**Chapter 1**

**INTRODUCTION**

# 1.1 About the Company

Vision Digital India in association with Imperial Group for 10 years has inspired to provide digital, analytical and app developing skills in the present generation to meet corporate needs. To support the Digital India initiative, Vision Digital India is offering courses like Certified Machine Learning Course in association with Google, Certified Analytics for all in association with IBM and iOS Mobile Application Development in association with Apple.

Vision Digital India aims in providing high-quality education services to all the students who have the zeal to learn new technology which is going to take over the computing world for the next 5 years. They offer first in its kind of most innovative programs on Machine Learning and Data Analytics, started this journey with the interaction of college students across India on topic “Machine Learning – The Future” and covered more than 5,000 students and also part of various conferences related to machine learning.

Vision Digital India creates courses that are precisely tailored to meet the demands of the corporate world, and ensure that students are industry-ready. Through its endeavor, Vision Digital India is trying to make PM Narendra Modi’s flagship campaign Digital India a resounding success and it is succinctly summed up in their motto ‘Enable Digital’. Vision Digital India has secured the help of industry professionals who know the real world, how it works, the latest updates and emerging trends in the industry.

# 1.2 Introduction to Machine Learning

The term Machine Learning was coined by Arthur Samuel in 1959, an American pioneer in the field of computer gaming and artificial intelligence and stated that “it gives computers the ability to learn without being explicitly programmed”.

Machine learning is a subfield of artificial intelligence (AI). Because of this, machine learning facilitates computers in building models from sample data in order to automate decision-making processes based on data inputs. Any technology user today has benefitted from machine learning**.**

Intelligence, as we know, is the ability to acquire and apply knowledge. Knowledge is the information acquired through experience. Experience is the knowledge gained through exposure(training). Summing the terms up, we get artificial intelligence as the “copy of something natural(i.e. human beings) ‘WHO’ is capable of acquiring and applying the information it has gained through exposure.” Intelligence is composed of:

* Reasoning
* Learning
* Problem Solving
* Perception
* Linguistic Intelligence

## Need for Artificial Intelligence

1. To create expert systems that exhibit intelligent behavior with the capability to learn, demonstrate, explain and advice its users.

2. Helping machines find solutions to complex problems like humans do and applying them as algorithms in a computer-friendly manner.

Arthur Samuel, a pioneer in the field of artificial intelligence and computer gaming, coined the term “Machine Learning”. He defined machine learning as – “Field of study that gives computers the capability to learn without being explicitly programmed”. In a very layman manner, Machine Learning(ML) can be explained as automating and improving the learning process of computers based on their experiences without being actually programmed i.e. without any human assistance. The process starts with feeding good quality data and then training our machines(computers) by building machine learning models using the data and different algorithms. The choice of algorithms depends on what type of data do we have and what kind of task we are trying to automate.

And in 1997, Tom Mitchell gave a “well-posed” mathematical and relational definition that “A computer program is said to learn from experience E with respect to some task T and some performance measure P, if its performance on T, as measured by P, improves with experience E”.

Artificial intelligence is used in the production unit in most big manufacturing companies. AI system is used to give a specific shape to an object, move objects from one place to another, etc. This application is also used in the management of most companies to get their tasks efficiently done on time.

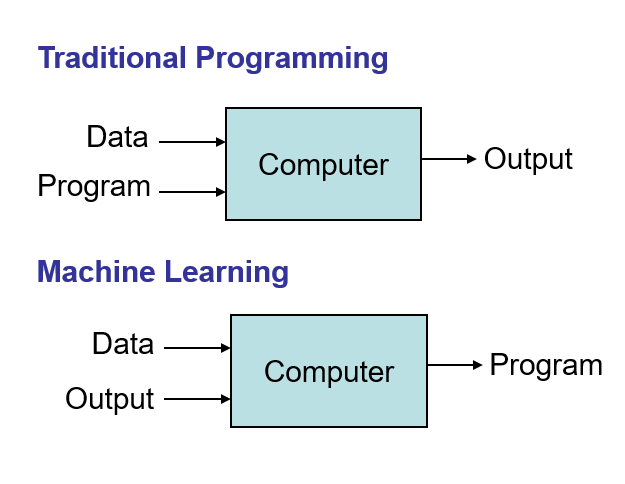


Figure 1.1 Traditional program and Machine learning

**Basic Difference between ML and Traditional Programming**

Traditional Programming: We feed in DATA (Input) + PROGRAM (logic), run it on a machine and get the output as in figure1.1. Machine Learning: We feed in DATA(Input) Output, run it on the machine during training and the machine creates its own program(logic), which can be evaluated while testing as in figure 1.1

**Working of ML**

Gathering past data in any form suitable for processing. The better the quality of data, the more suitable it will be for modeling.Data Processing – Sometimes, the data collected is in the raw form and it needs to be pre-processed.

Example: Some tuples may have missing values for certain attributes, an, in this case, it has to be filled with suitable values in order to perform machine learning or any form of data mining. Missing values for numerical attributes such as the price of the house may be replaced with the mean value of the attribute whereas missing values for categorical attributes may be replaced with the attribute with the highest mode. This invariably depends on the types of filters we use. If data is in the form of text or images then converting it to numerical form will be required, be it a list or array or matrix. Simply, Data is to be made relevant and consistent. It is to be converted into a format understandable by the machine.Divide the input data into training, cross-validation and test sets. The ratio between the respective sets must be 6:2: 2. Building models with suitable algorithms and techniques on the training set. Testing our conceptualized model with data that was not fed to the model at the time of training and evaluating its performance using metrics such as F1 score, precision, and recall.

