

PROFESSIONAL TRAINING REPORT
AT
Sathyabama Institute of Science and Technology
(Deemed to be University)

Submitted in partial fulfillment of the requirements for the award
of Bachelor of Technology Degree in

Information Technology

By

NAME :KEERTHIVASAN

E(REG NO: 37120040)



DEPARTMENT OF INFORMATION TECHNOLOGY
SCHOOL OF COMPUTING

SATHYABAMA
INSTITUTE OF SCIENCE AND
TECHNOLOGY (DEEMED TO BE
UNIVERSITY)
Accredited with Grade “A” by NAAC
JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI -
600 119

AUGUST 2019



SATHYABAMA
INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)
Accredited with "A" Grade by NAAC
Jeppiaar Nagar, Rajiv Gandhi Salai, Chennai - 600 119.
Phone: 044 - 2450 3150 / 51 / 52 / 54 / 55 Fax: 044 - 2450 2344
www.sathyabama.ac.in



SCHOOL OF COMPUTING

BONAFIDE CERTIFICATE

This is to certify that this Professional Training Report is the bonafide work of KEERTHIVASAN E (Reg. No. 37120040.) who carried out the project entitled "**Housing Price Prediction using machine learning**" under our supervision from Jul 2020 to August 2020.

Internal Guide

Dr. MARYGLADENCE L., M.E., Ph.D.,

Head of the Department

Dr. R. SUBHASHINI, M.E., Ph.D.,

Submitted for Viva voice Examination held on _____

Internal Examiner

External Examiner

DECLARATION

I am E.KEERTHIVASAN hereby declare that the mini project report entitled on “**Housing Price Prediction using machine learning approach**” done by me under the guidance of Dr.Marygladence L.,ME.,Ph.D., at Sathyabama Institute of Science And Technology, Semmancheri, Chennai – 600 119 and Mr. Sathish, Senior Embedded Engineer, LCS Controls PVT. LTD. is submitted in partial fulfillment of the requirements for the award of Bachelor of Technology degree in Information Technology.

DATE :

PLACE:

Signature of candidate

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to **Board of Management of SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY** for their kind

encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T.Sasikala M.E., Ph.D**, Dean, School of Computing , **Dr.SUBHASHINI M.E., Ph.D.** , Head of the Department of Information and Technology for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Dr.L.Marygladence M.E.,Ph.D.**,for his valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **Department of Information Technology** who were helpful in many ways for the completion of the project.

Google Analytics Academy

Google Analytics for Beginners

Certificate of Completion

Keerthivasan E

Awarded for successfully completing
the course "Google Analytics for
Beginners"



Certificate expires January 19, 2024

TABLE OF CONTENTS

CHAPTER NO	TITLE	PAGE NO
1	Abstract	1
1.1	Existing Process	1
1.2	Drawbacks	2
1.3	Proposal System	2
1.4	Data Wrangling	2
1.5	Data collection	3
1.6	Preprocessing	3
1.7	Building classification model	3
2	Construction of predictive model	4
3	Environmental Requirements	4
3.1	Software requirements	4
3.2	Hardware requirements	5
4	Introduction	6
4.1	Domain overview	7
4.2	Preparing Dataset	7
4.3	Overview of the system	8
5	Objectives	9
6	Scope	9
7	Literature Survey	9
7.1	General	10
7.2	Review of Literature survey	10
8	Exploratory Data Analysis	11
8.1	Data wrangling	12
8.2	Data collection	13
8.3	Preprocessing	13
8.4	Building classification model	14
8.5	constructing primitive model	14
8.6	Training the dataset	15
8.7	Testing the dataset	16
8.8	General properties	11
8.9	Project goals	12
9	List of modules	13
9.1	Variable identification process	14
9.2	Outlier detection process	15

10	Comparing algorithm with prediction	15
11	GUI Application	16
11.1	Algorithm Explanation	17
11.2	Logistic Regression	18
11.3	Decision Tree	18
12	K-Nearest Neighbor(KNN)	18
12.1	Random forest	18
12.2	Support vector machines	19
13	Project requirements	19
13.1	General	20
14	Functional requirements	20
14.1	Non- functional requirements	21
15	Design Engineering	22
15.1	General	23
15.2	Software description	24
15.3	Anaconda Navigator	24
15.4	Working process	24
16	System Architecture	25
17	Screen shots	25
18	Conclusion	26
19	Future work	27

LIST OF FIGURES

FIGURE NO	FIGURE NAME	PAGE NO
1	Process of machine learning	4
2	Machine learning model	5
3	Process of data flow diagram	11
4	Architecture of proposal model	20
5	Workflow diagram	25
6	Jupyter notebook platform	26

GUI based Housing Price Prediction using machine learning approach

Abstract:

House prices increase every year, it is a need for a system to predict house prices in the future. House price prediction can help the developer determine the selling price of a house and can help the customer to arrange the right time to purchase a house. To prevent this problem in influence the price of a house which include physical conditions, concept and location.. The aim is to investigate machine learning based techniques for Housing Price Prediction forecasting by prediction results in best accuracy. The analysis of dataset by supervised machine learning technique(SMLT) to capture several information's like, variable identification, uni-variate analysis, bi-variate and multi-variate analysis, missing value treatments and analyze the data validation, data cleaning/preparing and data visualization will be done on the entire given dataset. Our analysis provides a comprehensive guide to sensitivity analysis of model parameters with regard to performance in prediction of air quality pollution by accuracy calculation. To propose a machine learning-based method to accurately predict the house price prediction results in the form of best accuracy from comparing supervise classification machine learning algorithms. Additionally, to compare and discuss the performance of various machine learning algorithms GUI based user interface house price prediction by attributes.

Keywords: dataset, Machine learning-Classification method, python, GUI results.

Existing Process

It is a privacy-preserving online EO recommendation system with big data support for ES in smart home. It model the problem as a CMAB problem. It provided detailed theoretical analysis and evaluate the performance on real big datasets. It proposal can achieve the following design objectives: protection of users' privacy, protection of utility provider's privacy, ES, accurate predictions of users' behavior and activities and users' satisfaction. it proposed context-aware online learning algorithm could handle both the coarse-grained and fine-grained human activity as context adaptively. And, it is capable to enlarge the recommended EO acceptance rate, the task completion degree and to save more energy. Furthermore, the system adaptability is improved with another 13.6% in peak power reduction and about 37.4% ES performance with an additional ES of 10.1% comparing to state-of-the-art methods. it indicates that PRCOES could enhance users' experience and prolong users' engagement in everyday ES while guarantee the privacy for both residents and utility provider. Energy-saving (ES) systems developed on the basis of the Internet-of-Things (IoT) by heavily relying on automated understanding of human behaviors and activities recognition is of paramount importance in smart home. However, classic approaches are incapable to understand the relations among users' contexts and ES of appliances very well, and they cannot handle massive metering and time-varying user context datasets. Moreover, privacy concern is thoroughly aroused from both the residential and utility provider sides as to its essentiality. To tackle these problems, it is a Privacy-preserving and Residential Context-aware Online ES (PRCOES) system in an IoT-enabled smart home environment. it model the repeated interaction of ES of appliances and the activity recognition of user context as a contextual multi-armed bandits problem, where the context-aware online learning algorithm can predict appropriate energy offers (EOs) that could meet the users' satisfaction, task completion rate and ES purposes for appliances. it utilize a tree-based structure expanding from top to bottom to recommend EOs, which supports ever-increasing big metering datasets with user context-awareness. Theoretical analysis shows that it proposal achieves sublinear regret and differential privacy for both residents and utility provider. Experiments results validate that

