**CHAPTER 4**

**DEVELOPMENT PROCESS**

* 1. **REQUIREMENT ANALYSIS**

Requirements are a feature of a system or description of something that the system is capable of doing in order to fulfil the system’s purpose. It provides the appropriate mechanism for understanding what the customer wants, analyzing the needs assessing feasibility, negotiating a reasonable solution, specifying the solution unambiguously, validating the specification and managing the requirements as they are translated into an operational system.

* + 1. **PYTHON:**

Python is a dynamic, high level, free open source and interpreted programming language. It supports object-oriented programming as well as procedural oriented programming. In Python, we don’t need to declare the type of variable because it is a dynamically typed language.

For example, x=10. Here, x can be anything such as String, int, etc.

Python is an interpreted, object-oriented programming language similar to PERL, that has gained popularity because of its clear [syntax](https://whatis.techtarget.com/definition/syntax) and readability. Python is said to be relatively easy to learn and portable, meaning its statements can be interpreted in a number of [operatingsystem](https://whatis.techtarget.com/definition/operating-system-OS)s, including UNIX-based systems, Mac OS, MS-DOS, OS/2, and various versions of Microsoft Windows 98. Python was created by Guido van Rossum, a former resident of the Netherlands, whose favourite comedy group at the time was Monty Python's Flying Circus. The source code is freely available and open for modification and reuse. Python has a significant number of users.

**Features in Python**

There are many features in Python, some of which are discussed below

* Easy to code
* Free and Open Source
* Object-Oriented Language
* GUI Programming Support
* High-Level Language
* Extensible feature
* Python is Portable language
* Python is Integrated language
* Interpreted Language
  1. **ANACONDA**

Anaconda distribution comes with over 250 packages automatically installed, and over 7,500 additional open-source packages can be installed from [PyPI](https://en.wikipedia.org/wiki/Python_Package_Index) as well as the [conda](https://en.wikipedia.org/wiki/Conda_(package_manager)) package and virtual environment manager. It also includes a GUI, Anaconda Navigator,[[12]](https://en.wikipedia.org/wiki/Anaconda_(Python_distribution)#cite_note-12) as a graphical alternative to the command line interface (CLI).

The big difference between conda and the [pip package manager](https://en.wikipedia.org/wiki/Pip_(package_manager)) is in how package dependencies are managed, which is a significant challenge for Python data science and the reason conda exists.

When pip installs a package, it automatically installs any dependent Python packages without checking if these conflict with previously installed packages. It will install a package and any of its dependencies regardless of the state of the existing installation. Because of this, a user with a working installation of, for example, Google Tensorflow, can find that it stops working having used pip to install a different package that requires a different version of the dependent numpy library than the one used by Tensorflow. In some cases, the package may appear to work but produce different results in detail.

In contrast, conda analyses the current environment including everything currently installed, and, together with any version limitations specified (e.g. the user may wish to have Tensorflow version 2,0 or higher), works out how to install a compatible set of dependencies, and shows a warning if this cannot be done.

Open source packages can be individually installed from the Anaconda repository, Anaconda Cloud (anaconda.org), or the user's own private repository or mirror, using the conda install command. Anaconda, Inc. compiles and builds the packages available in the Anaconda repository itself, and provides binaries for Windows 32/64 bit, Linux 64 bit and MacOS 64-bit. Anything available on [PyPI](https://en.wikipedia.org/wiki/Python_Package_Index) may be installed into a conda environment using pip, and conda will keep track of what it has installed itself and what pip has installed.

Custom packages can be made using the conda build command, and can be shared with others by uploading them to Anaconda Cloud, [PyPI](https://en.wikipedia.org/wiki/Python_Package_Index) or other repositories.

The default installation of Anaconda2 includes Python 2.7 and Anaconda3 includes Python 3.7. However, it is possible to create new environments that include any version of Python packaged with conda.

### ANACONDA NAVIGATOR

Anaconda Navigator is a desktop [graphical user interface (GUI)](https://en.wikipedia.org/wiki/Graphical_user_interface) included in Anaconda distribution that allows users to launch applications and manage conda packages, environments and channels without using [command-line commands](https://en.wikipedia.org/wiki/Command-line_interface). Navigator can search for packages on Anaconda Cloud or in a local Anaconda Repository, install them in an environment, run the packages and update them. It is available for [Windows](https://en.wikipedia.org/wiki/Windows), [macOS](https://en.wikipedia.org/wiki/MacOS) and [Linux](https://en.wikipedia.org/wiki/Linux).