1.3 Introduction to House Price Prediction**:**

According to the US Census Bureau, 560,000 houses were sold in the United States in 2016. In addition, 65% of all American families owned houses in 2016. For the Americans who sold and bought these houses, a good housing price prediction would decisions in their lives. A recent report from the Zillow Group, a popular housing database website, indicates that house sellers and buyers are increasingly turning to online research in order to estimate house prices before contacting real estate agents. Researching how much the house you are interested in is worth it on your own can be difﬁcult for multiple reasons. One particular reason is that there many factors that inﬂuence the potential price of a house, making it more complicated for an individual to decide how much a house is worth on their own without external help. This can lead to people making poorly informed decisions about whether to buy or sell their houses and which prices are reasonable. Because houses are long term investments, it is imperative that people make their decisions with the most accurate information possible. Therefore, housing websites such as Zillow, Trulia and Redﬁn 1, exist to provide estimations of housing valuations based on the houses’ characteristics, at no cost.

However, the estimations provided by these housing websites are not always accurate. For example, Zillow states that their housing price prediction algorithm, called

“Zestimate”, only estimates 54.4% of houses within the 5% of their actual sale prices. For Trulia, only 48.2% of houses have Trulia-estimated prices to be within the 5% range of their actual sold prices. Therefore, the ﬁrst question of this project is whether I can outperform Zestimate’s prediction score or come close to it. In this project, I deﬁne the prediction score as the percentage of houses whose estimated prices fall within the 5% range of their actual sold prices. dictions, I compute Zillow’s prediction scores and use them as the baselines to see how well my own models perform. I choose Zillow’s estimator as a benchmark instead of its competitors’ because Zillow is widely regarded as the most popular housing website due to its large databases of 110 million houses and their 11 years of expertise in pricing estimations. According to Hitwise, a consumer analytics company, Zillow’s market share, based on online visits to the site, is 27.2% in 2016, while the numbers for Trulia and Redﬁn are 9.4% and 3.7%, respectively.

# Chapter 2

**LITERATURE REVIEW**

The House price prediction data set is a multivariate data set consisting of 570 records which split into training and testing samples. Each sample contains 8 features/attributes such as house id, area, number of rooms, number of bricks, etc. which describes the various biological features of a house. Similar to human learning behaviors, machine learning relies on past experiences to make judgments. The more experiences it accumulates the better judgments are made.

Sirmans, Macpherson and Zietz (2005) provides a study of 125 papers that use hedonicpricing model to estimate house prices in the past decade [4]. The paper provides a listof 20 attributes that are frequently used to specify hedonic pricing models. This datasetcontains 12 attributes on this list. Moreover, Sirmans, Macpherson, and Zietz (2005) also discusses the effects of some variables on housing price. For example, number of bathrooms is usually positively correlated to the final sale price. Out of 40 times appearing in housing price studies, this attribute has a positive effect 34 times and is statistically significant 35 times. On average, keeping other variables unchanged, an increase of 1 bathroom leads to 10% to 12% increase in the property’s value. Similarly, my paper shows that,based on the dataset of sold houses in five counties, the number of bathroom has a statistically significant and positive effect on sold price. On average, an increase of 1 bathroom could increase a house’s price by $15,787.

Cebula (2009) conducts a study on the housing prices in the City of Savannah, Georgia using the hedonic pricing model [3]. The paper’s data contains 2,888 single-family houses for the period between 2000 and 2005. Cebula (2009) shows that the log price of houses is positively and significantly correlated with the number of bathrooms, bedrooms, fireplaces, garage spaces, stories and the total square feet of the house.

Selim (2009) seeks to study the effects of different housing characteristics on housing prices in Turkey using two different methods: hedonic pricing model and artificial neural network [1]. The paper’s dataset, which was collected from the 2004 Household Budget Survey Data for Turkey, contains 5,741 observations with 46 housing characteristics.

Jirong, Mingcang, and Liuguangyan (2010) uses support vector machine (SVM) regression to forecast the housing prices in China in between 1993 and 2002 and in certain district in Tangshan city in between 2000 to 2002 [6]. The paper utilizes the genetic algorithm to tune the hyper-parameters in the SVM regression model. The error scores for the SVM regression model for both China and a Tangshan City’s district are both lower than 4%.This indicates that the SVM regression model perform well in forecasting housing prices in China.

In the Singapore’s housing market, Fan, Ong and Koh (2006) uses decision tree model study the housing characteristics’ effects on prices [5]. The paper concludes that the owners of 2-room to 4-room flats are more concerned with the flats’ basic characteristics such as model type and age more than the owners of 5-or-more-room flats. Moreover,owners of executive flats care more about the services characteristics such as the neighborhood location and recreational facilities than basic housing characteristics.

Tay and Ho (1991/1992) compared the pricing prediction between regression analysis and artificial neural network in predicting apartments’ prices in Singapore [2]. They found that the neural network model outperforms regression analysis model with a mean absolute error of 3.9%.

A technical paper which is given by Artima Developer. Retrieved 2007-03-22 on the topic of "The fate of reducing () in Python 3000" have given their thesis on machine learning predictions on house price prediction where they used various models of machine learning which has given pictorial representations of the data using graphs, bar charts and used built in function of data preprocessing which made the work easier evaluation of classification model.

**Chapter 3**

**SYSTEM ANALYSIS**

The house predicting dataset is used for predicting the price of the house using the Linear Regression algorithm. The dataset consists of information about the area, number of rooms, direction, compactness, etc. When we work on this kind of data, we need to see which feature is important for us and which is not. Our main aim is to make a model which can give us a correct prediction of the house price. We are going to use linear Regression for this dataset and see if this gives us a good accuracy or not.

**3.1 Purpose**

The purpose of this project was to gain an introductory exposure to programmatic data analysis concepts by analyzing the factors that determine the stage of the cancer based on the inputs given by the user. The project makes heavy use of NumPy, Pandas and Linear Regression algorithm.