PRCOES could enhance users' experience and prolong users' engagement in everyday ES while guarantee the privacy for both residents and utility provider

Drawbacks

- Its real estate value is felt critical to help the choices in urban arranging. The land framework is a precarious stochastic process
- It can't thereby better determine the regularity of house price prediction data and achieve more accurate prediction results, like precision, recall, F1 score

Proposed System

This helps all others department to carried out other formalities. It have to find Accuracy of the training dataset, Accuracy of the testing dataset, Specification, False Positive rate, precision and recall by comparing algorithm using python code. The following Involvement steps are,

- Define a problem
- Preparing data
- Evaluating algorithms
- Improving results
- Predicting results

Data Wrangling

In this section of the report will load in the data, check for cleanliness, and then trim and clean given dataset for analysis. Make sure that the document steps carefully and justify for cleaning decisions.

Data collection

The data set collected for predicting given data is split into Training set and Test set. Generally, 7:3 ratios are applied to split the Training set and Test set. The Data Model which was created using Random Forest, logistic, Decision tree algorithms, K-Nearest Neighbor (KNN) and Support vector classifier (SVC) are applied on the Training set and based on the test result accuracy, Test set prediction is done.

Preprocessing

The data which was collected might contain missing values that may lead to inconsistency. To gain better results data need to be preprocessed so as to improve the efficiency of the algorithm. The outliers have to be removed and also variable conversion need to be done.

Building the classification model

The predicting the air quality problem, decision tree algorithm prediction model is effective because of the following reasons: It provides better results in classification problem.

- It is strong in preprocessing outliers, irrelevant variables, and a mix of continuous, categorical and discrete variables.
- It produces out of bag estimate error which has proven to be unbiased in many tests and it is relatively easy to tune with.

Construction of a Predictive Model

Machine learning needs data gathering have lot of past data's. Data gathering have sufficient historical data and raw data. Before data pre-processing, raw data can't be used directly. It's used to preprocess then, what kind of algorithm with model. Training and testing this model working and predicting correctly with minimum errors. Tuned model involved by tuned time to time with improving the accuracy.

Environmental Requirements:

1. Software Requirements:

Operating System	: Windows
Tool	: Anaconda with Jupyter Notebook

2. Hardware requirements:

Processor	: Pentium IV/III
Hard disk	: minimum 80 GB
RAM	: minimum 2 GB

Frontend: GUI Application

Backend: Anaconda tool

Language:Python

INTRODUCTION

Domain overview

Machine learning is to predict the future from past data. Machine learning (ML) is a type of artificial intelligence (AI) that provides computers with the ability to learn without being explicitly programmed. Machine learning focuses on the development of Computer Programs that can change when exposed to new data and the basics of Machine Learning, implementation of a simple machine learning algorithm using python. Process of training and prediction involves use of specialized algorithms. It feed the training data to an algorithm, and the algorithm uses this training data to give predictions on a new test data. Machine learning can be roughly separated in to three categories. There are supervised learning, unsupervised learning and reinforcement learning. Supervised learning program is both given the input data and the corresponding labeling to learn data has to be labeled by a human being beforehand. Unsupervised learning is no labels. It provided to the learning algorithm. This algorithm has to figure out the clustering of the input data. Finally, Reinforcement learning dynamically interacts

with its environment and it receives positive or negative feedback to improve its performance.

Data scientists use many different kinds of machine learning algorithms to discover patterns in python that lead to actionable insights. At a high level, these different algorithms can be classified into two groups based on the way they “learn” about data to make predictions: supervised and unsupervised learning. Classification is the process of predicting the class of given data points. Classes are sometimes called as targets/ labels or categories. Classification predictive modeling is the task of approximating a mapping function from input variables(X) to discrete output variables(y). In machine learning and statistics, classification is a supervised learning approach in which the computer program learns from the data input given to it and then uses this learning to classify new observation. This data set may simply be bi-class (like identifying whether the person is male or female or that the mail is spam or non-spam) or it may be multi-class too. Some examples of classification problems are: speech recognition, handwriting recognition, bio metric identification, document classification etc.



Fig: Process of Machine learning

Supervised Machine Learning is the majority of practical machine learning uses supervised learning. Supervised learning is where have input variables (X) and an output variable (y) and use an algorithm to learn the mapping function from the input to the output is $y = f(X)$. The goal is to approximate the mapping function so well that when you have new input data (X) that you can predict the output variables (y) for that data. Techniques of Supervised Machine Learning algorithms include logistic regression, multi-class classification, Decision Trees and support vector machines etc. Supervised learning requires that the data used to train the algorithm is already labeled with correct answers. Supervised learning problems can be further grouped into Classification problems. This problem has as goal the construction of a succinct model that can predict the value of the dependent attribute from the attribute variables. The difference between the two tasks is the fact that the dependent attribute is numerical for regression and categorical for classification. A classification model attempts to draw some conclusion from observed values. Given one or more inputs a classification model will try to predict the value of one or more outcomes. A classification problem is when the output variable is a category, such as “red” or “blue”.

Preparing the Dataset

The dataset is now supplied to machine learning model on the basis of this data set the model is trained. Every new applicant details filled at the time of application form acts as a test data set. After the operation of testing, model predicts whether the house price is increased or not based upon the inference it conclude on the basis of the training data sets.

Overview of the system

This helps all others department to carried out other formalities. It have to find Accuracy of the training dataset, Accuracy of the testing dataset, Specification, False Positive rate, precision and recall by comparing algorithm using python code. The following Involvement steps are,

- Define a problem
- Preparing data
- Evaluating algorithms
- Improving results
- Predicting results

The steps involved in Building the data model is depicted below.

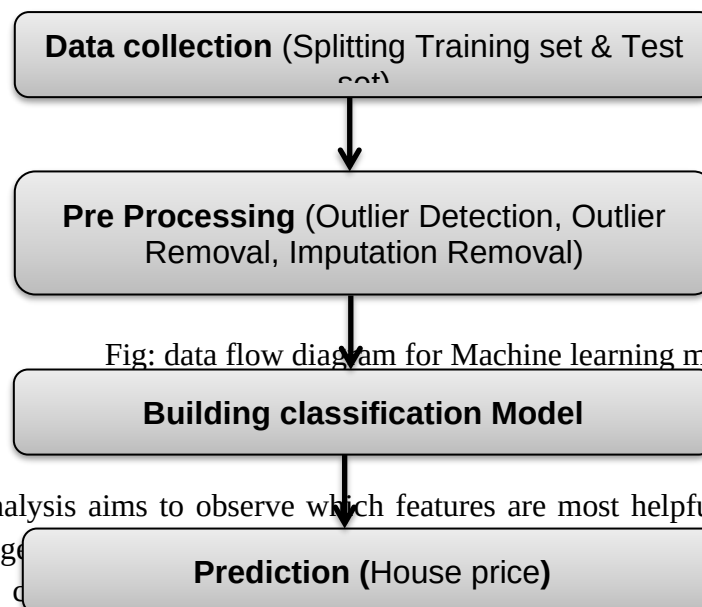


Fig: data flow diagram for Machine learning model

Objectives

This analysis aims to observe which features are most helpful in predicting house prize and to see the good features and bad features. The goal is to do hyper parameter selection. To achieve used machine learning classification methods to fit a function that can predict the discrete class of new input.

