# IN-HOUSE PROJECT REPORT

ON

# House price prediction

IV Semester

BACHELOR OF COMPUTER SCIENCE

(B.Sc.CAMS)

By

**N.Anand kumar (223621P)**

Under the esteemed guidance of

##### P.LAKSHMI (P.hd)



###### DEPARTMENT OF COMPUTER SCIENCE

**P. B. SIDDHARTHA COLLEGE OF ARTS AND SCIENCE**

VIJAYAWADA – 10

(An Autonomous college under the jurisdiction of Krishna University, Machilipatnam.

A college with potential for excellence, re – accredited at the level ‘A+’ by NAAC, Bangalore)

2022-2025

#### P. B. SIDDHARTHA COLLEGE OF ARTS AND SCIENCE VIJAYAWADA – 10

An Autonomous college under the jurisdiction of Krishna University, Machilipatnam.

(A college with potential for excellence, re – accredited at the level ‘A+’ by NAAC, Bangalore)

#### DEPARTMENT OF COMPUTER SCIENCE



**CERTIFICATE**

This is to certify that the project entitled “**HOUSE PRICE PREDICTION”,** is a bonified work of **N.ANAND KUMAR (223621P)** submitted in partial fulfillment of the requirements for the award of the degree of BACHELOR OF COMPUTER SCIENCE (B.Sc.) from P. B. Siddhartha College of Arts and Science, Vijayawada.

Project Guide Head of the Department

Examiner

#### DECLARATION

I hereby declare that the In-House Project (IHP) report entitled “HOUSE PRICE PREDICTION” submitted by me in the partial fulfillment of the requirement for the award of degree in BACHELOR OF COMPUTER SCIENCE (B.SC.) is the record of work originally carried out by from May 2024 – June2024 under the guidance of **P.LAKSHMI** ,vision computers IT Solutions private limited, Vijayawada.

N.ANAND KUAMR (223621p)

# ACKNOWLEDGEMENT

It is our privilege to acknowledge with deep sense of gratitude and devotion for keen personal interest and invaluable guidance rendered by our Project Guide **P.LAKSHMI**.

We express my profound sense of gratitude to my guide and mentor **BADHRAJA, MBA.,M.Sc.,(IT),M.TECH.(CSE),MA(ASTRO),PH.D.,GoldMedal,Wor**

**ld Book of records,FSIESRP,Assistant Professor,Dept.of Computer Science**,

for her support and Guidence for the successful completion of this project.

Our respects and regards to **.K. Mr SUDHIR, M.Sc. (IT), M. Tech (CSE)** Head of the Department, co-ordinator of P.B. Siddhartha College of Arts & Science, for his invaluable suggestions that helped us in successful completion of the project.

We are grateful to our Principal **Dr Ramesh Meka**. P.B. Siddhartha College of Arts & Science, for his cooperation and encouragement.

Finally, we also thank all the faculty of **Dept. of Computer Science, PBSC, our friends**, and all our family members who with their valuable suggestions and support, directly or indirectly helped us in completing this project work.

N.ANAND KUMAR

# ABSTRACT

Investment is a business activity on which most people are interested in this globalization era. There are several objects that are often used for investment, for example, gold, stocks and property. In particular, property investment has increased significantly. Housing price trends are not only the concern of buyers and sellers, but it also indicates the current economic situation. There are many factors which has impact on house prices, such as numbers of bedrooms and bathrooms. Even the nearby location, a location with a great accessibility to highways, expressways, schools, shopping malls and local employment opportunities contributes to the rise in house price. Manual house predication becomes difficult, hence there are many systems developed for house price prediction. We have proposed an advanced house prediction system using linear regression. This system aim is to make a model which can give us a good house pricing prediction based on other variables. We are going to use Linear Regression for this dataset and hence it gives a good accuracy. This house price prediction project has two modules namely, Admin and User. Admin can add location and view the location. Admin has authority to add density on the basis of per unit area. User can view the location and see the predicted housing price for the particular location.

**Contents**

|  |  |  |
| --- | --- | --- |
| SNO | **Topics** | **PageNo** |
| **1** | **Introduction** |  |
| **2** | **System Analysis** |  |
| **3** | **System Requirements** |  |
| **4** | **System Environment** |  |
| **5** | **Source Code** |  |
| **6** | **Implementation & Output screens** |  |
| **7** | **Conclusion** |  |
| **8** | **Biblography** |  |

**Introduction**

This article refers together with latest Forecast on Research predictions considering trends to further plan their Economics. The main motivation of the project FORECASTING VARIATIONS ON HOUSE PRICE was to make the best possible prediction of house prices by using linear regression algorithm and decision tree finding out which among them is best suitable for predicting the price with low error rate. This is an interesting problem because most of the people will eventually buy/sell a home. This problem allows us, as house price analysts, to learn more about the housing market and helps with making more informed decisions. The analysis that was done in this paper is mainly based on the Data sets of Vijayawada, A.P. because of unexpected changes in price of houses in and around Vijayawada due to

Emergence of new capital city Amravati because of formation of new state. In this paper, we try to demonstrate the Regression techniques which are suitable to our problem. The brief overview of all the reference taken are as follows: Linear Regression is used which uses more than one attributes for prediction.

**System Analysis**

**Existing system**

* Much manu coding is required for analysis. On large data it is very difficult to write the code. Analysis on large data is very difficult. Traditional expert systems are not suited to do this kind of analysis. Day by day data is increasing so for prediction sake data is important. But programming may take much time to do analysis on large data

**Disadvantages of existing system**

* No training on data
* Much time with expert systems
* Less efficient

**Proposed System**

The proposed system involves with Machine Learning supervised approach is used to analyse house price. This approach gathers house information in the form of dataset. This dataset later on gets preprocessed to build the model. In order to build model, training process is required. The model gets build with regression algorithms

advantages of proposed system

Useful when there are large datasets.