**3.2 Scope**

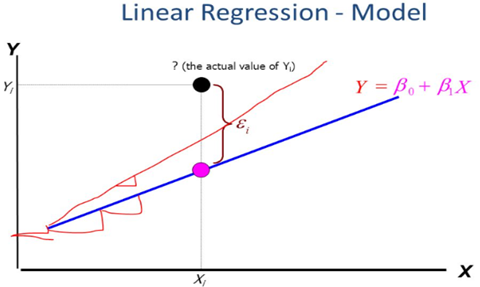


Figure 3.1 Linear Regression model

This project has a scope that is allowed to maintain the datasets, it is used to train it, relevant and critical, for the application to run. It uses the Linar Regression to predict the maximum accuracy. It describes the data and explains the relationship between one dependent binary variable and one or more nominal, ordinal, interval or ratio-level independent variables as shown in figure 3.1. Other uses of Linear Regression algorithm are studying engine performance from test data in automobiles, market research studies, customer survey result in analysis and other business applications.

**3.3 Motivation**

The main motivation behind the selection of this project is to design, develop and implement a machine learning model that helps us to predict the price of the house whether it is malignant or benign using the features provided by the user in the dataset. It also helps in reducing the eliminating human error and making the learning system more reliable and easy with higher accuracy and automated decisions to be produced.

**3.4 Software Requirements**

1. Anaconda Navigator

2. Jupyter Notebook

3. Operating system: Windows 10/7/8

4. Numpy Library

5. Pandas Library

6. sklearn Library

7. Python 3

**3.5 Hardware Requirements**

1. 32 GB ROM

2. 64-bit processor

3. 8 GB memory

**Chapter 4**

**DETAILED SYSTEM DESIGN**

## 4.1 High-level design

High-level design explains the architecture that would be used for developing software products. The architecture diagram provides an overview of an entire system, identifying the main components that would be developed for the product and their interfaces.

## 4.1.1 Architecture of IPython Notebook

The notebook extends the console-based approach to interactive computing in a qualitatively new direction, providing a web-based application suitable for capturing the whole computation process: developing, documenting, and executing code, as well as communicating the results. The Jupyter notebook combines two components:

* A web application: a browser-based tool for interactive authoring of documents that combine explanatory text, mathematics, computations, and their rich media output.
* Notebook documents: a representation of all content visible in the web application, including inputs and outputs of the computations, explanatory text, mathematics, images, and rich media representations of objects.

## The IPython kernel

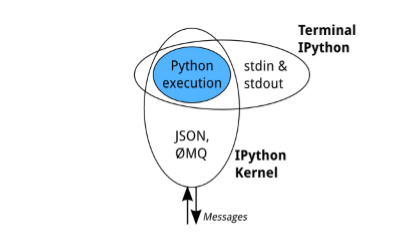


Figure 4.1 Kernel Execution

All the interfaces, like The Notebook, the Qt console, IPython console in the terminal, and third party interfaces use the IPython Kernel. The IPython Kernel is a separate process which is responsible for running user code, and things like computing possible completions. Frontends, like the notebook or the Qt console, communicate with the IPython Kernel using JSON messages sent over ZeroMQ sockets. The core execution machinery for the kernel is shared with terminal IPython as shown in figure 4.1. A kernel process can be connected to more than one frontend simultaneously. In this case, different frontends will have access to the same variables.

## The Notebook

The Notebook frontend does something extra. In addition to running your code, it stores code and output, together with markdown notes, in an editable document called a notebook as shown in figure 4.2. When you save it, this is sent from your browser to the notebook server, which saves it on disk as a JSON file with a .ipynb extension.

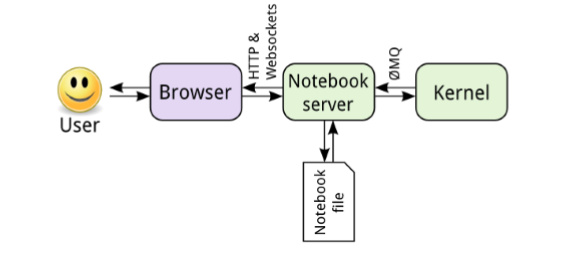


Figure 4.2 User to Notebook Interface

The notebook server, not the kernel, is responsible for saving and loading notebooks, so you can edit notebooks even if you don’t have the kernel for that language you just won’t be able to run code. The kernel doesn’t know anything about the notebook document: it just gets sent cells of code to execute when the user runs them.

## 4.2 The Notebook design

A design notebook is a way for a designer or engineer to keep a history of his or her design project from start to finish. It is a place to record research, observation, ideas, drawings, comments, and questions during the design process.

## 4.2.1 Notebook dashboard

When you create a new notebook document, you will be presented with the notebook name, a menu bar, a toolbar, and an empty code cell as shown in figure 4.3.

* Notebook name: The name displayed at the top of the page, next to the Jupyter logo, reflects the name of the .ipynb file. Clicking on the notebook name brings up a dialog that allows you to rename it. Thus, renaming a notebook from “Untitled0” to “My first notebook” in the browser, renames the Untitled0.ipynb file to My first notebook.ipynb.
* Menu bar: The menu bar presents different options that may be used to manipulate the way the notebook functions.
* Toolbar: The toolbar gives a quick way of performing the most-used operations within the notebook, by clicking on an icon.
* Code cell: the default type of cell; read on for an explanation of cells.

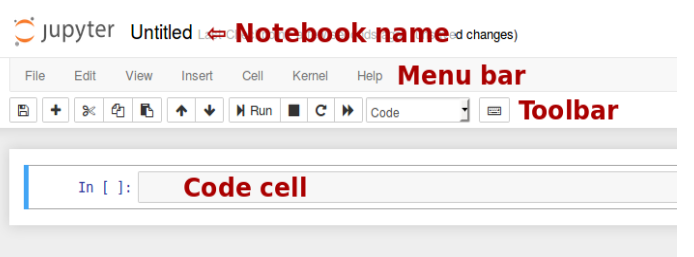


Figure 4.3 Notebook Dashboard

## 4.2.2 Structure of a Notebook document

The notebook consists of a sequence of cells. A cell is a multiline text input field, and its contents can be executed by using Shift-Enter, or by clicking either the “Play” button the toolbar, or “Cell Run” in the menu bar. The execution behavior of a cell is determined by the cell’s type. There are three types of cells: code cells, markdown cells, and raw cells. Every cell starts off being a code cell, but its type can be changed by using a drop-down on the toolbar.