The repository is a learning exercise to:

- Apply the fundamental concepts of machine learning from an available dataset and Evaluate and interpret my results and justify my interpretation based on observed dataset.
- Create notebooks that serve as computational records and document my thought process and investigate applications of statistics for house pri to analyses the data set.
- Evaluate and analyses statistical and visualized results, which find the standard patterns for all regiments.

Scope:

The scope of this project is to investigate a dataset of real estate records for banking sector using machine learning technique. To identifying fraud behavior person is more difficult. We try to reduce this risk factor behind selecting the safe person and assets and to predict.

LITERATURE SURVEY

General

A literature review is a body of text that aims to review the critical points of current knowledge on and/or methodological approaches to a particular topic. It is secondary sources and discuss published information in a particular subject area and sometimes information in a particular subject area within a certain time period. Its ultimate goal is to bring the reader up to date with current literature on a topic and forms the basis for another goal, such as future research that may be needed in the area and precedes a research proposal and may be just a simple summary of sources. Usually, it has an organizational pattern and combines both summary and synthesis.

A summary is a recap of important information about the source, but a synthesis is a re-organization, reshuffling of information. It might give a new interpretation of old material or combine new with old interpretations or it might trace the intellectual progression of the field, including major debates. Depending on the situation, the literature review may evaluate the sources and advise the reader on the most pertinent or relevant of them. House price default trends have been long studied from a socio-economic stand point. Most economics surveys believe in empirical modeling of these complex systems in order to be able to predict the house price default rate for a particular individual. The use of machine learning for such tasks is a trend which it is observing now.

Review of Literature Survey

Title : A Comparison Study on Stochastic Modeling Methods for Home Energy Management System

Author: M. Yousefi, M. Soltani and A. Hajizadeh, S

Year : 2019

It several techniques were studied for modeling the different components of the smart home, which was proposed by the research community. The methods were implemented on the Matlab simulation and compared with the real measured data from a single-family home in Esbjerg and Data-port website. The PV model comparison result presented that model-based methods especially Sandia model have a better performance and is more accurate than other methods. It realized that the PV array performance model could be generalized for every system by knowing the specification of their system, while trained ANN and other black box models are unique just for a specific system. By analyzing the results, it is noticed that the ANN and ANFIS give some un-relevant outputs. Obtaining an appropriate model is very crucial to develop an efficient

energy management system (EMS) for the smart home including, Photovoltaic array (PV), Plug-in Electric Vehicle (PEV), home loads and Heat Pump (HP). Stochastic modeling methods of smart home explain random parameters and uncertainties of the above components. In this paper, a concise yet comprehensive analysis and comparison are presented for these techniques. First, modeling methods are implemented to find appropriate and precise forecasting models for PV, PEV, HP and home load demand. Then, the accuracy of each model is validated by the real measured data. Finally, the pros and cons of each method are discussed and reviewed. The obtained results show the conditions, under which the methods can provide a reliable and accurate description of smart home dynamics.

Title : A Heuristic-Based Appliance Scheduling Scheme for Smart Homes

Author: Anish Jindal, Bharat Singh Bhambhu, Mukesh Singh, Neeraj Kumar, and Kshirasagar Naik

Year : 2019

With the advancements in SHs, the load fluctuations in grids can be reduced so that the stability of the grid is maintained. In order to do so, the load demand of SHs needs to comply with the power supplied by the utility. In this paper, we have considered the use of PV panels and BESS in addition to the power supplied by the grid, to manage the load demands of SHs. Unlike other approaches, the proposed scheme handles the power requirement of SHs by scheduling the appliances while complying with the power supplied by the utility. For this purpose, the power supplied from the SG is distributed amongst the SHs based on their power requirements. The appliances are then scheduled according to the available power and user's priority. The interruptible load is shifted to the time instances having less power requirement. In addition to it, instantaneous load change is also catered by rescheduling the appliances of less preference in other time-slots. Moreover, the excess energy available after the scheduling of appliances is stored in the BESS. Simulation results show the effectiveness of the proposed scheme to handle the load requirements of SHs with respect to the power supplied by the utility. The ever-growing demand for electricity in the residential sector results in creating a severe burden on electric grids. However, with the emergence of smart homes (SHs) and smart grids (SGs), this burden can be reduced to some extent. To address this issue, we propose an energy management system in this paper which manages the power requirements of SHs automatically according to the utility constraints and user priorities. The proposed system is based on a heuristic technique which considers the user's priority and power available from the grid as well as distributed energy resources (DERs) for scheduling of appliances. It works by dividing the appliance scheduling problem in an SH into sub-problems for different time-slots. Then, a heuristic solution is designed for each sub-problem. The instantaneous load demands are handled in real-time to comply with the available power from the grid/utility. The data from different SHs is gathered to test the performance of the proposed scheme in real-time. Results show that the proposed scheme efficiently manages the load demand of the SH with respect to power available from the utility, battery energy storage system, and user preferences.

Title : Predicting the Housing Price Direction using Machine Learning Techniques

Author: Debanjan Banerjee and Suchibrota Dutta

Year : 2017

From our experimental results we can surely conclude that Random Forest provides more accuracy however at the same time this particular type of classifier also prone to over fitting therefore the performance of Support Vector Machine classifier can said to be reliable and consistent over the rest of the two classifiers. The phenomenon of the falling or rising of the

house prices has attracted interest from the researcher as well as many other interested parties. There have been many previous research works that used various regression techniques to address the question of the changes house price. This work considers the issue of changing house price as a classification problem and applies machine learning techniques to predict whether house prices will rise or fall. This work applies various feature selection techniques such as variance influence factor, Information value, principle component analysis and data transformation techniques such as outlier and missing value treatment as well as box-cox transformation techniques. The performance of the machine learning techniques is measured by the four parameters of accuracy, precision, specificity and sensitivity. The work considers two discrete values 0 and 1 as respective classes. If the value of the class is 0 then we consider that the price of the house has decreased and if the value of the class is 1 then we consider that the price of the house has increased.