•Prediction performance is high

## SYSTEM REQUIREMENTS

**Hardware Requirements :**

* Processor - I3
* RAM - 8GB(min)
* Hard Disk - 340 GB

**Software Requirements:**

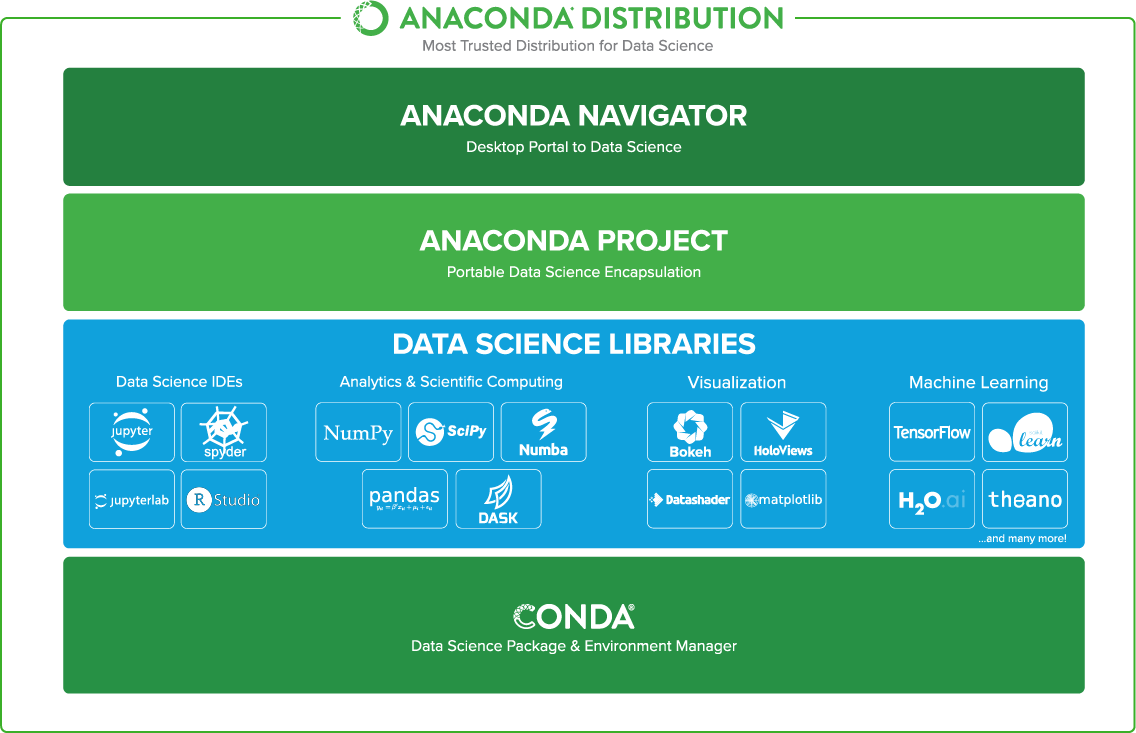
* Operating System : Windows 10 onwards
* Tools : pycharm/jupyter notebook (Anaconda)
* Programming Language : Python
* Libraries : Pandas,matplot-lib,seaborn

**System Environment**

## ANACONDA

Anconda is a complete, open source data science package with a community of over 6 million users. It is easy to download and install, and it is supported on Linux, MacOs, and Windows.

The distribution comes with more than 1,000 data packages as well as the Conda package and virtual environment manager, so it elminates the need to learn to install each library independently. As Anaconda’s website says, “The Python and R conda packages in the Anaconda Repository are curated and compiled in our secure environment so you get optimized binaries that ‘just work’ on your system”.



Anaconda Distribution

## What is Anaconda Navigator?

Anaconda Navigator is a desktop graphical user interface (GUI) included in Anaconda® distribution that allows you to launch applications and easily manage conda packages, environments, and channels without using command-line commands. Navigator can search for packages on Anaconda Cloud or in a local Anaconda Repository. It is available for Windows, macOS, and Linux.

## Why use Navigator?

In order to run, many scientific packages depend on specific versions of other packages. Data scientists often use multiple versions of many packages and use multiple environments to separate these different versions.

The command-line program conda is both a package manager and an environment manager. This helps data scientists ensure that each version of each package has all the dependencies it requires and works correctly.

Navigator is an easy, point-and-click way to work with packages and environments without needing to type conda commands in a terminal window. You can use it to find the packages you want, install them in an environment, run the packages, and update them – all inside Navigator.

## What applications can we access using Navigator?

The following applications are available by default in Navigator:

* [Jupyter Notebook](https://jupyter.readthedocs.io/en/latest/)
* [Spyder](https://www.spyder-ide.org/)
* [PyCharm](https://www.jetbrains.com/pycharm/documentation/)
* [VSCode](https://code.visualstudio.com/docs)
* [Glueviz](http://glueviz.org/en/stable/)
* [Orange 3 App](http://orange.biolab.si/docs/)
* [RStudio](http://docs.rstudio.com/)
* Anaconda Prompt (Windows only)
* Anaconda PowerShell (Windows only)
* Jupyter Lab
* **JupyterLab:** This is an extensible working environment for interactive and reproducible computing, based on the Jupyter Notebook and Architecture.
* **Qt Console:** It is the PyQt GUI that supports inline figures, proper multiline editing with syntax highlighting, graphical calltips and more.
* **Spyder:** Spyder is a scientific Python Development Environment. It is a powerful Python IDE with advanced editing, interactive testing, debugging and introspection features.
* **VS Code:** It is a streamlined code editor with support for development operations like debugging, task running and version control.
* **Glueviz:** This is used for multidimensional data visualization across files. It explores relationships within and among related datasets.
* **Orange 3:** It is a component-based data mining framework. This can be used for data visualization and data analysis. The workflows in Orange 3 are very interactive and provide a large toolbox.
* **Rstudio:** It is a set of integrated tools designed to help you be more productive with R. It includes R essentials and notebooks.
* **Jupyter Notebook:** This is a web-based, interactive computing notebook environment. We can edit and run human-readable docs while describing the data analysis.

The Jupyter Notebook is an open source web application that you can use to create and share documents that contain live code, equations, visualizations, and text. Jupyter Notebook is maintained by the people at [Project Jupyter.](http://jupyter.org/)

Jupyter Notebooks are a spin-off project from the IPython project, which used to have an IPython Notebook project itself. The name, Jupyter, comes from the core supported programming languages that it supports: Julia, Python, and R. Jupyter ships with the IPython kernel, which allows you to write your programs in Python, but there are currently over 100 other kernels that you can also use.

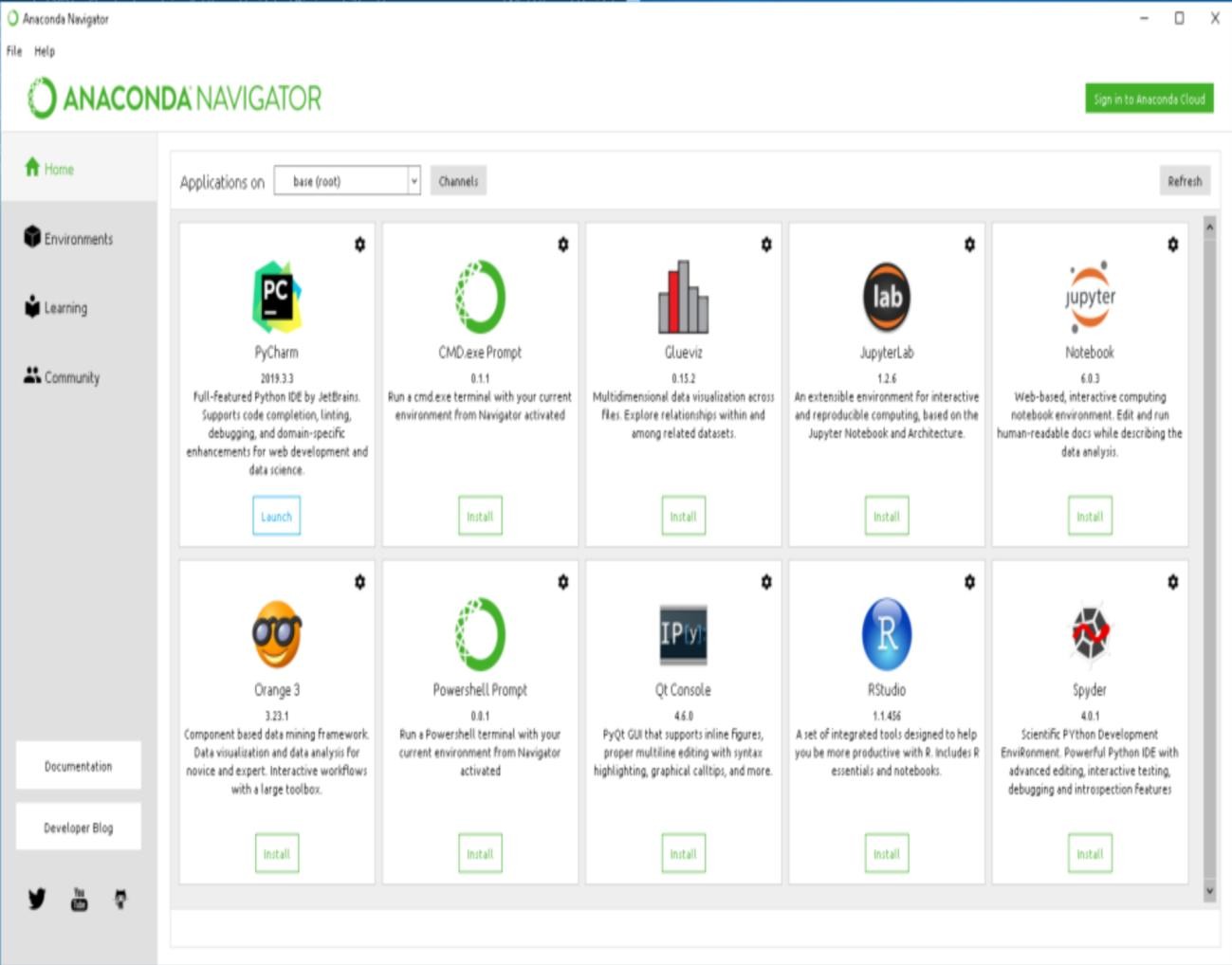
The Jupyter Notebook is not included with Python, so if you want to try it out, you will need to install Jupyter.

