higher value, just change the numerator and you are good to go.

**2.6 software and packages used:**

**We use variety of software and packages to implement different work like statistically calculation, graphs.**

**Software:**

1.python

2.anaconda

3.spyder

**Packages:**

1.sklearn

2.pandas

3.numpy

4.mathplotlib

**Chapter 3: Methodology**

**Processing the data, modelling, and predicting.**

**3.1 Processing:**

**We have data from hdfc, reliance and ongc.**

**First we need to see all attributes which we have as input for the model:**

X1= 'Open Price'

X2='High Price '

X3='Low Price '

X4='Last Traded Price '

X5='Close Price '

X6='Total Traded Quantity '

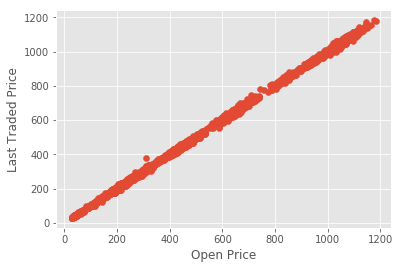
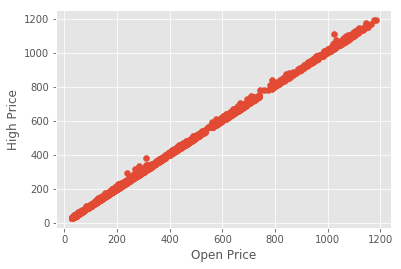
**And the output or prediction:**

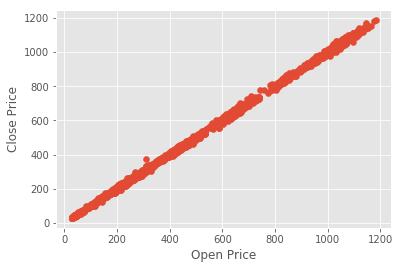
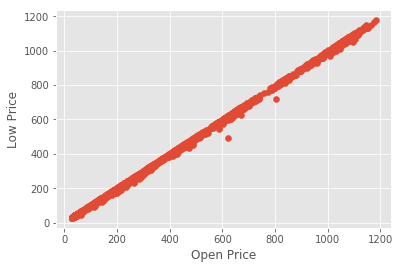
Y=’ Turnover (in Lakhs)’

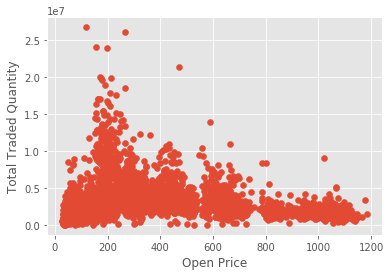
**3.1.2 scatter graphs:**

**Between input attributes:**

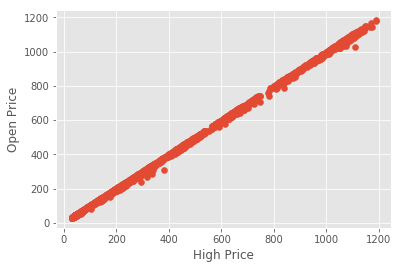
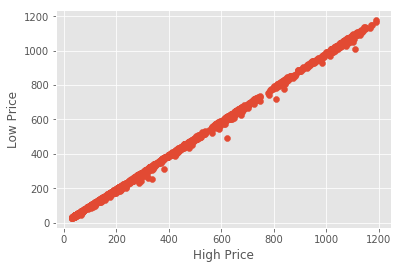
**for X1= ‘Open Price’**

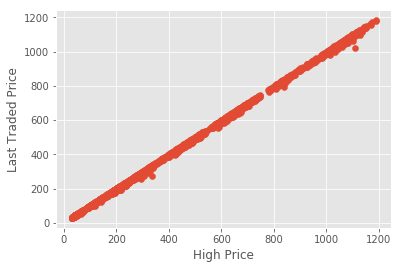
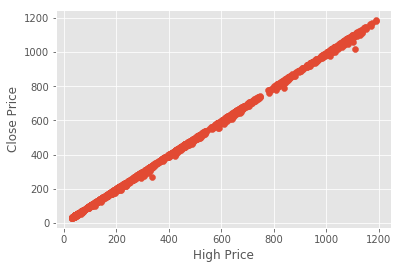