The following applications are available by default in Navigator:[[16]](https://en.wikipedia.org/wiki/Anaconda_(Python_distribution)#cite_note-16)

* [JupyterLab](https://en.wikipedia.org/wiki/Project_Jupyter#JupyterLab)
* [Jupyter Notebook](https://en.wikipedia.org/wiki/Project_Jupyter#Jupyter_Notebook)
* QtConsole
* [Spyder](https://en.wikipedia.org/wiki/Spyder_(software))
* [Glue](https://en.wikipedia.org/wiki/Glue_(software))
* [Orange](https://en.wikipedia.org/wiki/Orange_(software))
* [RStudio](https://en.wikipedia.org/wiki/RStudio)
* [Visual Studio Code](https://en.wikipedia.org/wiki/Visual_Studio_Code)
  + 1. **JUPYTER NOTEBOOK**

Jupyter [Notebook](https://en.wikipedia.org/wiki/Notebook_interface) (formerly IPython Notebooks) is a [web-based interactive](https://en.wikipedia.org/wiki/Rich_Internet_application) computational environment for creating Jupyter notebook documents. The "notebook" term can colloquially make reference to many different entities, mainly the Jupyter [web application](https://en.wikipedia.org/wiki/Web_application), Jupyter Python web server, or Jupyter document format depending on context. A Jupyter Notebook document is a [JSON](https://en.wikipedia.org/wiki/JSON) document, following a versioned schema, containing an ordered list of input/output cells which can contain code, text (using [Markdown](https://en.wikipedia.org/wiki/Markdown)), mathematics, plots and rich media, usually ending with the ".ipynb" extension.

Jupyter Notebook can connect to many kernels to allow programming in different languages. By default, Jupyter Notebook ships with the IPython kernel. As of the 2.3 release[[11]](https://en.wikipedia.org/wiki/Project_Jupyter#cite_note-releasenote23-11)[[12]](https://en.wikipedia.org/wiki/Project_Jupyter#cite_note-releasenote20-12) (October 2014), there are currently 49 Jupyter-compatible kernels for many programming languages, including [Python](https://en.wikipedia.org/wiki/Python_(programming_language)), [R](https://en.wikipedia.org/wiki/R_(programming_language)), [Julia](https://en.wikipedia.org/wiki/Julia_(programming_language)) and [Haskell](https://en.wikipedia.org/wiki/Haskell_(programming_language)).

The Notebook interface was added to IPython in the 0.12 release[[14]](https://en.wikipedia.org/wiki/Project_Jupyter#cite_note-releasenote012-14) (December 2011), renamed to Jupyter notebook in 2015 (IPython 4.0 – Jupyter 1.0). Jupyter Notebook is similar to the notebook interface of other programs such as [Maple](https://en.wikipedia.org/wiki/Maple_(software)), [Mathematica](https://en.wikipedia.org/wiki/Mathematica), and [SageMath](https://en.wikipedia.org/wiki/SageMath), a computational interface style that originated with Mathematica in the 1980s. According to [*The Atlantic*](https://en.wikipedia.org/wiki/The_Atlantic), Jupyter interest overtook the popularity of the Mathematica notebook interface in early 2018.

* 1. **RESOURCE REQUIREMENTS:**

**SOFTWARE REQUIREMENTS**:

|  |  |
| --- | --- |
| Operating System | Windows 7or later |
| Simulation Tool | Anaconda (Jupyter notebook) |
| Documentation | Ms – Office |

**HARDWARE REQUIREMENTS:**

|  |  |
| --- | --- |
| CPU type | Intel Pentium |
| Ram size | 4GB |
| Hard disk capacity | 80 GB |
| Keyboard type | Internet keyboard |
| Monitor type | 15 Inch colour monitor |
| CD -drive type | 52xmax |

* 1. **PROPOSED SYSTEM**

In this proposed model the use of deep learning algorithms such as Multilayer perceptron (MLP) and the Long short-term Memory (LSTM) are used to predict the price of the bitcoin for both long term and short-term dependencies. In predicting the future price in the market, traders and investors use the technical analysis or fundamental analysis. These multilayer perceptron uses the 2-hidden layer feed forward networks with a formulated number of hidden nodes and the Long short term memory which is a special type of RNN which allows the network to learn from its previous state, it also makes this network really useful for time-series data. The architecture used is many-to-one with various experimented time-window.