There are many distributions of the Python language. This article will focus on just two of them for the purposes of installing Jupyter Notebook. The most popular is CPython, which is the reference version of Python that you can get from their website. It is also assumed that you are using Python.

## Installation on Windows

Visit the link <https://www.python.org/downloads/> to download the latest release of Python. In this process, we will install Python 3.6.7 on our Windows operating system.

Double-click the executable file which is downloaded; the following window will open. Select Customize installation and proceed.



Anaconda Navigator

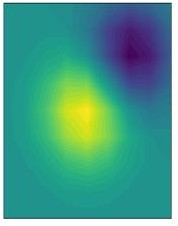
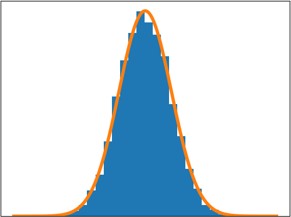
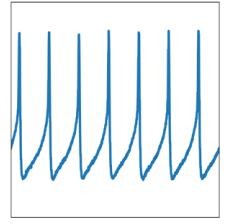
## How can I run code with Navigator?

The simplest way is with Spyder. From the Navigator Home tab, click Spyder, and write and execute your code.

You can also use Jupyter Notebooks the same way. Jupyter Notebooks are an increasingly popular system that combine your code, descriptive text, output, images, and interactive interfaces into a single notebook file that is edited, viewed, and used in a web browser.

## LIBRARIES Matplotlib:

* Matplotlib is a Python 2D plotting library which produces publication quality figures in a variety of hardcopy formats and interactive environments across platforms.
* Matplotlib can be used in Python scripts, the Python and IPython shells, the Jupyter notebook, web application servers, and four graphical user interface toolkits.



[Matplotlib images](https://matplotlib.org/tutorials/introductory/sample_plots.html)

* Matplotlib tries to make easy things easy and hard things possible.
* You can generate plots, histograms, power spectra, bar charts, error charts, scatterplots, etc., with just a few lines of code.
* For simple plotting the pyplot module provides a MATLAB-like interface, particularly when combined with IPython.
* For the power user, you have full control of line styles, font properties, axes properties, etc, via an object oriented interface or via a set of functions familiar to MATLAB users.

## Numpy:

NumPy is the fundamental package for scientific computing with Python. It contains among other things:

* a powerful N-dimensional array object
* sophisticated (broadcasting) functions
* tools for integrating C/C++ and Fortran code
* useful linear algebra, Fourier transform, and random number capabilities
* Besides its obvious scientific uses, NumPy can also be used as an efficient multi- dimensional container of generic data. Arbitrary data-types can be defined. This allows NumPy to seamlessly and speedily integrate with a wide variety of databases.
* NumPy is licensed under the [BSD license,](https://numpy.org/license.html#license) enabling reuse with few restrictions.

## Pandas:

**History of development**

In 2008, **pandas** development began at [AQR Capital Management](https://www.aqr.com/). By the end of 2009 it had been [open sourced,](https://en.wikipedia.org/wiki/Open_source) and is actively supported today by a community of like-minded individuals around the world who contribute their valuable time and energy to help make open source **pandas** possible.

Since 2015, **pandas** is a [NumFOCUS sponsored project.](https://numfocus.org/sponsored-projects) This will help ensure the success of development of **pandas** as a world-class open-source project.

## Timeline

* **2008**: Development of **pandas** started
* **2009**: **pandas** become open source
* **2012**: First edition of **Python for Data Analysis** is published
* **2015**: **pandas** becomes a [NumFOCUS sponsored project](https://numfocus.org/sponsored-projects)
* **2018**: First in-person core developer sprint

## Library Highlights

* A fast and efficient **DataFrame** object for data manipulation with integrated indexing.
* Tools for **reading and writing data** between in-memory data structures and different formats: CSV and text files, Microsoft Excel, SQL databases, and the fast HDF5 format;
* Intelligent **data alignment** and integrated handling of **missing data**: gain automatic label-based alignment in computations and easily manipulate messy data into an orderly form.
* Flexible **reshaping** and pivoting of data sets.
* Intelligent label-based **slicing**, **fancy indexing**, and **subsetting** of large data sets.
* Columns can be inserted and deleted from data structures for **size mutability**.
* Aggregating or transforming data with a powerful **group by** engine allowing split- apply-combine operations on data sets.
* High performance **merging and joining** of data sets.
* **Hierarchical axis indexing** provides an intuitive way of working with high- dimensional data in a lower-dimensional data structure.
* **Time series**-functionality: date range generation and frequency conversion, moving window statistics, date shifting and lagging. Even create domain-specific time offsets and join time series without losing data.
* Highly **optimized for performance**, with critical code paths written in [Cython](http://www.cython.org/) or C.
* Python with **pandas** is in use in a wide variety of **academic and commercial** domains, including Finance, Neuroscience, Economics, Statistics, Advertising, Web Analytics, and more.

## Mission

Pandas aims to be the fundamental high-level building block for doing practical, real world data analysis in Python. Additionally, it has the broader goal of becoming the most powerful and flexible open source data analysis / manipulation tool available in any language.

## Vision

* Accessible to everyone
* Free for users to use and modify
* Flexible
* Powerful
* Easy to use
* Fast

## Values

Is in the core of **pandas** to be respectful and welcoming with everybody, users, contributors and the broader community. Regardless of level of experience, gender, gender identity and expression, sexual orientation, disability, personal appearance, body size, race, ethnicity, age, religion, or nationality.