* **Code cells**

A code cell allows you to edit and write new code, with full syntax highlighting and tab completion. The programming language you use depends on the kernel, and the default kernel (IPython) runs Python code. When a code cell is executed, the code that it contains is sent to the kernel associated with the notebook. The results that are returned from this computation are then displayed in the notebook as the cell’s output.

* **Markdown Cells**

You can document the computational process in a literate way, alternating descriptive text with code, using rich text. In IPython this is accomplished by marking up text with the Markdown language. The corresponding cells are called Markdown cells. The Markdown language provides a simple way to perform this text markup, that is, to specify which parts of the text should be emphasized (italics), bold, form lists, etc. When a Markdown cell is executed, the Markdown code is converted into the corresponding formatted rich text. Markdown allows arbitrary HTML code for formatting.

* **Raw Cells**

Raw cells provide a place in which you can write output directly. Raw cells are not evaluated by the notebook. Raw cells arrive in the destination format unmodified. For example, you can type full LaTeX into a raw cell, which will only be rendered by LaTeX.

## 4.2.3 The Output format

A code cell can have a variety of outputs (stream data or rich mime-type output). These correspond to messages produced as a result of executing the cell. All outputs have an output\_type field, which is a string defining what type of output it is.

## Stream output and Display data

{ "output\_type" : "stream",

"name" : "stdout",

# or stderr

"text" : ["multiline stream text"],

}

Rich display outputs, as created by display\_data messages, contain data keyed by mime-type. This is often called a mime-bundle and shows up in various locations in the notebook format and message spec. The metadata of these messages may be keyed by mime-type as well. This program explanation can be seen below.

{ "output\_type" : "display\_data",

"data" : {

"text/plain" : ["multiline text data"],

"image/png": ["base64-encoded-png-data"],

"application/json": {

# JSON data is included as-is

"json": "data", }, },

"metadata" :

{ "image/png": {

"width": 640,

"height": 480, }, }, }

* **Execute result**

Results of executing a cell are stored in execute\_result outputs. execute\_result outputs are identical to display\_data, adding only a execution\_count field, which must be an integer. This program explanation can be seen below.

{ "output\_type" : "execute\_result",

"execution\_count": 42,

"data" : { "text/plain" : ["multiline text data"],

"image/png": ["base64-encoded-png-data"],

"application/json": {

# JSON data is included as-is

"json": "data",

},

}, "metadata" : {

"image/png": {

"width": 640,

"height": 480,

}, }, }

## Error

Failed execution may show a traceback.

{ 'ename' : str, # Exception name, as a string

'evalue' : str, # Exception value, as a string

# The traceback will contain a list of frames,

# represented each as a string. 'traceback' : list, }

## Raw NBConvert cells

A raw cell is defined as content that should be included unmodified in nbconvert output. For example, this cell could include raw LaTeX for nbconvert to pdf via latex, or restructured text for use in Sphinx documentation. The notebook authoring environment does not render raw cells. The only logic in a raw cell is the format metadata field. If defined, it specifies which nbconvert output format is the intended target for the raw cell. When outputting to any other format, the raw cell’s contents will be excluded.

{

"cell\_type" : "raw",

"metadata" : {

# the mime-type of the target nbconvert format.

# nbconvert to formats other than this will exclude this cell.

"format" : "mime/type"

},

"source" : ["some nbformat mime-type data"] }

**Chapter 5**

**IMPLEMENTATION**

Implementation is the process of defining how the system should be built, ensuring that it is operational and meets quality standards. It is a systematic and structured approach for effectively integrating a software-based service or component into the requirements of end-users.

## 5.1 Overview of system implementation

The plan contains an overview of the system, a brief description of the major tasks involved in the implementation, the overall resources needed to support the implementation effort and any site-specific implementation requirements.

## 5.1.1 Selection of programming language – Python

Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. It's high-level built-in data structures, combined with dynamic typing and dynamic binding, make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together. Python's simple, easy to learn syntax emphasizes readability and therefore reduces the cost of program maintenance. Python supports modules and packages, which encourages program modularity and code reuse. The Python interpreter and the extensive standard library are available in source or binary form without charge for all major platforms and can be freely distributed.

Programmers prefer python because of the increased productivity it provides. Since there is no compilation step, the edit-test-debug cycle is incredibly fast. Debugging Python programs is easy. A bug or bad input will never cause a segmentation fault. Instead, when the interpreter discovers an error, it raises an exception. When the program doesn't catch the exception, the interpreter prints a stack trace.

A source-level debugger allows inspection of local and global variables, evaluation of arbitrary expressions, setting breakpoints, stepping through the code a line at a time, and so on. On the other hand, often the quickest way to debug a program is to add a few print statements to the source. The fast edit-test-debug cycle makes this simple approach very effective. The various IDE for python is spider, pycharm, atom, jupyter notebook to name a few. The ide used in this program is jupyter notebook by Anaconda.

## 5.1.2 Implementation support

## Anaconda

Anaconda is a free and open-source distribution of the Python and R programming languages for data science and learning-related applications (large-scale data processing, predictive analytics, scientific computing), that aims to simplify management and deployment. Anaconda3 includes Python 3.6. Anaconda Navigator is a desktop graphical user interface (GUI) included in Anaconda distribution that allows users to launch applications and manage anaconda packages, environments, and channels without using command-line commands. Navigators 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, macOS, and Linux. The following are the system requirements

* License: Free use and redistribution under the terms of the Anaconda End User License Agreement.
* Operating system: Windows Vista or newer, 64-bit macOS 10.10+, or Linux, including Ubuntu, RedHat, CentOS 6+, and others. Windows XP supported on Anaconda versions 2.2 and earlier. See lists. Download it from our archive.
* System architecture: 64-bit x86, 32-bit x86 with Windows or Linux, Power8 or Power9. Minimum 3 GB disk space to download and install.