* + 1. **ADVANTAGES**
* It is convincing in high dimensional space
* It works well with clear with clear margin of separation
* It help to understand the result very nice
* It easily calculate the lager
  1. **SYSTEM ARCHITECTURE**
  2. **MODULE DESCRIPTION**
* Data collection
* Data pre-processing
* EDA concept
* Model implementation
* Prediction

**MODULE 1: DATA COLLECTION**

To collect the Dataset. Collecting data for training the ML model is the basic step in the machine learning pipeline. The predictions made by ML systems can only be as good as the data on which they have been trained. Following are some of the problems that can arise in data collection:

* Inaccurate data. The collected data could be unrelated to the problem statement.
* Missing data. Sub-data could be missing. That could take the form of empty values in columns or missing images for some class of prediction.
* Data imbalance. Some classes or categories in the data may have a disproportionately high or low number of corresponding samples. As a result, they risk being under-represented in the model.
* Data bias. Depending on how the data, subjects and labels themselves are chosen, the model could propagate inherent biases on gender, politics, age or region, for example. Data bias is difficult to detect and remove.

**MODULE 2: DATA PRE-PROCESSING**

Once the data is extracted from the twitter source as the datasets, this information has to be passed to the classifier. The classifier cleans the dataset by removing redundant data like stop words, emoticons in order to make sure that non textual content is identified and removed before the analysis.

Text pre-processing is an essential a part of any NLP method and the significance of the NLP pre-processing are

* To minimize indexing (or knowledge) records dimension of the textual content records

1. Stop words bills 20-30% of total phrase counts in a special textual content record
2. Stemming may just diminish indexing size as much as forty- 50%

* To make stronger the efficiency and effectiveness of the IR method

1. Stop words aren't valuable for shopping or textual content mining
2. Stemming used for matching the similar words in a text record

The sklearn.preprocessing package provides several common utility functions and transformer classes to change raw feature vectors into a representation that is more suitable for the downstream estimators.

In general, learning algorithms benefit from standardization of the data set. If some outliers are present in the set, robust scalers or transformers are more appropriate. The behaviors of the different scalers, transformers, and normalizers on a dataset containing marginal outliers is highlighted in [Compare the effect of different scalers on data with outliers](https://scikit-learn.org/stable/auto_examples/preprocessing/plot_all_scaling.html#sphx-glr-auto-examples-preprocessing-plot-all-scaling-py).

**Standardization, or Mean removal and Variance Scaling**

**Standardization**of datasets is a **common requirement for many machine learning estimators** implemented in scikit-learn; they might behave badly if the individual features do not more or less look like standard normally distributed data: Gaussian with **zero mean and unit variance.**

**Scaling features to a range**

In practice we often ignore the shape of the distribution and just transform the data to center it by removing the mean value of each feature, then scale it by dividing non-constant features by their standard deviation.

For instance, many elements used in the objective function of a learning algorithm (such as the RBF kernel of Support Vector Machines or the l1 and l2 regularizers of linear models) assume that all features are centered around zero and have variance in the same order. If a feature has a variance that is orders of magnitude larger than others, it might dominate the objective function and make the estimator unable to learn from other features correctly as expected.

An alternative standardization is scaling features to lie between a given minimum and maximum value, often between zero and one, or so that the maximum absolute value of each feature is scaled to unit size. This can be achieved using [MinMaxScaler](https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.MinMaxScaler.html" \l "sklearn.preprocessing.MinMaxScaler" \o "sklearn.preprocessing.MinMaxScaler) or [MaxAbsScaler](https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.MaxAbsScaler.html" \l "sklearn.preprocessing.MaxAbsScaler" \o "sklearn.preprocessing.MaxAbsScaler), respectively.

The motivation to use this scaling include robustness to very small standard deviations of features and preserving zero entries in sparse data.

[MaxAbsScaler](https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.MaxAbsScaler.html#sklearn.preprocessing.MaxAbsScaler) works in a very similar fashion, but scales in a way that the training data lies within the range [-1,1] by dividing through the largest maximum value in each feature. It is meant for data that is already cantered at zero or sparse data.

**Normalization**

**Normalization**is the process of **scaling individual samples to have unit norm.** This process can be useful if you plan to use a quadratic form such as the dot-product or any other kernel to quantify the similarity of any pair of samples.