## Regex:

* A **regular expression**, **regex** or **regexp** (sometimes called a **rational expression**) is a sequence of [characters](https://en.wikipedia.org/wiki/Character_(computing)) that define a *search* [*pattern*](https://en.wikipedia.org/wiki/Pattern_matching).
* Usually such patterns are used by [string searching algorithms](https://en.wikipedia.org/wiki/String_searching_algorithm) for "find" or "find and replace" operations on [strings,](https://en.wikipedia.org/wiki/String_(computer_science)) or for input validation.
* It is a technique developed in [theoretical computer science](https://en.wikipedia.org/wiki/Theoretical_computer_science) and [formal language](https://en.wikipedia.org/wiki/Formal_language) theory.
* Regular expressions are used in [search engines](https://en.wikipedia.org/wiki/Search_engine), search and replace dialogs of [word processors](https://en.wikipedia.org/wiki/Word_processor) and [text editors](https://en.wikipedia.org/wiki/Text_editor), in text processing utilities such as [sed](https://en.wikipedia.org/wiki/Sed) and [AWK](https://en.wikipedia.org/wiki/AWK) and in [lexical analysis.](https://en.wikipedia.org/wiki/Lexical_analysis)
* Many [programming languages](https://en.wikipedia.org/wiki/Programming_language) provide regex capabilities either built-in or via [libraries](https://en.wikipedia.org/wiki/Library_(computing)).

## Requests:

* **Requests** is a [Python](https://en.wikipedia.org/wiki/Python_(programming_language)) HTTP library, released under the [Apache2 License.](https://en.wikipedia.org/wiki/Apache_License)
* The goal of the project is to make HTTP requests simpler and more human-friendly.
* The current version is 2.22.0
* The [requests](http://docs.python-requests.org/en/master/) library is the de facto standard for making HTTP requests in Python.
* It abstracts the complexities of making requests behind a beautiful, simple API so that you can focus on interacting with services and consuming data in your application.

## Scikit-learn:

* **scikit-learn** (formerly **scikits.learn** and also known as **sklearn**) is a [free software](https://en.wikipedia.org/wiki/Free_software) [machine learning](https://en.wikipedia.org/wiki/Machine_learning) [library](https://en.wikipedia.org/wiki/Library_(computing)) for the [Python](https://en.wikipedia.org/wiki/Python_(programming_language)) programming language.
* It features various [classification](https://en.wikipedia.org/wiki/Statistical_classification), [regression](https://en.wikipedia.org/wiki/Regression_analysis) and [clustering](https://en.wikipedia.org/wiki/Cluster_analysis) algorithms including [support vector machines,](https://en.wikipedia.org/wiki/Support_vector_machine) [random forests,](https://en.wikipedia.org/wiki/Random_forests) [gradient boosting](https://en.wikipedia.org/wiki/Gradient_boosting), [*k*-means](https://en.wikipedia.org/wiki/K-means_clustering) and [DBSCAN](https://en.wikipedia.org/wiki/DBSCAN), and is designed to interoperate with the Python numerical and scientific libraries [NumPy](https://en.wikipedia.org/wiki/NumPy) and [SciPy.](https://en.wikipedia.org/wiki/SciPy)
* Scikit-learn is largely written in Python, and uses [numpy](https://en.wikipedia.org/wiki/Numpy) extensively for high- performance linear algebra and array operations.
* Furthermore, some core algorithms are written in [Cython](https://en.wikipedia.org/wiki/Cython) to improve performance.
* Support vector machines are implemented by a Cython wrapper around [LIBSVM](https://en.wikipedia.org/wiki/LIBSVM); logistic regression and linear support vector machines by a similar wrapper around [LIBLINEAR](https://en.wikipedia.org/wiki/LIBLINEAR).
* In such cases, extending these methods with Python may not be possible.
* Scikit-learn integrates well with many other Python libraries, such as matplotlib and plotly for plotting, numpy for array vectorization, pandas dataframes, scipy, and many more.
* Scikit-learn is one of the most popular machine learning libraries on [GitHub](https://en.wikipedia.org/wiki/GitHub).
* NumPy provides some functions for linear algebra, [Fourier transforms](https://en.wikipedia.org/wiki/Fourier_transform), and [random number generation,](https://en.wikipedia.org/wiki/Random_number_generation) but not with the generality of the equivalent functions in SciPy.
* NumPy can also be used as an efficient multidimensional container of data with arbitrary datatypes.
* This allows NumPy to seamlessly and speedily integrate with a wide variety of databases.
* Older versions of SciPy used Numeric as an array type, which is now deprecated in favor of the newer NumPy array code.

**Python Introduction**

**Python** is a general purpose, dynamic, high level and interpreted programming language. It supports Object Oriented programming approach to develop applications. It is simple and easy to learn and provides lots of high-level data structures.

Python is easy to learn yet powerful and versatile scripting language which makes it attractive for Application Development.

Python's syntax and dynamic typing with its interpreted nature, makes it an ideal language for scripting and rapid application development.

Python supports multiple programming pattern, including object oriented, imperative and functional or procedural programming styles.

Python is not intended to work on special area such as web programming. That is why it is known as multipurpose because it can be used with web, enterprise, 3D CAD etc.

We don't need to use data types to declare variable because it is dynamically typed so we can write a=10 to assign an integer value in an integer variable.

Python makes the development and debugging *fast* because there is no compilation step included in python development and edit-test-debug cycle is very fast.

# Python Features

Python provides lots of features that are listed below.

#### 1) Easy to Learn and Use

Python is easy to learn and use. It is developer-friendly and high level programming language.

#### 2) Expressive Language

Python language is more expressive means that it is more understandable and readable.

#### 3) Interpreted Language

Python is an interpreted language i.e. interpreter executes the code line by line at a time. This makes debugging easy and thus suitable for beginners.

#### 4) Cross-platform Language

Python can run equally on different platforms such as Windows, Linux, Unix and Macintosh etc. So, we can say that Python is a portable language.

#### 5) Free and Open Source

Python language is freely available at [offical web address](https://www.python.org/" \t "blank).The source-code is also available. Therefore it is open source.

#### 6) Object-Oriented Language

Python supports object oriented language and concepts of classes and objects come into existence.

#### 7) Extensible

It implies that other languages such as C/C++ can be used to compile the code and thus it can be used further in our python code.

#### 8) Large Standard Library

Python has a large and broad library and prvides rich set of module and functions for rapid application development.

#### 9) GUI Programming Support

Graphical user interfaces can be developed using Python.

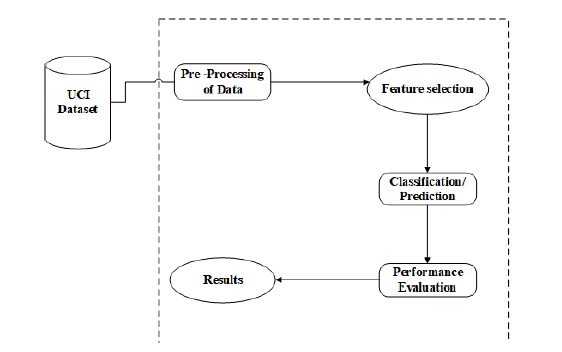
#### 10) Integrated

It can be easily integrated with languages like C, C++, JAVA etc.