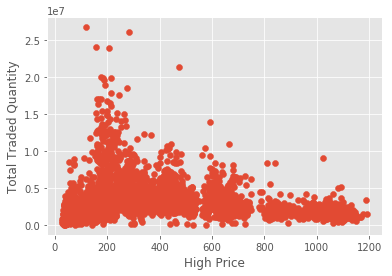




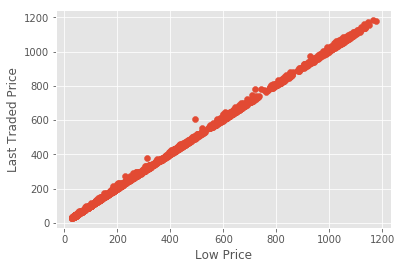
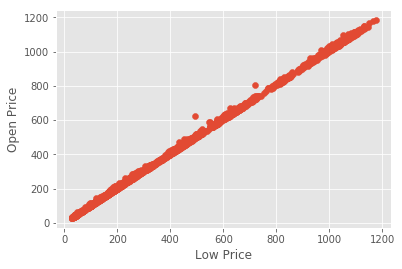
**For X2=’High Price ‘**

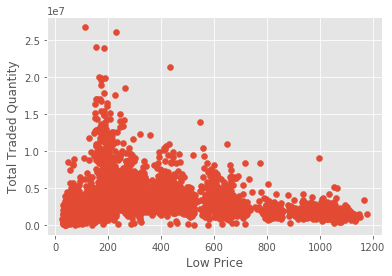
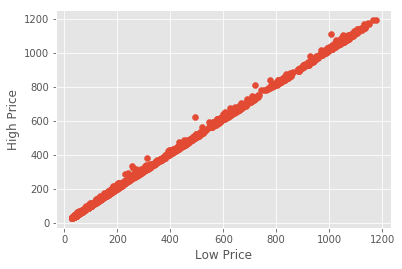


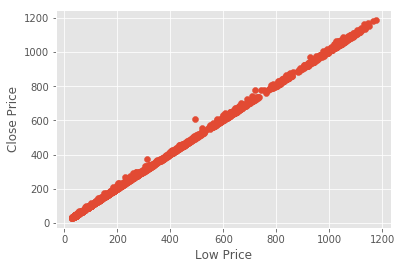




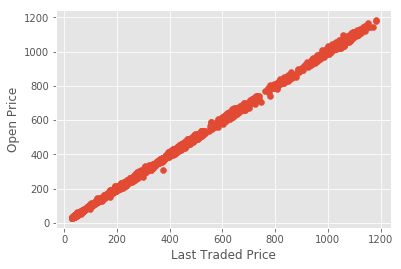
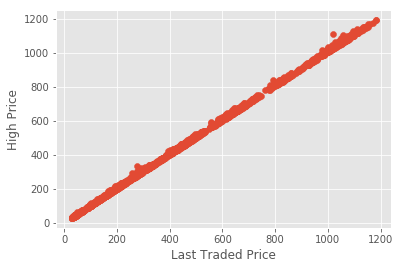
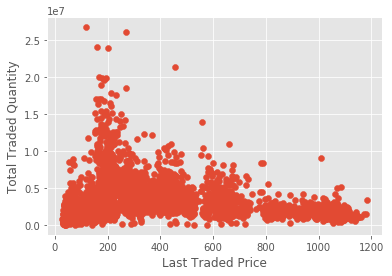
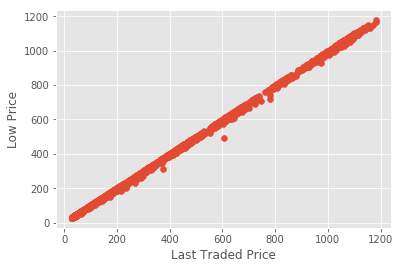
**For X3=’Low Price ‘**

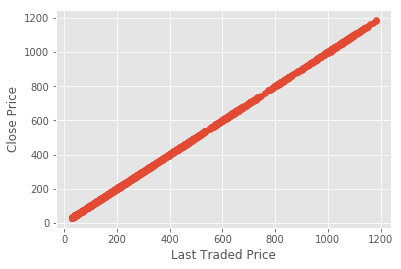




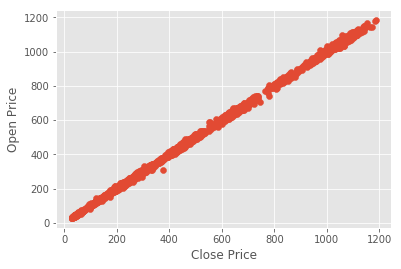
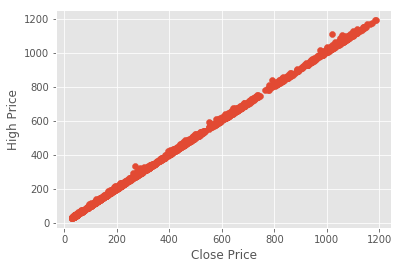
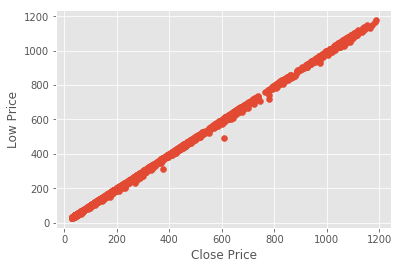
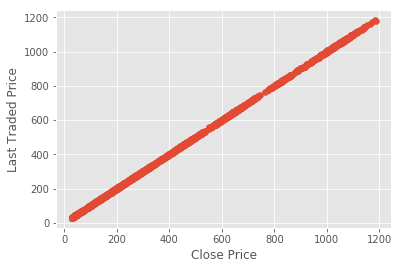
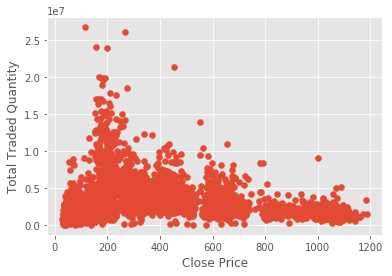