This assumption is the base of the [Vector Space Model](https://en.wikipedia.org/wiki/Vector_Space_Model) often used in text classification and clustering contexts.

**MODULE 3: EDA concept (Exploratory data/Visualized Analysis)**

Exploratory Data Analysis or (EDA) understands the data sets by summarizing their main characteristics often plotting them visually. This step is very important especially when we arrive at modelling the data in order to apply Machine learning. Plotting in EDA consists of Histograms, Box plot, Scatter plot and many more. It often takes much time to explore the data. Through the process of EDA, we can ask to define the problem statement or definition on our data set which is very important.So in this tutorial; we will explore the data and make it ready for modelling.

**1. Importing the required libraries for EDA**

The libraries that are used in order to perform EDA (Exploratory data analysis)

**2. Loading the data into the data frame.**

Loading the data into the pandas data frame is certainly one of the most important steps in EDA, as we can see that the value from the data set is comma-separated. One thing to remember in this step is that uploaded files will get deleted when this runtime is recycled.

**3. Checking the types of data**

The datatypes because sometimes the MSRP or the price of the car would be stored as a string or object, if in that case, we have to convert that string to the integer data only then we can plot the data via a graph.

**4. Dropping irrelevant columns**

This step is certainly needed in every EDA because sometimes there would be many columns that we never use in such cases dropping is the only solution. In this case, the columns such as Engine Fuel Type, Market Category, Vehicle style, Popularity, Number of doors, Vehicle Size doesn't make any sense to me so I just dropped for this instance**.**

**5. Renaming the column**

In this instance, most of the column names are very confusing to read, so I just tweaked their column names. This is a good approach it improves the readability of the data set.

**6. Dropping the duplicate rows**

This is often a handy thing to do because a huge data set as in this case contains more than 10, 000 rows often have some duplicate data which might be disturbing, so here I remove all the duplicate value from the data-set. For example prior to removing I had 11914 rows of data but after removing the duplicates 10925 data meaning that I had 989 of duplicate data.

**7. Dropping the missing or null values.**

An outlier is a point or set of points that are different from other points. This is mostly similar to the previous step but in here all the missing values are detected and are dropped later. Now, this is not a good approach to do so, because many people just replace the missing values with the mean or the average of that column.

**8. Detecting Outlier**

Sometimes they can be very high or very low. It’s often a good idea to detect and remove the outliers. Because outliers are one of the primary reasons for resulting in a less accurate model. Hence it’s a good idea to remove them. The outlier detection and removing that I am going to perform is called IQR score technique. Often outliers can be seen with visualizations using a box plot.

**MODULE 4: MODEL IMPLEMENTATION**

**Long Short-Term Memory** (**LSTM)**

* A general **LSTM**unit is composed of a cell, an input gate, an output gate, and a forget gate. The cell remembers values over arbitrary time intervals, and three gates regulate the flow of information into and out of the cell. LSTM is well-suited to classify, process, and predict the time series given of unknown duration.

**Root Mean Square Error (RMSE)**

In machine learning, it is extremely helpful to have a single number to judge a model's performance, whether it be during training, cross-validation, or monitoring after deployment. Root mean square error is one of the most widely used measures for this. Root Mean Square Error (RMSE) is the standard deviation of the residuals (prediction errors). Residuals are a measure of how far from the regression line data points are; RMSE is a measure of how spread out these residuals are. (RMSE) is a frequently used measure of the differences between values (sample or population values) predicted by a model or an estimator and the values observed.

**MONTE CARLO SIMULATION**

Monte Carlo simulation is a computerized mathematical technique to generate random sample data based on some known distribution for numerical experiments. This method is applied to risk quantitative analysis and decision making problems. This method is used by the professionals of various profiles such as finance, project management, energy, manufacturing, engineering, research & development, insurance, oil & gas, transportation, etc.

**MODULE 5: PREDICTION**

* LSTM and MLP are very powerful in sequence prediction problems because they’re able to store past information. This is important in case of predicting the future bitcoin prices. The pre-processed data is provided as the input to the model and then compile the model after the compilation of the model the evaluation is done.
* The prediction is visualized by plotting the graph with importing the matplotlib library. Two hidden LSTM layers were chosen. For a time series task two layers is enough to find non-linear relationships among the data for a time series task. Three and four layers were tested but didn’t improve validation performance. Several layers can be required for tasks such as image recognition when the number of features is significantly large.