# 6.3Python Applications

Python is known for its general purpose nature that makes it applicable in almost each domain of software development. Python as a whole can be used in any sphere of development.

**System Methodology**

****

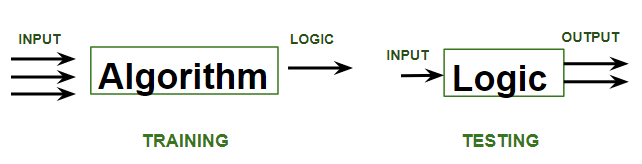
**Algorithms:**

Machine learning is the study of algorithms that improve their performance at same task with experience.

**Types:**

* Supervised learning (training data + desired outputs(labels)).
* Unsupervised learning (training data without desired outputs).
* Semi-supervised learning (training data + a few desired outputs).
* Reinforcement learning (rewards from sequence of actions).

**Supervised Algorithm:**

****Supervised learning is when the model is getting trained on a labelled dataset. Labelled dataset is one which have both input and output parameters. In this type of learning both training and validation datasets are labelled

**Types of Supervised Learning:**

Classification: It is a Supervised Learning task where output is having defined labels (discrete value). For example, in above Figure A, Output Purchased has defined labels i.e., 0 or 1; 1 means the customer will purchase and 0 means that customer won’t purchase. The goal here is to predict discrete values belonging to a particular class and evaluate on the basis of accuracy.

**Source Code**

*#!/usr/bin/env python  
# coding: utf-8  
  
# # House Price Analysis  
  
-info">  
# <b> Training </b> </div>*

*# In[11]:*model\_dtr.fit(x\_train,y\_train)  
  
  
*# <div class="alert alert-block alert-info">  
# <b>Prediction and Evaluation</b> </div>  
  
# In[17]:***from** sklearn.metrics **import** r2\_score  
DT = r2\_score(y\_test, model\_dtr.predict(x\_test))\*100  
  
  
*# <div class="alert alert-block alert-success">  
# <b>Here we got 62% of Accuracy in Decision Tree Regressor</b> </div>  
#   
  
# # ----------------------------------------------------------------------------------------------------------  
  
# <div class="alert alert-block alert-info">  
# <b>LINEAR-REGRESSION</b> </div>  
  
# In[18]:*x1 = df.iloc[:,:-1]  
y1 = df[**'listprice'**]  
  
  
*# In[19]:*

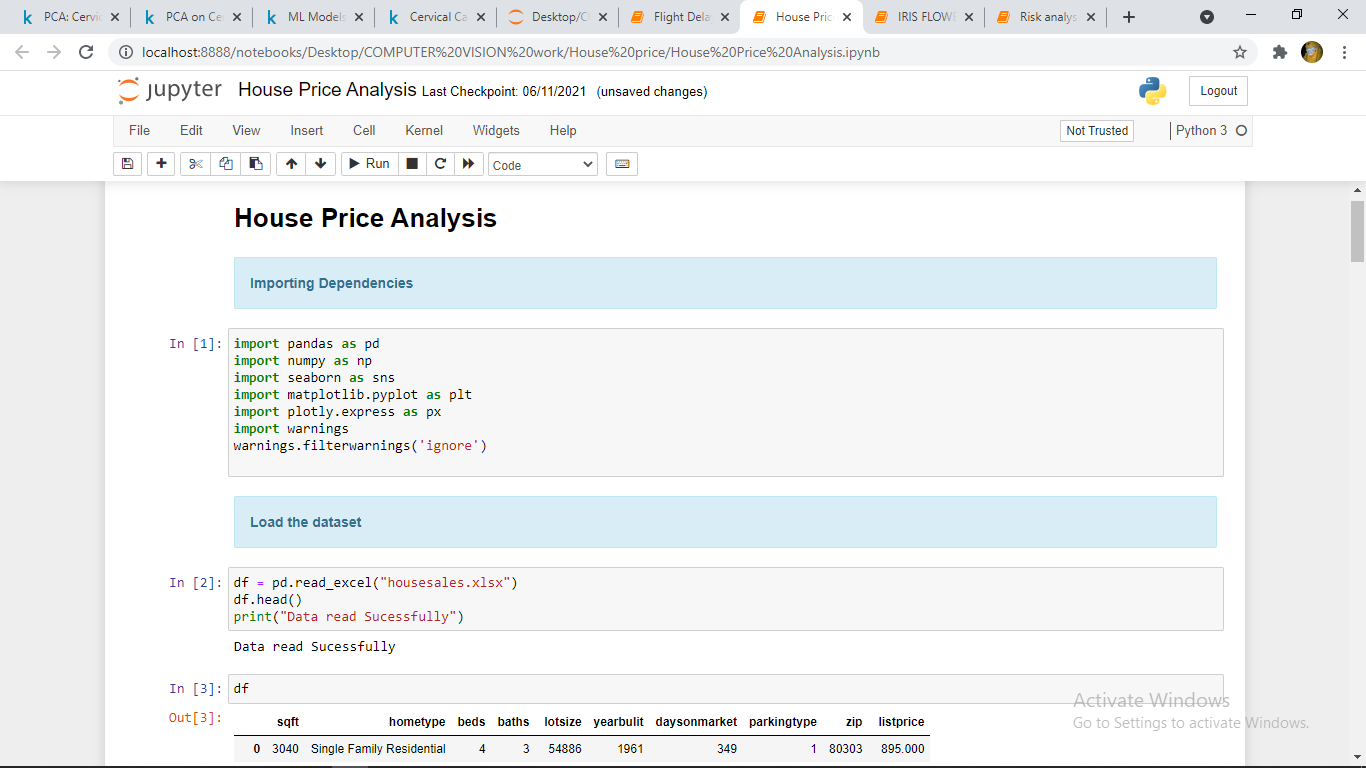
**from** sklearn.model\_selection **import** train\_test\_split  
x1\_train, x1\_test, y1\_train, y1\_test = train\_test\_split(x1,y1,test\_size=0.3,random\_state=10)  
*# In[20]:***from** sklearn.linear\_model **import** LinearRegression  
model\_lr = LinearRegression()  
  
  
*# In[21]:*model\_lr.fit(x1\_train,y1\_train)  
  
  
*# In[22]:***from** sklearn.metrics **import** r2\_score  
LR = r2\_score(y\_test, model\_lr.predict(x\_test))\*100  
  
  
*# <div class="alert alert-block alert-success">  
# <b>Here we got 74% of Accuracy in LINEAR-REGRESSION</b> </div>  
#   
  