**Installation of Anaconda on Windows**

The most common way of downloading anaconda is to go to www.python.org and choose the appropriate version and click on download. The following steps must be followed to install anaconda, after downloading it.

1. Download the Anaconda installer.
2. Optional: Verify data integrity with MD5 or SHA-256. More info on hashes
3. Double click the installer to launch. Click
4. Click Next.
5. Read the licensing terms and click “I Agree”.
6. Select an install for “Just Me” unless you’re installing for all users (which requiresWindows Administrator privileges) and click next.
7. Select a destination folder to install Anaconda and click the Next button. See FAQ.
8. Choose whether to add Anaconda to your PATH environment variable. We recommend not adding Anaconda to the PATH environment variable since this can interfere with other software. Instead, use Anaconda software by opening Anaconda Navigator or the Anaconda Prompt from the Start Menu.
9. Choose whether to register Anaconda as your default Python 3.6. Unless you plan on installing and running multiple versions of Anaconda, or multiple versions of Python, accept the default and leave this box checked.
10. Click the Install button. If you want to watch the packages Anaconda is installing, click Show Details.
11. Click the Next button.
12. Optional: To install VS Code, click the Install Microsoft VS Code button. After the install completes click the Next button.
13. Or to install Anaconda without VS Code, click the Skip button.
14. After your install is complete, verify it by opening Anaconda Navigator, a program that is included with Anaconda: from your Windows Start menu, select the shortcut Anaconda Navigator. If Navigator opens, you have successfully installed Anaconda. If not, check that you completed each step above, then see our Help page.

**5.1.3 Implementation using Linear regression**

**Supervised Learning :** Supervised learning is the Data mining task of inferring a function from labeled training data. The training data consist of a set of training examples. In supervised learning, each example is a pair consisting of an input object (typically a vector) and the desired output value (also called the supervisory signal).A supervised learning algorithm analyzes the training data and produces an inferred function, which can be used for mapping new examples as shown in figure 5.1.

* Labeled data: Data consisting of a set of training examples, where each example is a pair consisting of an input and the desired output value (also called the supervisory signal, labels, etc.)
* Classification: The goal is to predict discrete values, e.g. {1, 0}, {True, False}, {spam, not spam}.
* Regression: The goal is to predict continuous values, e.g. home prices.

A regression problem is when the output variable is a real or continuous value, such as “salary” or “weight”. Many different models can be used; the simplest is the linear regression. It tries to fit data with the best hyper-plane which goes through the points.

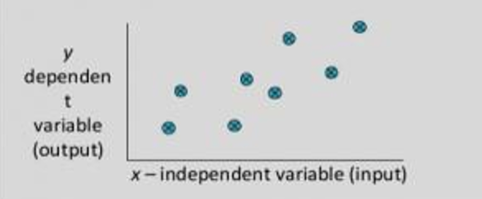


Figure 5.1 Dependency between Two Dataset

## Supervised learning workflow

## 

Figure 5.2 Supervised Learning Workflow

The supervised learning workflow as shown in figure 5.2 tells us how the training and testing of the machine take place. There are certain rules/steps which we have to follow in the ordered to train our machine these steps are as follows:

1. One has to take a data set in order to train and test the machine, the data set should relevant to our problem, for which they are training the machine.
2. Then they have to find a relation between all the attribute of the data set and have to find the dependency of the data set with each other, this help to select most appropriate attributes to train the machine.
3. Then they have to break our data set in two sets: a) Training data set (consist of 70 to 85 % of data set), b) Test data set (consist of about 15 to 30 % of the data set).
4. They select the most appropriate algorithm to train the machine, in initial stage of machine there are some errors that can be resolved with the error function they have used in our algorithm in each repeated iterations of training.
5. After the machine is trained with the algorithm used and training data with the minimum error they have to calculate the accuracy of the machine by using the testing data.
6. The expected result is labeled and matches with the original result to check the accuracy.

## 5.2 Pseudocode

Pseudocode is an informal high-level description of the operating principle of a computer program or other algorithm. It uses the structural conventions of a normal programming language but is intended for human reading rather than machine reading. Pseudocode resembles, but should not be confused with, programs that can be compiled without errors.

**5.2.1 Reading the CSV Data File and Cleaning the Data**



Figure 5.3 Code for Reading the CSV Data File and Cleaning the Data

As shown in figure 5.3, the first step to train the machine is to read and clean the dataset which we are using to train and test the machine. We use read\_csv() function which is defined under the pandas module, the read\_csv() function takes one argument which is the name of dataset file name.

In the next line, we are using pandas.DataFrame.astype() function which we are using to cast a pandas objects to specific Dtype in this case we are typecasting the attribute ‘DIAGNOSIS’ whose value is not ‘B’ as an integer value.

In the next step, we are dropping the column value which contains ‘unnamed’ value using pandas.Series.str.Contains (pat,case=False). After this we will check for the attributes which contain the NULL values, we first use df.isnull ().sum () this give how many null values are present in the total data set.

**5.2.2 Removing the least dependent attribute from dataset**

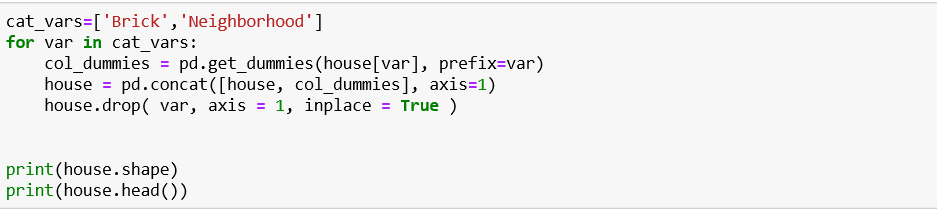


Figure 5.4 Code to Remove the least dependent attribute from Dataset

We have to remove all the attributes from the data set which are least dependable. As a result of detection does not depend on these data attributes. For this case we are dropping the attribute such as [“brick”,” neighborhood”] and assign rest attribute as for X-axis +parameters as shown in figure 5.4. We have to choose the parameters for x, we assign the x value to new variable x1 which does not contain the prediction value. As we do not want diagnosis as x parameter instead we assign diagnosis as y parameters to judge the output result with respect to y-axis.