**X4=’Last Traded Price ‘**

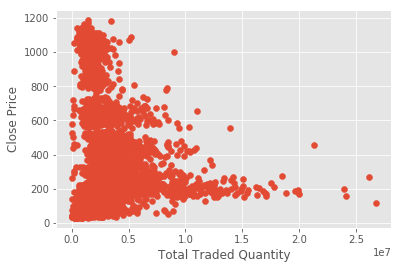
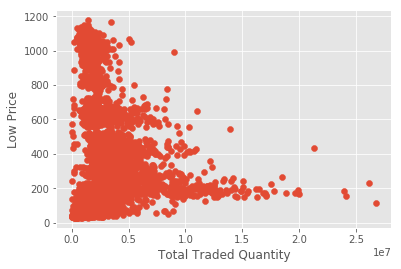
 

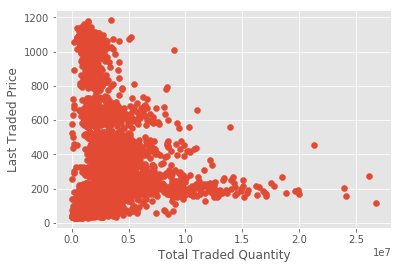


**X5=’Close Price ‘**

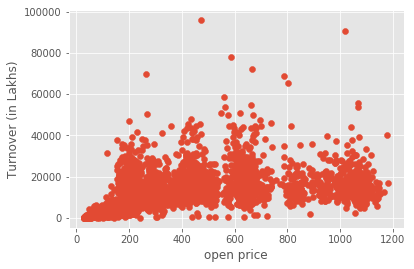
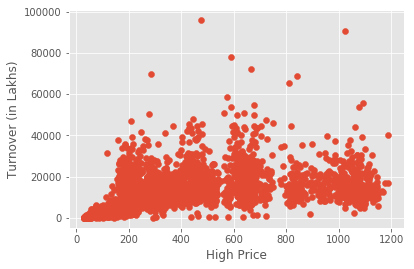


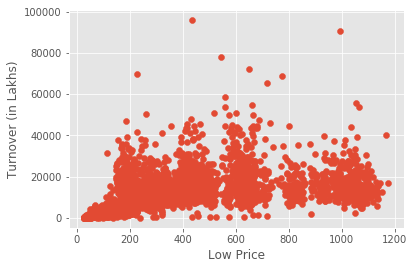
**X6=’Total Traded Quantity ‘**

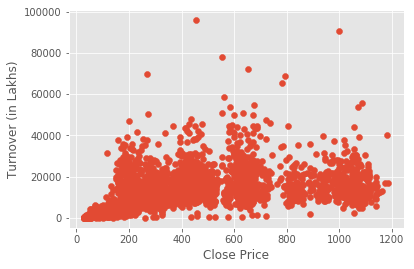




**Input vs output attribute:**







**\*\*\*By seeing scatter plots\*\*\***

**conclusions:**

**Multicollinearity and overfitting:** by seeing scatter plots of x1, x2, x3, x4, x5. As they are highly correlated and to avoid multicollinearity and overfitting we will take anyone of them in model building.

Therefore, x1 and x6 are taken for model making and to increase the accuracy of the model.

**3.2 Feature scaling, preprocessing, testing and training data:**

**Feature scaling** :

Generally, you want your features in machine learning to be in a range of -1 to 1. This may do nothing, but it usually speeds up processing and can also help with accuracy

**Preprocessing:** it is done to clean the data make it in more useful and deleting errors.

**testing and training data:** done to test the model first we train the model with the training data and then wetest the model with the testing model and then we find accuracy.