Title : Demand Response for Home Energy Management using Reinforcement Learning and Artificial Neural Network

Author: Renzhi Lu, Seung Ho Hong and Mengmeng Yu

Year : 2019

To proposed an hour-ahead DR algorithm for HEMS, taking an AI approach with the aim of minimizing the user energy bill and degree of discomfort. Specifically, to overcome future price uncertainties, a steady price forecasting model based on ANN is presented. In cooperation with the forecasted prices, multi-agent RL is adopted to make optimal decisions for different appliances. Simulation results showed that the proposed DR algorithm can handle the energy management and minimize the user energy bill and dissatisfaction costs. Furthermore, the electricity costs of two different cases without and with DR are compared, indicating that the DR algorithm in this work can help the user to reduce its energy cost, significantly. Ever-changing variables in the electricity market require energy management systems to make optimal real-time decisions adaptively. Demand response is the latest approach being used to accelerate the efficiency and stability of power systems. This paper proposes an hour-ahead demand response algorithm for home energy management systems. To deal with the uncertainty in future prices, a steady price prediction model based on artificial neural network is presented. In cooperation with forecasted future prices, multi-agent reinforcement learning is adopted to make optimal decisions for different home appliances in a decentralized manner. To verify the performance of the proposed energy management scheme, simulations are conducted with non-shiftable, shiftable, and controllable loads. Experimental results demonstrate that the proposed demand response algorithm can handle energy management for multiple appliances, minimize user energy bills and dissatisfaction costs, and help the user to significantly reduce its electricity cost compared with a benchmark without demand response.

Title : An intelligent fuzzy decision support system for flexible adjustment of dye pricing to manage customer-supplier relationship

Author: Jason C.H. Lee, K.L. Choy, K.H. Leung

Year : 2018

Integrating light-weight IT solutions is a sound and effective mean to enhance a business, or a decision-making process of an enterprise engaged in a traditional industry. The introduction of light-weight IT solution can enable industry practitioners, such as the dye house in the case study of this paper, to better govern the consumer purchasing behavior throughout the dye service provision in a cost-effective manner. In view of a lack of decision support in the field of pricing strategy in the dye industry, this paper proposes a decision support system, which embeds

a fuzzy logic technique to assist the top management of a dye house in making a wise decision in offering a proper discount based on customer behavior. In turn, the discount offered to each customer is dynamic and unique. The results of the case study have shown positive indications that the system creates a win-win relationship between the dye house and the customer, as customers, in the long run, have been paying more respect to the rules of the game in dye service subscription, and on the other hand the dye house is able to deliver the end product to customer with a better timing accuracy, as a result of having a better forecast and schedule the dye production. Proper dye pricing strategy is one of the essential factors that determine the success of a dye business. However, dye pricing decision is a difficult one to make due to the complicated business scenarios that dye practitioners often face. In case of the customers failed to commit the full amount of their pre-booked dye service, in the perspective of a dye house, the dye practitioner would end up losing the profit margin due to the excess dye machinery capacity remained unallocated to a customer for production. To avoid such problematic issues to happen repeatedly and periodically, dye practitioners often 'penalize' the customers who failed to fully commit the booking by, either reducing the discount that was originally provided for the customer at first, or delaying the delivery of the end products. Although the top management who make the final pricing decision are experienced in deciding and applying a proper 'penalty', the problem of inconsistency during the decision-making process with the existence of diverse factors affecting the dye service pricing has resulted in the top management spending large efforts as well as amount of time to make the final decision. In view of improving the quality of such decision and alleviate the pressure of the top management, this paper presents an intelligent system, integrating database management and fuzzy logic technique, for providing decision support in adjusting the dyeing price. Results upon conducting a case study in a dye house located in the Pearlrivier delta region of the mainland China indicate that the proposed solution outperforms the manual decision-making process.

EXPLORATORY DATA ANALYSIS OF HOUSE PRICE PREDICTION

This analysis is not meant to be providing a final conclusion on the reasons leading to real estate house price approval as it doesn't involve using any inferential statistics techniques/machine learning algorithms. Machine learning supervised classification algorithms will be used to give the house dataset and extract patterns, which would help in predicting the likely defaulters/ Not-Fraudulent, thereby helping the banks for making better decisions in the future. Multiple datasets from different sources would be combined to form a generalized dataset, and then different machine learning algorithms would be applied to extract patterns and to obtain results with maximum accuracy.

Data Wrangling

In this section of the report will load in the data, check for cleanliness, and then trim and clean given dataset for analysis. Make sure that the document steps carefully and justify for cleaning decisions.

Data collection

The data set collected for predicting the house price is split into Training set and Test set. Generally, 7:3 ratios are applied to split the Training set and Test set. The Data Model which was created using Random Forest, logistic, Decision tree algorithms, K-Nearest Neighbor (KNN) and Support vector classifier (SVC) are applied on the Training set and based on the test result accuracy, Test set prediction is done.

Preprocessing

The data which was collected might contain missing values that may lead to inconsistency. To gain better results data need to be preprocessed so as to improve the efficiency of the algorithm. The outliers have to be removed and also variable conversion need to be done. Based on the correlation among attributes it was observed that attributes that are significant individually include property area, sqr-feet, distance which is the strongest among all. Some variables such as applicant income and co- applicant income are not significant alone, which is strange since by intuition it is considered as important.

The correlation among attributes can be identified using plot diagram in data visualization process. Data preprocessing is the most time consuming phase of a data mining process. Data integration, data reduction and data transformation are also to be applicable for real estate data. For easy analysis, the data is reduced to some minimum amount of records. Initially the Attributes which are critical to make a real estate credibility prediction is identified with information gain as the attribute-evaluator and Ranker as the search-method.

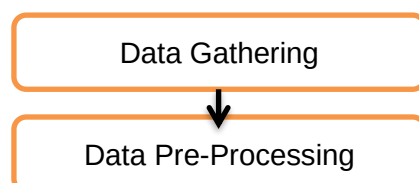
Building the classification model

The predicting the house price problem, decision tree algorithm prediction model is effective because of the following reasons: It provides better results in classification problem.

- It is strong in preprocessing outliers, irrelevant variables, and a mix of continuous, categorical and discrete variables.
- It produces out of bag estimate error which has proven to be unbiased in many tests and it is relatively easy to tune with.

Construction of a Predictive Model

Machine learning needs data gathering have lot of past data's. Data gathering have sufficient historical data and raw data. Before data pre-processing, raw data can't be used directly. It's used to preprocess then, what kind of algorithm with model. Training and testing this model working and predicting correctly with minimum errors. Tuned model involved by tuned time to time with improving the accuracy. Finally, once model is ready, deployed and model to do the predictions and the aims and objectives due to the inconsistency in historical data on bank accountant therefore perform an analysis of the given dataset and describe how to repair it automatically.