# <div class="alert alert-block alert-info">  
# <b>RIDGE REGRESSION</b> </div>  
  
# In[23]:***from** sklearn.linear\_model **import** Ridge  
model\_rdg = Ridge()

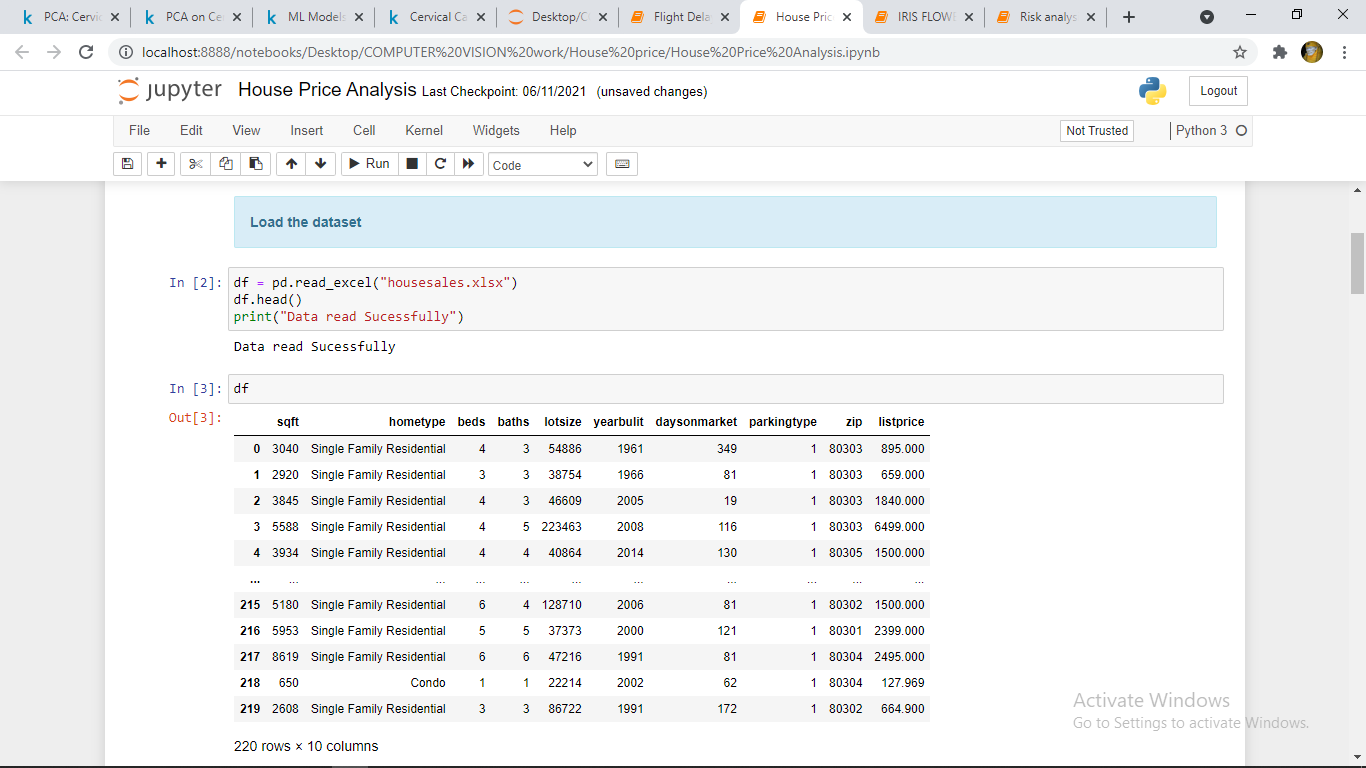
*# In[24]:*model\_rdg.fit(x\_train,y\_train)  
  
  
*# In[25]:***from** sklearn.metrics **import** r2\_score  
RR = r2\_score(y\_test, model\_rdg.predict(x\_test))\*100  
  
*# <div class="alert alert-block alert-success">  
# <b>Here we got 74% of Accuracy in RIDGE-REGRESSION</b> </div>  
#  
# <div class="alert alert-block alert-info">  
# <b>LASSO REGRESSION</b> </div>  
  
# In[26]:  
  
# <div class="alert alert-block alert-info">  
# <b>Importing Dependencies</b> </div>  
  
# In[1]:***import** pandas **as** pd  
**import** numpy **as** np  
**import** seaborn **as** sns  
**import** matplotlib.pyplot **as** plt  
**import** warnings  
warnings.filterwarnings(**'ignore'**)  
  
*# <div class="alert alert-block alert-info">  
# <b>Load the dataset</b> </div>  
  
# In[2]:*df = pd.read\_excel(**"housesales.xlsx"**)  
df.head()  
**print**(**"Data read Sucessfully"**)  
  
  
*# In[3]:*df  
  
  
*# <div class="alert alert-block alert-info">  
# <b>To display columns,datatype, Null values </b> </div>  
  
# In[4]:*df.info()  
  
  
*# In[5]:*df.shape  
  
  
*# In[6]:*

df.describe()  
  
  
*# <div class="alert alert-block alert-info">  
# <b>To display columns Number of unique values </b> </div>  
  
# In[7]:*df.nunique()  
  
  
*# In[8]:*df[**'zip'**].nunique()  
  
  
*# <div class="alert alert-block alert-info">  
# <b>To display columns missing values </b> </div>  
  
# In[9]:  
  
  
# Missing Values identification*missing\_val\_count\_by\_column = (df.isnull().sum())  
missing\_val\_count\_by\_column  
*# In[10]:*df.head()  
*# <div class="alert alert-block alert-info">  
# <b>LABEL ENCODING --> </b> Converting Categorical variables to numerical</div>*

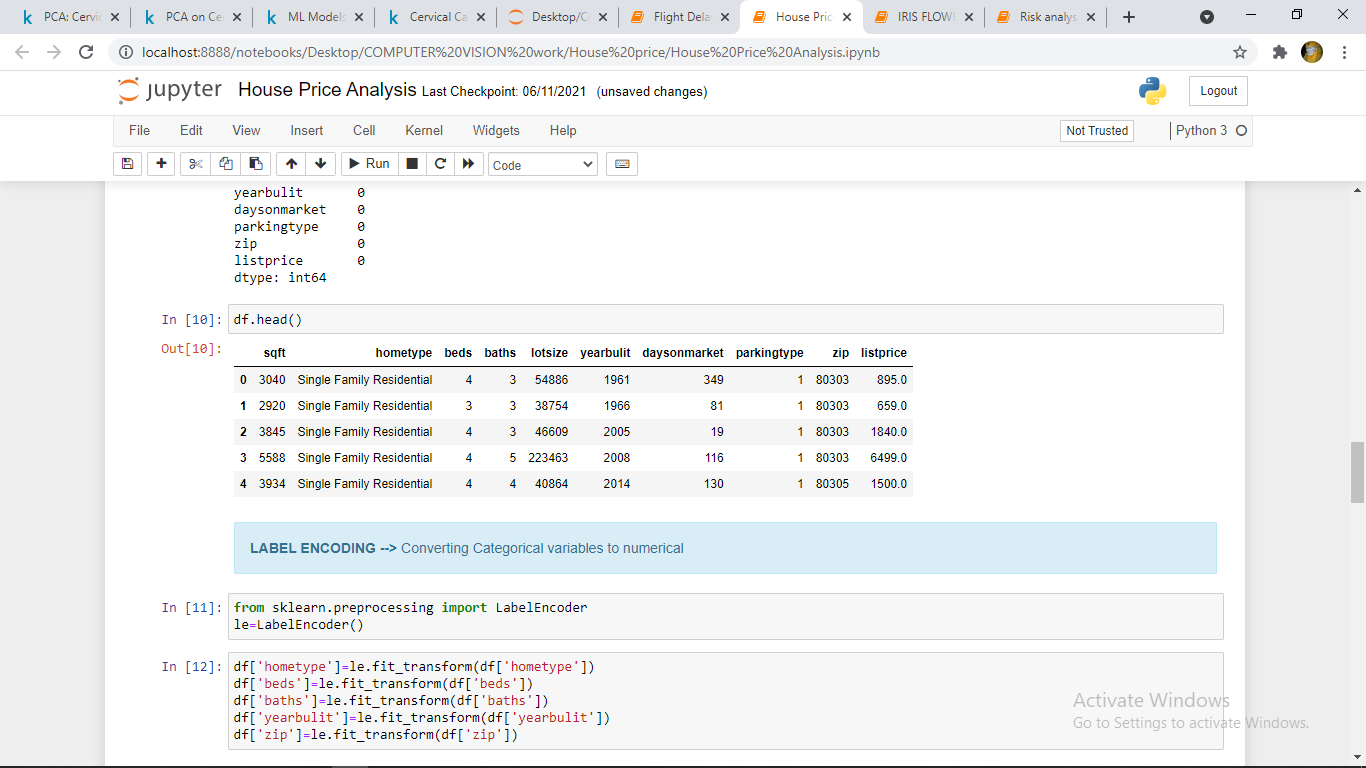
**Implementation and Result Screen shots**