**5.2.3 Training the Model and Testing the Model**

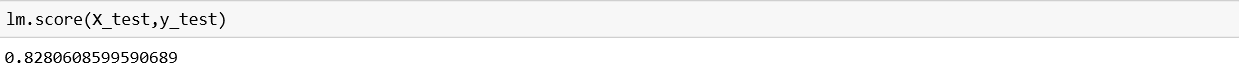
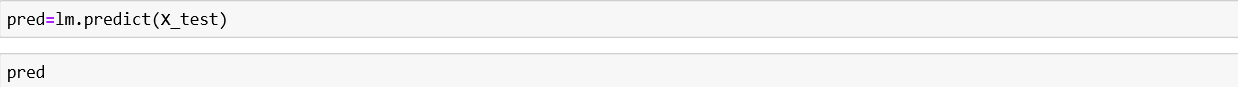
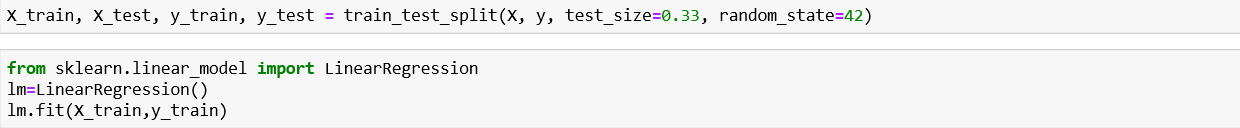


Figure 5.5 Code to Train and Test the Model

The above figure shows the code to split the dataset into the training and testing dataset. We distribute the data set in the ratio of (70:30) % for the training and testing part, then we train our machine using linear regression which is supported in sklearn library. In order to achieve this we split data set as x\_train and x\_test and Y\_train and Y\_test using train\_test\_split module, after which we set our model as linear Regression which is defined in linear\_model, then we use lm.fit(x\_train,Y\_train ) to train the model, after training gets over we test this against the testing data and predict the accuracy of our model using lm.pred(), and then check the score of model as shown in figure 5.5.The accuracy of this model using linear regression is 82.80%.

# Chapter 6

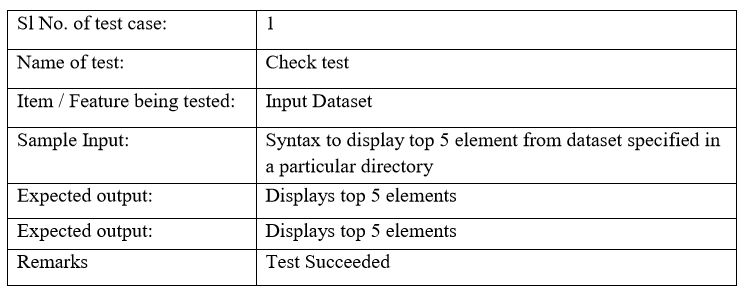
**TESTING**

Software Testing is the process used to help identify the correctness, completeness, security, and quality of the developed computer software. Testing is the process of technical investigation and includes the process of executing a program or application with the intent of finding errors.

# 6.1 Unit Testing

Unit testing is a level of software testing where individual units/ components of the software are tested. The purpose is to validate that each unit of the software performs as designed. A unit is the smallest testable part of any software. It usually has one or a few inputs and usually a single output. Unit testing has been done to check whether the cancer dataset is loaded correctly. In table 6.1 to check whether the dataset is loaded accordingly as the output shown in figure 6.1.

**Table 6.1 Unit test case for Dataset Loading**



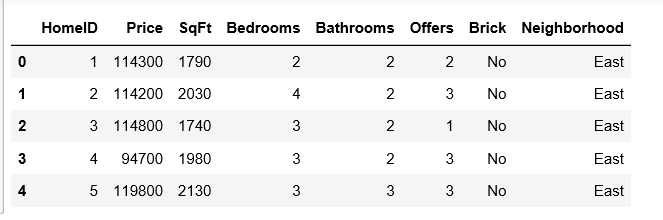
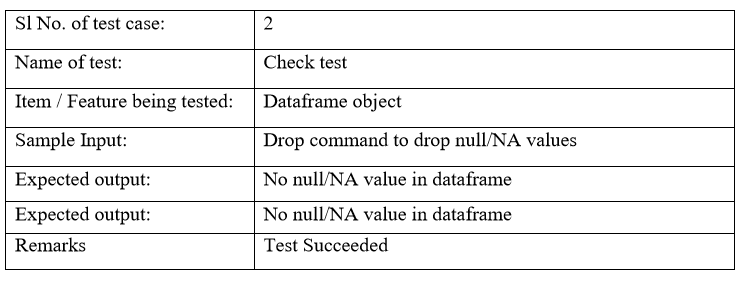


Figure 6.1 Dataset loaded

In table 6.2, Checking is done for null/NA value after dropping.

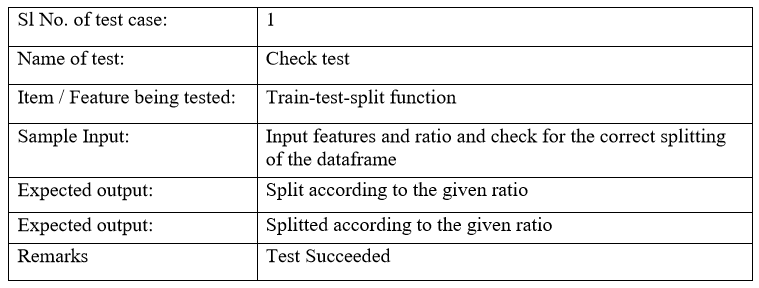
**Table 6.2 Unit test to check for null/NA value**



# 6.2 Integration Testing

Integration testing is done to test the modules/components when integrated to verify that they work as expected i.e. to test the modules which are working fine individually does not have issues when integrated. It is used to check the train and test split function as shown in table 6.3.