#pre-processing data

X = preprocessing.scale(X)

# Splitting the dataset into the Training set and Test set

from sklearn.cross\_validation import train\_test\_split

#accuracy libaility or confidence score are samething

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, Y, test\_size = 0.2, random\_state = 0)

# Feature Scaling

from sklearn.preprocessing import StandardScaler

sc\_X = StandardScaler()

X\_train = sc\_X.fit\_transform(X\_train)

X\_test = sc\_X.transform(X\_test)

sc\_y = StandardScaler()

y\_train = sc\_y.fit\_transform(y\_train)

**3.3 modelling algorithms using:**

1.linear regression

2.svm

3.polynomial multivariate regression

4.rbf function

5.sigmoid function

**Algorithm:**

for k in ['linear','rbf','sigmoid',’svm’]:

clf = svm.SVR(kernel=k)

clf.fit(X\_train, y\_train)

confidence = clf.score(X\_test, y\_test)

print(k,confidence)

X\_lately = X[-forecast\_out:]

X = X[:-forecast\_out]

df.dropna(inplace=True)

forecast\_set = clf.predict(X\_lately)

print(forecast\_set, confidence, forecast\_out)

from sklearn.preprocessing import PolynomialFeatures

poly\_reg = PolynomialFeatures(degree = 4)

X\_poly = poly\_reg.fit\_transform(X)

poly\_reg.fit(X\_poly, y)

lin\_reg\_2 = LinearRegression()

lin\_reg\_2.fit(X\_poly, y)

confidence = poly\_reg.score(X\_test, y\_test)

print('polynomial:')

**3.3 prediction and accuracy of algorithm:**

accuracy = clf.score(X\_test, y\_test)

print(accuracy)

X\_lately = X[-forecast\_out:]

X = X[:-forecast\_out]

df.dropna(inplace=True)

forecast\_set = clf.predict(X\_lately)

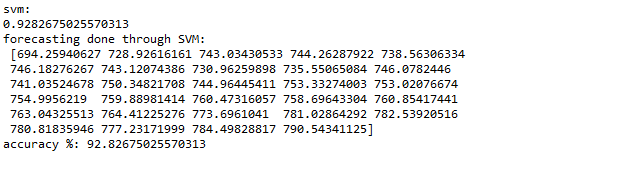
print(forecast\_set, confidence)

**Chapter 4: Results**

**Accuracy and predict values for next 30 days of the stocks turnover of the company.**

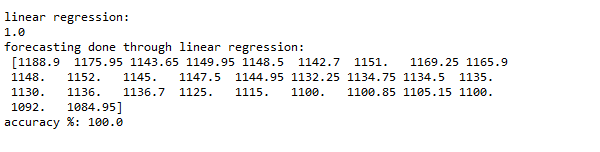
**1.SVM:**

**Accuracy%=92.83**



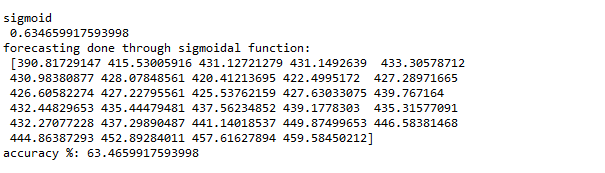
**2. linear regression:**

**Accuracy%=100.0**



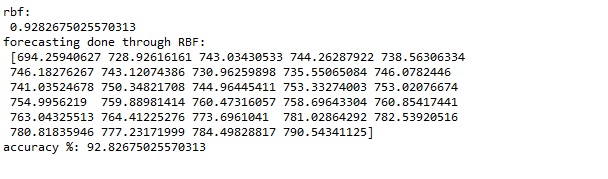
**3.sigmoidal function:**

**Accuracy%= 63.47**



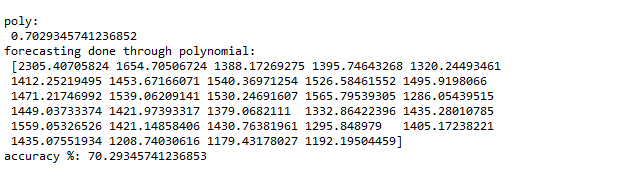
**4.rbf:**

**Accuracy%=92.83**



**5.polynomial multivariate regresssion:**

**Accuracy%=70.30**



**Chapter 5: Conclusion**

It was an amazing experience to work alongside such professional and welcoming people.

I’ll be taking a lot away from my time here with working at numeric InfoTech and have learnt a lot. The company plunged me into a wide array of challenges and gave me the opportunity to explore myself as a professional.

Since I’m only starting my career, this was a huge boost to my confidence and I feel a lot more comfortable with machine learning after this.

This has also opened up my eyes to the real world and how professionals in the Information Technology development sector work and live their lives every day. This will allow me to much better prepare myself for the future.

I do hope to come back to numeric InfoTech and work with them again.

I had the chance to meet some of the most talented and experienced people and learnt a lot from every single one of them.

I would like to thank everyone involved for making this such an amazing experience and do hope to keep in touch with them.