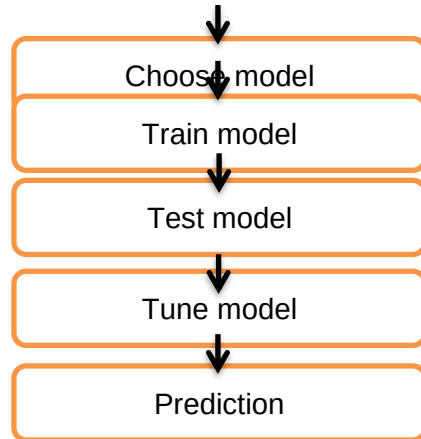


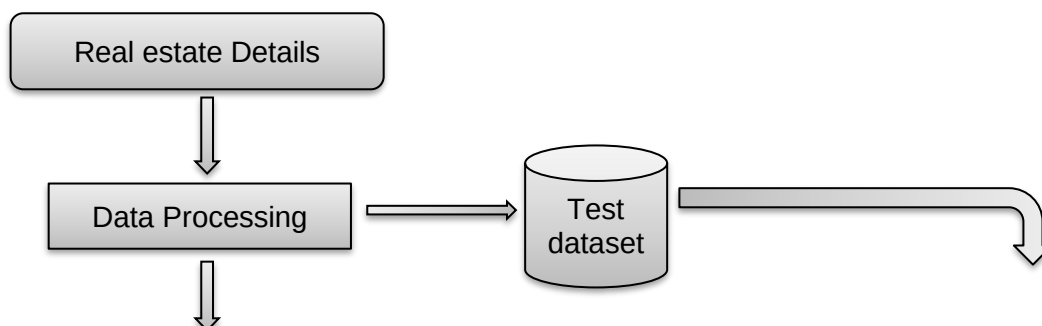
Fig: Process of dataflow diagram

Training the Dataset

- The first line imports iris data set which is already predefined in sklearn module and raw data set is basically a table which contains information about various varieties.
- For example, to import any algorithm and `train_test_split` class from sklearn and numpy module for use in this program.
- To encapsulate `load_data()` method in `data_dataset` variable. Further divide the dataset into training data and test data using `train_test_split` method. The X prefix in variable denotes the feature values and y prefix denotes target values.
- This method divides dataset into training and test data randomly in ratio of 67:33 / 70:30. Then we encapsulate any algorithm.
- In the next line, we fit our training data into this algorithm so that computer can get trained using this data. Now the training part is complete.

Testing the Dataset

- Now, the dimensions of new features in a numpy array called 'n' and we want to predict the species of this features and to do using the `predict` method which takes this array as input and spits out predicted target value as output.
- So, the predicted target value comes out to be 0. Finally to find the test score which is the ratio of no. of predictions found correct and total predictions made and finding accuracy score method which basically compares the actual values of the test set with the predicted values.



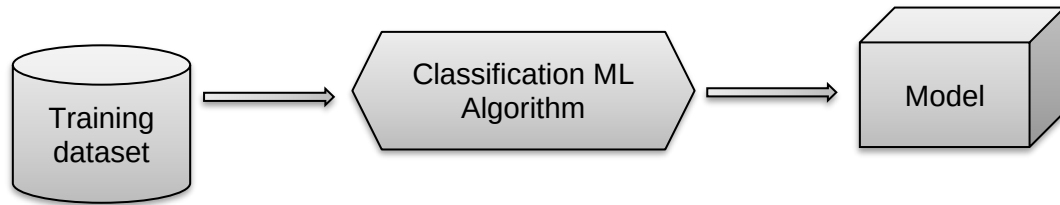


Fig: Architecture of Proposed model

General Properties

Create cells freely to explore the given data and it should *not* perform too many operations in each cell. One option that can take with this is to do a lot of explorations in an initial notebook. These don't have to be organized, but make sure you use enough comments to understand the purpose of each code cell. Then, after done with your analysis, create a duplicate notebook where it will trim the excess and organize steps so that have a flowing, cohesive report and make sure that informed on the steps that are taking in your investigation. Follow every code cell, or every set of related code cells, with a markdown cell to describe to the reader what was found in the preceding cell. Try to make it so that the reader can then understand what they will be seeing in the following cell.

Project Goals

It will be used in order to do the analysis. Later on, some analysis will also be done to find the most relevant attributes, the factors that affect the prediction result the most.

- Exploration data analysis of variable identification
 - Loading the given dataset
 - Import required libraries packages
 - Analyze the general properties
 - Find duplicate and missing values
 - Checking unique and count values
- Uni-variate data analysis
 - Rename, add data and drop the data
 - To specify data type
- Exploration data analysis of bi-variate and multi-variate
 - Plot diagram of pairplot, heatmap, bar chart and Histogram
- Method of Outlier detection with feature engineering
 - Pre-processing the given dataset
 - Splitting the test and training dataset
 - Comparing the Decision tree and Logistic regression model and random forest

- Comparing algorithm to predict the result
 - Based on the best accuracy

List of Modules:

- Variable Identification Process
- Exploration data analysis of preprocessing
- To train a model from ML algorithms
- Prediction of GUI based Application results

Variable Identification Process

Validation techniques in machine learning are used to get the error rate of the Machine Learning (ML) model, which can be considered as close to the true error rate of the dataset. If the data volume is large enough to be representative of the population, you may not need the validation techniques. However, in real-world scenarios, to work with samples of data that may not be a true representative of the population of given dataset. To finding the missing value, duplicate value and description of data type whether it is float variable or integer. The sample of data used to provide an unbiased evaluation of a model fit on the training dataset while tuning model hyper parameters. The following diagram is given dataset.

The evaluation becomes more biased as skill on the validation dataset is incorporated into the model configuration. The validation set is used to evaluate a given model, but this is for frequent evaluation. It as machine learning engineers use this data to fine-tune the model hyper parameters. Data collection, data analysis, and the process of addressing data content, quality, and structure can add up to a time-consuming to-do list. During the process of data identification, it helps to understand your data and its properties; this knowledge will help you choose which algorithm to use to build your model. For example, time series data can be analyzed by regression algorithms; classification algorithms can be used to analyze discrete data. For example to show the data type format of given dataset.

A number of different data cleaning tasks using Python's [Pandas library](#) and specifically, it focus on probably the biggest data cleaning task, missing values and it able to more quickly clean data. It wants to spend less time cleaning data, and more time exploring and modeling. Some of these sources are just simple random mistakes. Other times, there can be a deeper reason why data is missing. It's important to understand these [different types of missing data](#) from a statistics point of view. The type of missing data will influence how to deal with filling in the missing values and to detect missing values, and do some basic imputation and detailed statistical approach for [dealing with missing data](#). Before, joint into code, it's important to understand the sources of missing data. Here are some typical reasons why data is missing:

- User forgot to fill in a field.
- Data was lost while transferring manually from a legacy database.
- There was a programming error.
- Users chose not to fill out a field tied to their beliefs about how the results would be used or interpreted.

Variable identification with Uni-variate, Bi-variate and Multi-variate analysis:

- import libraries for access and functional purpose and read the given dataset
- General Properties of Analyzing the given dataset
- Display the given dataset in the form of data frame
- show columns
- shape of the data frame
- To describe the data frame
- Checking data type and information about dataset
- Checking for duplicate data
- Checking Missing values of data frame
- Checking unique values of data frame
- Checking count values of data frame
- Rename and drop the given data frame
- To specify the type of values
- To create extra columns

Data Validation/ Cleaning/Preparing Process

Importing the library packages with loading given dataset. To analyzing the variable identification by data shape, data type and evaluating the missing values, duplicate values. A validation dataset is a sample of data held back from training your model that is used to give an estimate of model skill while tuning model's and procedures that you can use to make the best use of validation and test datasets when evaluating your models. Data cleaning / preparing by rename the given dataset and drop the column etc. to analyze the uni-variate, bi-variate and multi-variate process. The steps and techniques for data cleaning will vary from dataset to dataset. The primary goal of data cleaning is to detect and remove errors and anomalies to increase the value of data in analytics and decision making.