**Table 6.3 Integration test to check for train-test-split**

****

# 6.3 System Testing

System Testing (ST) is a black-box testing technique performed to evaluate the complete system the system's compliance against specified requirements. The software or hardware is testing conducted on a complete, integrated system to evaluate the system’s compliance with its specified requirements. System testing takes, as its input, all of the ‘integrated’ software components that have successfully passed integration testing and also the software system itself integrated with any applicable hardware system(s). In table 6.4, one is checking accuracy using a logistic regression algorithm.

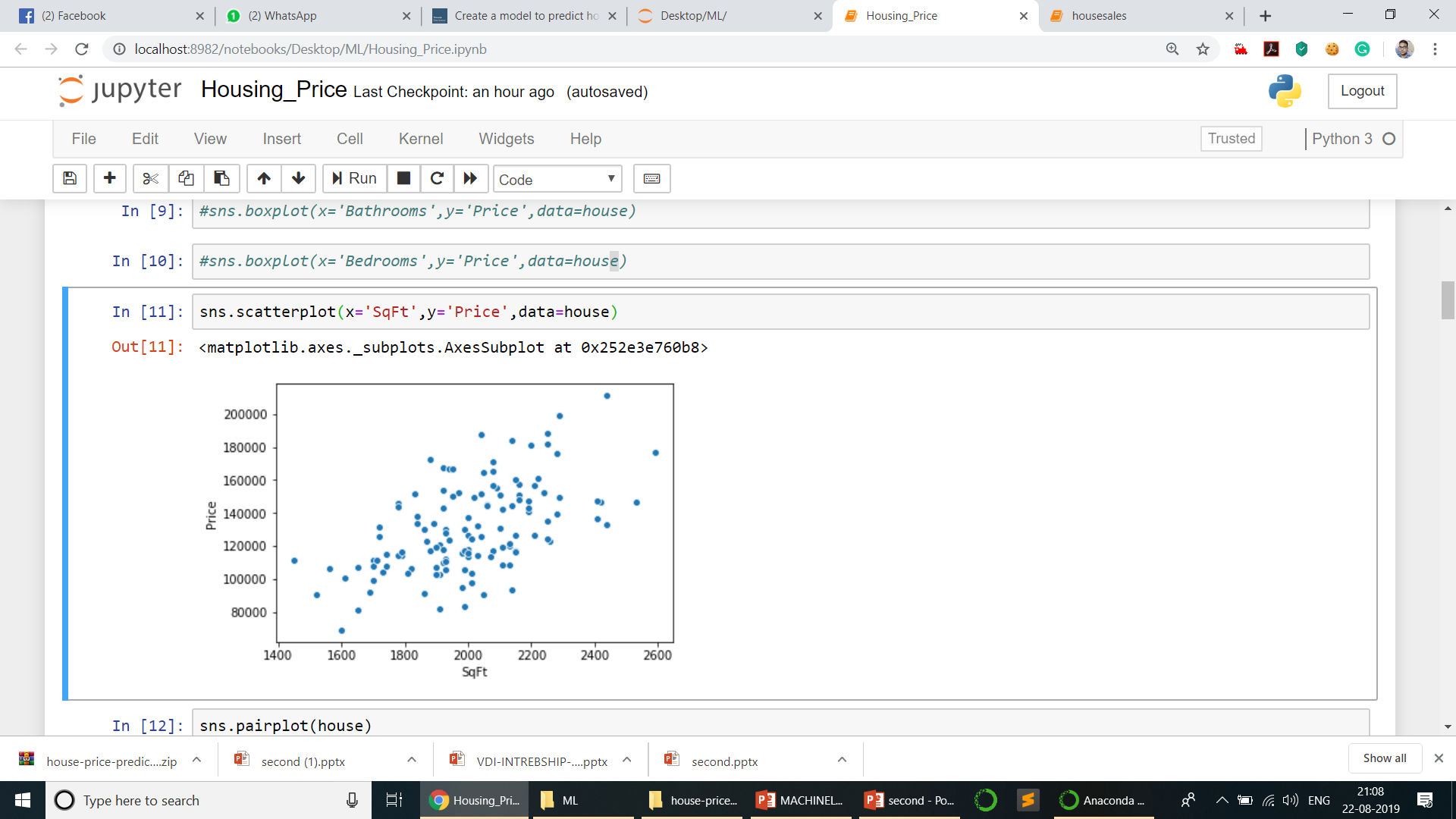
**Table 6.4 System test to check the accuracy using Linear Regression**

|  |  |
| --- | --- |
| Sl No. of a test case: | 1 |
| Name of Test: | Check test |
| Item/Feature being tested: | Learning algorithms efficiency |
| Sample Input: | Training features and output features |
| Expected output: | Some accuracy of the training model |
| Actual output: | Accuracy:82.80% |
| Remarks: | Test succeeded |

# Chapter 7

# DISCUSSION OF RESULTS

Matplotlib.pyplot.scatter showing the variation of ‘perimeter\_mean’ with respect to ‘area\_mean’ attributes of the size of the tumor. pandas.Dataframe.corr compute pairwise correlations of columns, excluding null/NA values as shown in figure 7.1 below.

Figure 7.1 Relation between price and sqft of the house

The model is able to predict the house price for given input features up to 82.80% accurately using linear regression as shown in figure 7.2.