**Dataset Used:**

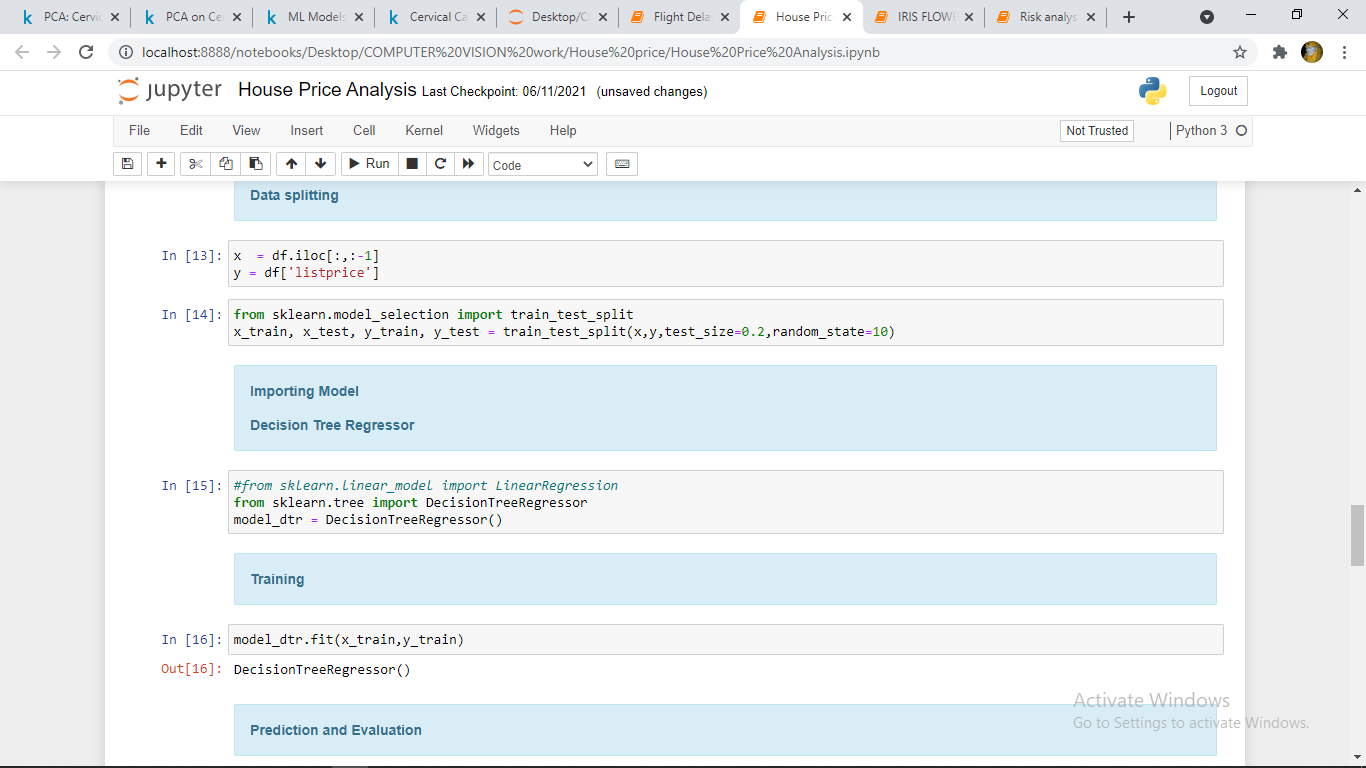


In this Screenshot I have import python libraries. After that I have load the “House price analysis” Dataset.

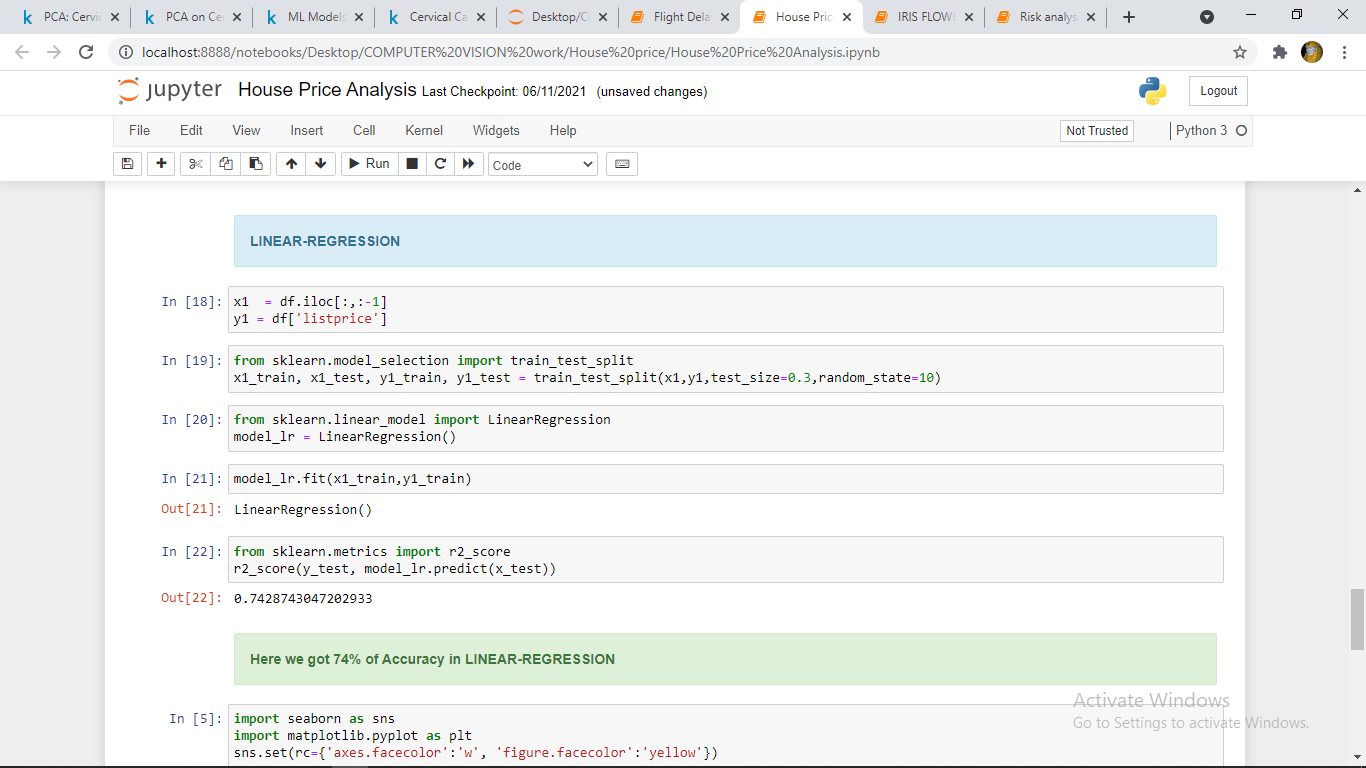
After loading the dataset, we have to do ‘’Exploratory Data Analysis” (EDA) for describing every variable data and analyse the data.