Outlier detection process

Many machine learning algorithms are sensitive to the range and distribution of attribute values in the input data. Outliers in input data can skew and mislead the training process of machine learning algorithms resulting in longer training times, less accurate models and ultimately poorer results.

Even before predictive models are prepared on training data, outliers can result in misleading representations and in turn misleading interpretations of collected data. Outliers can skew the summary distribution of attribute values in descriptive statistics like mean and standard deviation and in plots such as histograms and scatterplots, compressing the body of the data. Finally, outliers can represent examples of data instances that are relevant to the problem such as anomalies in the case of fraud detection and computer security.

It couldn't fit the model on the training data and can't say that the model will work accurately for the real data. For this, we must assure that our model got the correct patterns from

the data, and it is not getting up too much noise. Cross-validation is a technique in which we train our model using the subset of the data-set and then evaluate using the complementary subset of the data-set.

The three steps involved in cross-validation are as follows:

1. Reserve some portion of sample data-set.
2. Using the rest data-set train the model.
3. Test the model using the reserve portion of the data-set.

Advantages of train/test split:

1. This runs K times faster than Leave One Out cross-validation because K-fold cross-validation repeats the train/test split K-times.
2. Simpler to examine the detailed results of the testing process.

Advantages of cross-validation:

1. More accurate estimate of out-of-sample accuracy.
2. More “efficient” use of data as every observation is used for both training and testing.

Pre-processing refers to the transformations applied to our data before feeding it to the algorithm. Data Preprocessing is a technique that is used to convert the raw data into a clean data set. In other words, whenever the data is gathered from different sources it is collected in raw format which is not feasible for the analysis. To achieving better results from the applied model in Machine Learning method of the data has to be in a proper manner. Some specified Machine Learning model needs information in a specified format, for example, Random Forest algorithm does not support null values. Therefore, to execute random forest algorithm null values have to be managed from the original raw data set. And another aspect is that data set should be formatted in such a way that more than one Machine Learning and Deep Learning algorithms are executed in given dataset.

Comparing Algorithm with prediction in the form of best accuracy result

It is important to compare the performance of multiple different machine learning algorithms consistently and it will discover to create a test harness to compare multiple different machine learning algorithms in Python with scikit-learn. It can use this test harness as a template on your own machine learning problems and add more and different algorithms to compare. Each model will have different performance characteristics. Using resampling methods like cross validation, you can get an estimate for how accurate each model may be on unseen data. It needs to be able to use these estimates to choose one or two best models from the suite of models that you have created. When have a new dataset, it is a good idea to visualize the data using different techniques in order to look at the data from different perspectives. The same idea applies to model selection. You should use a number of different ways of looking at the estimated accuracy of your machine learning algorithms in order to choose the one or two to finalize. A way to do this is to use different visualization methods to show the average accuracy, variance and other

properties of the distribution of model accuracies.

In the next section you will discover exactly how you can do that in Python with scikit-learn. The key to a fair comparison of machine learning algorithms is ensuring that each algorithm is evaluated in the same way on the same data and it can achieve this by forcing each algorithm to be evaluated on a consistent test harness.

In the example below 5 different algorithms are compared:

- Logistic Regression
- Random Forest
- K-Nearest Neighbors
- Decision tree
- Support Vector Machines

The K-fold cross validation procedure is used to evaluate each algorithm, importantly configured with the same random seed to ensure that the same splits to the training data are performed and that each algorithm is evaluated in precisely the same way. Before that comparing algorithm, Building a Machine Learning Model using install Scikit-Learn libraries. In this library package have to done preprocessing, linear model with logistic regression method, cross validating by KFold method, ensemble with random forest method and tree with decision tree classifier. Additionally, splitting the train set and test set. To predicting the result by comparing accuracy.

GUI Application:

We have often been asked: "Is there no Tk for Python?" or "Is Tkinter the same as Tk?" Of course, there is Tk for Python. Without Tk Python would be less attractive to many users. Tk is called Tkinter in Python, or to be precise, Tkinter is the Python interface for Tk. Tkinter is an acronym for "Tk interface". If there were a single package which might be called the "standard" GUI toolkit for Python, it would be [Tkinter](#). Tkinter is a wrapper around [Tcl/Tk](#), a popular graphical interface and language

Algorithm Explanation

In machine learning and statistics, classification is a supervised learning approach in which the computer program learns from the data input given to it and then uses this learning to classify new observation. This data set may simply be bi-class (like identifying whether the person is male or female or that the mail is spam or non-spam) or it may be multi-class too. Some examples of classification problems are: speech recognition, handwriting recognition, bio metric identification, document classification etc. In Supervised Learning, algorithms learn from labeled data. After understanding the data, the algorithm determines which label should be given to new data based on pattern and associating the patterns to the unlabeled new data.

Used Python Packages:

sklearn:

- In python, sklearn is a machine learning package which include a lot of ML algorithms.
- Here, we are using some of its modules like train_test_split, DecisionTreeClassifier or Logistic Regression and accuracy_score.

NumPy:

- It is a numeric python module which provides fast maths functions for calculations.
- It is used to read data in numpy arrays and for manipulation purpose.

Pandas:

- Used to read and write different files.
- Data manipulation can be done easily with data frames.

Matplotlib:

- Data visualization is a useful way to help with identify the patterns from given dataset.
- Data manipulation can be done easily with data frames.

Logistic Regression

It is a statistical method for analysing a data set in which there are one or more independent variables that determine an outcome. The outcome is measured with a dichotomous variable (in which there are only two possible outcomes). The goal of logistic regression is to find the best fitting model to describe the relationship between the dichotomous characteristic of interest (dependent variable = response or outcome variable) and a set of independent (predictor or explanatory) variables. Logistic regression is a Machine Learning classification algorithm that is used to predict the probability of a categorical dependent variable. In logistic regression, the dependent variable is a binary variable that contains data coded as 1 (yes, success, etc.) or 0 (no, failure, etc.).

In other words, the logistic regression model predicts $P(Y=1)$ as a function of X . Logistic regression Assumptions:

- Binary logistic regression requires the dependent variable to be binary.
- For a binary regression, the factor level 1 of the dependent variable should represent the desired outcome.
- Only the meaningful variables should be included.
- The independent variables should be independent of each other. That is, the model should have little.
- The independent variables are linearly related to the log odds.

- Logistic regression requires quite large sample sizes.

Decision Tree

It is one of the most powerful and popular algorithm. Decision-tree algorithm falls under the category of supervised learning algorithms. It works for both continuous as well as categorical output variables. Assumptions of Decision tree:

- At the beginning, we consider the whole training set as the root.
- Attributes are assumed to be categorical for information gain, attributes are assumed to be continuous.
- On the basis of attribute values records are distributed recursively.
- We use statistical methods for ordering attributes as root or internal node.