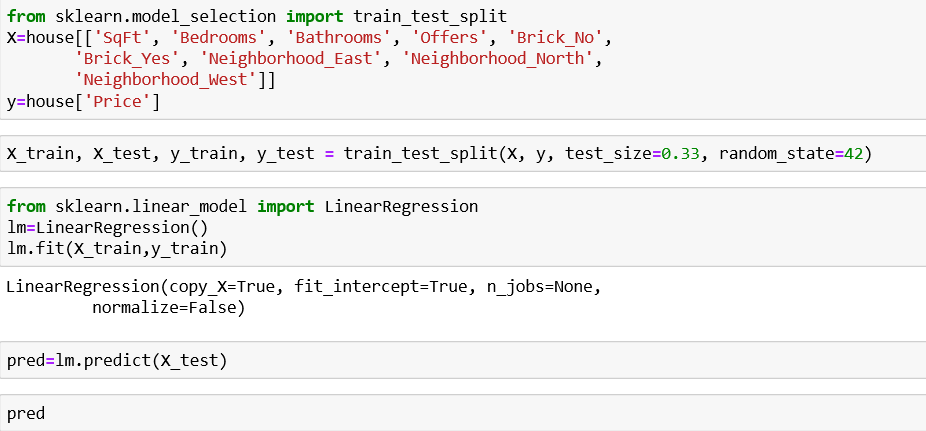
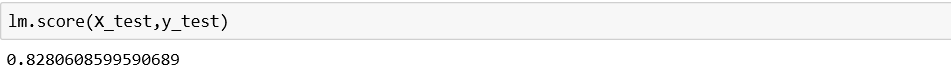
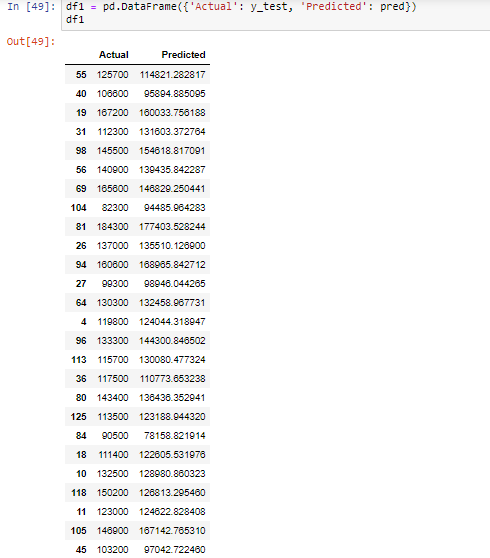
****

Figure 7.2 Accuracy prediction for the model



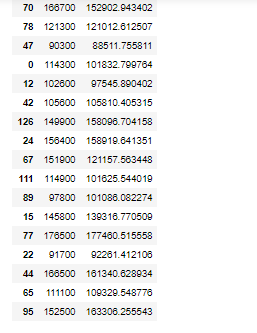


Figure 7.3 Actual and predicted house prices

# The figure 7.3 shows the actual house price for given home id and predicted house price based on the training dataset.

# o3.png

# Figure 7.4 Bar graph representation of actual and predicted price of houses

# The actual and predicted house prices can be represented using bar graph as shown in above figure 7.4.

# Chapter 8

**CONCLUSION AND FUTURE ENHANCEMENTS**

As the conclusion of the project has been represented briefly to show the work carried out in building a system model. It also talks about the other enhancements that can be made into the project so that it is user-friendly and make a more accurate classification of data.

# Conclusion

The project “House Price Prediction” is successful in predicting the house price with a good accuracy rate depending on the various attributes given by the user in the collected data set, thereby eliminating the human error as the decision process is successfully automated.

It shows Linear Regressionmodeling is fairly simple. It successfully classifies the house price based on the important features provided. For datasets with a small number of variables, it is a viable method. This project is a good introduction to training and test sets which are very important components not just to data science but to statistical learning overall.

# Future Enhancements

We can carry out the following enhancements in the project which would make it more suitable and an intelligent model to classify various real-world problems. Better regression techniques can be applied to obtain the best possible accuracy. It can be converted into a real-time analysis system for insights on-the-fly. It can include more illustrative graphs and plots for a deeper understanding of the dataset. The application can be used to forecast the future trends in price as the real dataset of the house price is collected from the website. Then the logistic regression model is trained on this dataset. After training the model, House price prediction is done.

# REFERENCES

[1] Gang-Zhi Fan, Seow Eng Ong, and Hian Chye Koh. “Determinants of House Price: A Decision Tree Approach”. In: Urban Studies 43.12 (2006)

[2] Danny P. H. Tay and David K. H. Ho. “Artificial Intelligence and the Mass Appraisal of Residential Apartments”. In: Journal of Property Valuation and Investment 10.2 (1992)

[3] Rochard J. Cebula. “The Hedonic Pricing Model Applied to the Housing Market of the City of Savannah and Its Savannah Historic Landmark District”. In: The Review of Regional Studies 39.1 (2009)

[4] Hasan Selim. “Determinants of house prices in Turkey: Hedonic regression versus artificial neural network”. In: Expert Systems with Applications 36 (2009)

[5] G. Stacy Sirmans, David A. Macpherson, and Emily N. Zietz. “The Composition of Hedonic Pricing Models”. In: Journal of Real Estate Literature 13.1 (2005)

[6] Gu Jirong, Zhu Mingcang, and Jiang Liuguangyan. “Housing price based on genetic algorithm and support vector machine”. In: Expert Systems with Applications 38 (2011)

[7] Neal Norwitz; Barry Warsaw (2006-06-29). "PEP 361 -- Python 2.6 and 3.0 Release Schedule". Retrieved 2012-10-07.

[8] Artima Developer "The fate of reducing () in Python 3000". Retrieved 2007-03-22

[9] www.developer.amazon.com/docs/custom-skills/steps-to-buildaprediction

[10] www.w3schools.com/python/default.asp

[11] www.tutorialspoint.com/randomforest

[12] www.codeacademy.com