In screenshot we have done Data pre-processing “Label encoding” (Converting Categorical variables data to numerical data).

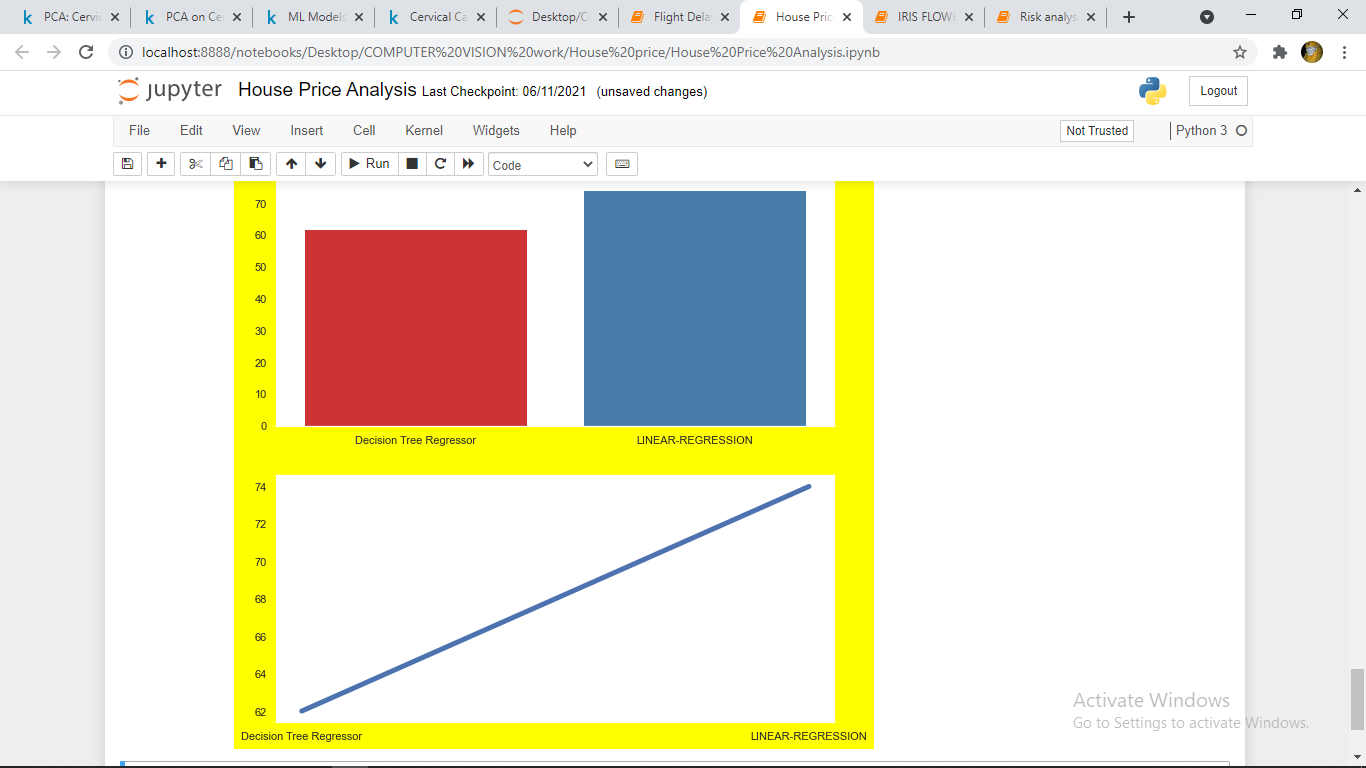


After completion of Data pre-processing, we have done Data splitting into xtrain and ytrain. Then we import the Machine learning model and fit into our data.



In this project we use two models those are Linear regression we got 74% accuracy, Decision tree regressor we got 62% accuracy.

And below are the Visualizations of accuracy comparisons.



# .

**CONCLUSION**

This Project mainly concentrates on the comparison between different machine learning algorithms (Multiple Linear Regression, Decision Tree Regressor) about House price prediction Analysis. From the above experiment results, linear algorithm has high accuracy value when compared to all the other algorithms regarding house price predictions. Here the [MSE] Mean Square Error and [RMSE] Root Mean Square Error are used in order to calculate the accuracy value of the algorithm on the King County Dataset which was collected from public data set.The paper can be extended by applying the above said algorithms to predict House resale value.

**Biblography**

1. Aminah Md Yusof and Syuhaida Ismail, Multiple Regressions in Analyzing House Price Variations. IBIMA Publishing Communications of the IBIMA Vol. 2012 (2012), Article ID383101, 9 pages DOI: 10.5171/2012.383101.
2. Babyak, M. A. What you see may not be what you get: A brief, non-technical introduction to over fitting regression-type models. Psychosomatic Medicine, 66(3), 411-4.
3. Atharva chogle, Priyanka khaire, Akshata gaud, Jinal Jain. House Price Forecasting using Data Mining Techniques International Journal of Advanced Research in Computer and communication Engineering ISO 3297:2007 Certified Vol. 6, Issue 12, December 2017
4. Darshan Sangani, Kelby Erickson, and Mohammad Al Hasan, Predicting Zillow Estimation Error Using Linear Regression and Gradient Boosting, IEEE 14th International Conference on Mobile Ad Hoc and Sensor Systems (MASS), Page(s):530 – 534
5. Azme Bin Khamis, Nur Khalidah Khalilah Binti Kamarudin Comparative Study on Estimate House Price Using Statistical and Neural Network, International journal of scientific and technology, research volume 3, ISSUE 12, December 2014, Page(s):126-131.
6. Adyan Nur Alfiyatin, Hilman Taufiq, Ruth Ema Febrita, WayanFirdaus Mahmudy, Modeling House Price Prediction using Regression Analysis and Particle Swarm Optimization Case Study: Malang, East Java, Indonesia. (IJACSA) International Journal of Advanced Computer Science and Applications, Vol. 8, No. 10,2017, Page(s):323-326