Decision tree builds classification or regression models in the form of a tree structure. It breaks down a data set into smaller and smaller subsets while at the same time an associated decision tree is incrementally developed. A decision node has two or more branches and a leaf node represents a classification or decision. The topmost decision node in a tree which corresponds to the best predictor called root node. Decision trees can handle both categorical and numerical data. Decision tree builds classification or regression models in the form of a tree structure. It utilizes an if-then rule set which is mutually exclusive and exhaustive for classification. The rules are learned sequentially using the training data one at a time. Each time a rule is learned, the tuples covered by the rules are removed.

This process is continued on the training set until meeting a termination condition. It is constructed in a top-down recursive divide-and-conquer manner. All the attributes should be categorical. Otherwise, they should be discretized in advance. Attributes in the top of the tree have more impact towards in the classification and they are identified using the information gain concept. A decision tree can be easily over-fitted generating too many branches and may reflect anomalies due to noise or outliers.

K-Nearest Neighbor (KNN)

K-Nearest Neighbor is a supervised machine learning algorithm which stores all instances correspond to training data points in n-dimensional space. When an unknown discrete data is received, it analyzes the closest k number of instances saved (nearest neighbors) and returns the most common class as the prediction and for real-valued data it returns the mean of k nearest neighbors. In the distance-weighted nearest neighbor algorithm, it weights the contribution of each of the k neighbors according to their distance using the following query giving greater weight to the closest neighbors.

Usually KNN is robust to noisy data since it is averaging the k-nearest neighbors. The k-nearest-neighbors algorithm is a classification algorithm, and it is supervised: it takes a bunch of labeled points and uses them to learn how to label other points. To label a new point, it looks at the labeled points closest to that new point (those are its nearest neighbors), and has those neighbors vote, so whichever label the most of the neighbors have is the label for the new point (the “k” is

the number of neighbors it checks). Makes predictions about the validation set using the entire training set. KNN makes a prediction about a new instance by searching through the entire set to find the k “closest” instances. “Closeness” is determined using a proximity measurement (Euclidean) across all features.

Random Forest

Random forests or random decision forests are an ensemble learning method for classification, regression and other tasks, that operate by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees. Random decision forests correct for decision trees' habit of over fitting to their training set. Random forest is a type of supervised machine learning algorithm based on [ensemble learning](#). Ensemble learning is a type of learning where you join different types of algorithms or same algorithm multiple times to form a more powerful prediction model. The [random forest](#) algorithm combines multiple algorithm of the same type i.e. multiple decision *trees*, resulting in *a forest of trees*, hence the name "Random Forest". The random forest algorithm can be used for both regression and classification tasks.

The following are the basic steps involved in performing the random forest algorithm:

- Pick N random records from the dataset.
- Build a decision tree based on these N records.
- Choose the number of trees you want in your algorithm and repeat steps 1 and 2.
- In case of a regression problem, for a new record, each tree in the forest predicts a value for Y (output). The final value can be calculated by taking the average of all the values predicted by all the trees in forest. Or, in case of a classification problem, each tree in the forest predicts the category to which the new record belongs. Finally, the new record is assigned to the category that wins the majority vote.

Support Vector Machines

A classifier that categorizes the data set by setting an optimal hyper plane between data. I chose this classifier as it is incredibly versatile in the number of different kernelling functions that can be applied and this model can yield a high predictability rate. Support Vector Machines are perhaps one of the most popular and talked about machine learning algorithms. They were extremely popular around the time they were developed in the 1990s and continue to be the go-to method for a high-performing algorithm with little tuning.

- How to disentangle the many names used to refer to support vector machines.
- The representation used by SVM when the model is actually stored on disk.
- How a learned SVM model representation can be used to make predictions for new data.
- How to learn an SVM model from training data.
- How to best prepare your data for the SVM algorithm.
- Where you might look to get more information on SVM.

Project Requirements

General:

Requirements are the basic constrains that are required to develop a system.

Requirements are collected while designing the system. The following are the requirements that are to be discussed.

1. Functional requirements
2. Non-Functional requirements
3. Environment requirements
 - A. Hardware requirements
 - B. software requirements

Functional requirements:

The software requirements specification is a technical specification of requirements for the software product. It is the first step in the requirements analysis process. It lists requirements of a particular software system. The following details to follow the special libraries like sk-learn, pandas, numpy, matplotlib and seaborn.

Non-Functional Requirements:

Process of functional steps,

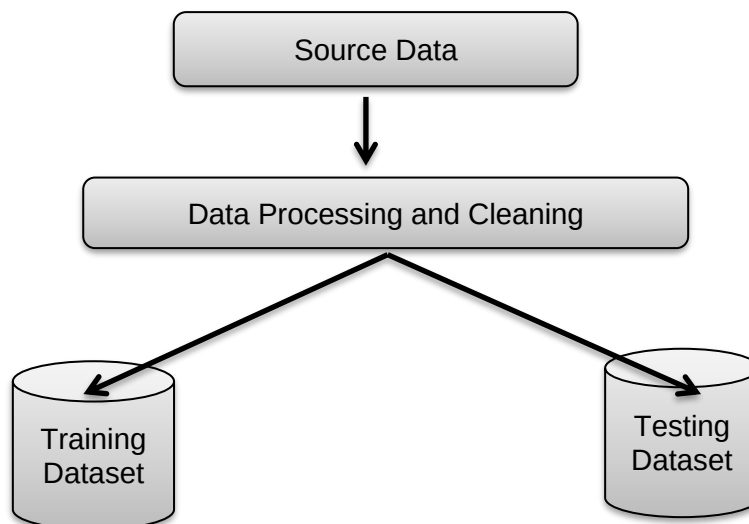
1. Problem define
2. Preparing data
3. Evaluating algorithms
4. Improving results
5. Prediction the result

DESIGN ENGINEERING

General

Design is meaningful engineering representation of something that is to be built. Software design is a process design is the perfect way to accurately translate requirements in to a finished software product. Design creates a representation or model, provides detail about software data structure, architecture, interfaces and components that are necessary to implement a system.

Work flow diagram



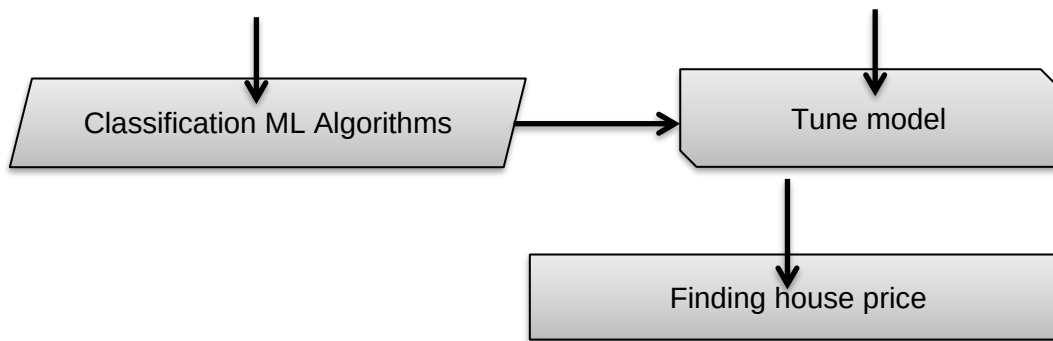


Fig: Workflow Diagram

Software Description:

Anaconda is a [free and open-source](#) distribution of the [Python](#) and [R](#) programming languages for [scientific computing](#) ([data science](#), [machine learning](#) applications, large-scale data processing, [predictive analytics](#), etc.), that aims to simplify [package management](#) and deployment. Package versions are managed by the [package management system](#) “Conda”. The Anaconda distribution is used by over 12 million users and includes more than 1400 popular data-science packages suitable for Windows, Linux, and MacOS. So, Anaconda distribution comes with more than 1,400 packages as well as the [Conda](#) package and virtual environment manager called Anaconda Navigator and it eliminates the need to learn to install each library independently. The open source packages can be individually installed from the Anaconda repository with the `conda install` command or using the `pip install` command that is installed with Anaconda. [Pip packages](#) provide many of the features of conda packages and in most cases they can work together. Custom packages can be made using the `conda build` command, and can be shared with others by uploading them to Anaconda Cloud,^[10] [PyPI](#) or other repositories. The default installation of Anaconda2 includes Python 2.7 and Anaconda3 includes Python 3.7. However, you can create new environments that include any version of Python packaged with conda.

Anaconda Navigator

Anaconda Navigator is a desktop graphical user interface (GUI) included in Anaconda distribution that allows users to launch applications and 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, install them in an environment, run the packages and update them. It is available for Windows, macOS and Linux.

The following applications are available by default in Navigator:

- JupyterLab
- Jupyter Notebook
- QtConsole

- Spyder
- Glueviz
- Orange
- Rstudio
- Visual Studio Code

Conda:

Conda is an [open source](#), [cross-platform](#), language-agnostic [package manager](#) and environment management system that installs, runs and updates packages and their dependencies. It was created for Python programs, but it can package and distribute software for any language (e.g., [R](#)), including multi-languages. The Conda package and environment manager is included in all versions of Anaconda, Miniconda, and Anaconda Repository.

The Jupyter Notebook

The Jupyter Notebook is an open-source web application that allows you to create and share documents that contain live code, equations, visualizations and narrative text. Uses include: data cleaning and transformation, numerical simulation, statistical modeling, data visualization, machine learning, and much more.

Notebook document:

Notebook documents (or “notebooks”, all lower case) are documents produced by the Jupyter Notebook App, which contain both computer code (e.g. python) and rich text elements (paragraph, equations, figures, links, etc...). Notebook documents are both human-readable documents containing the analysis description and the results (figures, tables, etc.) as well as executable documents which can be run to perform data analysis.

Jupyter Notebook App:

The *Jupyter Notebook App* is a server-client application that allows editing and running notebook documents via a web browser. The *Jupyter Notebook App* can be executed on a local desktop requiring no internet access (as described in this document) or can be installed on a remote server and accessed through the internet. In addition to displaying/editing/running notebook documents, the *Jupyter Notebook App* has a “Dashboard” (Notebook Dashboard), a “control panel” showing local files and allowing to open notebook documents or shutting down their kernels.

kernel:

A notebook *kernel* is a “computational engine” that executes the code contained in a Notebook document. The *ipython kernel*, referenced in this guide, executes python code.

Kernels for many other languages exist ([official kernels](#)). When you open a Notebook document, the associated *kernel* is automatically launched. When the notebook is *executed* (either cell-by-cell or with menu *Cell -> Run All*), the *kernel* performs the computation and produces the results. Depending on the type of computations, the *kernel* may consume significant CPU and RAM. Note that the RAM is not released until the *kernel* is shut-down.

Notebook Dashboard:

The *Notebook Dashboard* is the component which is shown first when you launch Jupyter Notebook App. The *Notebook Dashboard* is mainly used to open notebook documents, and to manage the running kernels (visualize and shutdown). The *Notebook Dashboard* has other features similar to a file manager, namely navigating folders and renaming/deleting files.

Working Process:

- Download and install anaconda and get the most useful package for machine learning in Python.
- Load a dataset and understand its structure using statistical summaries and data visualization.
- machine learning models, pick the best and build confidence that the accuracy is reliable.

Python is a popular and powerful interpreted language. Unlike R, Python is a complete language and platform that you can use for both research and development and developing production systems. There are also a lot of modules and libraries to choose from, providing multiple ways to do each task. It can feel overwhelming.

The best way to get started using Python for machine learning is to complete a project.

- It will force you to install and start the Python interpreter (at the very least).
- It will give you a bird's eye view of how to step through a small project.
- It will give you confidence, maybe to go on to your own small projects.

When you are applying machine learning to your own datasets, you are working on a project. A machine learning project may not be linear, but it has a number of well known steps:

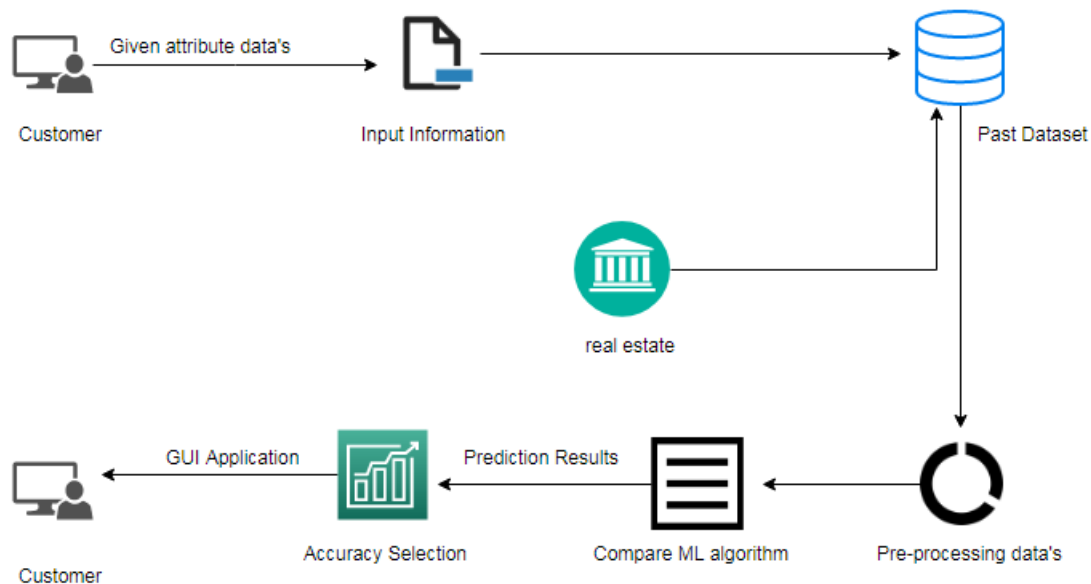
- Define Problem.
- Prepare Data.
- Evaluate Algorithms.
- Improve Results.
- Present Results.

The best way to really come to terms with a new platform or tool is to work through a machine learning project end-to-end and cover the key steps. Namely, from loading data, summarizing data, evaluating algorithms and making some predictions.

Here is an overview of what we are going to cover:

1. Installing the Python anaconda platform.
2. Loading the dataset.
3. Summarizing the dataset.
4. Visualizing the dataset.
5. Evaluating some algorithms.
6. Making some predictions.

System Architecture:



Screen Shots:

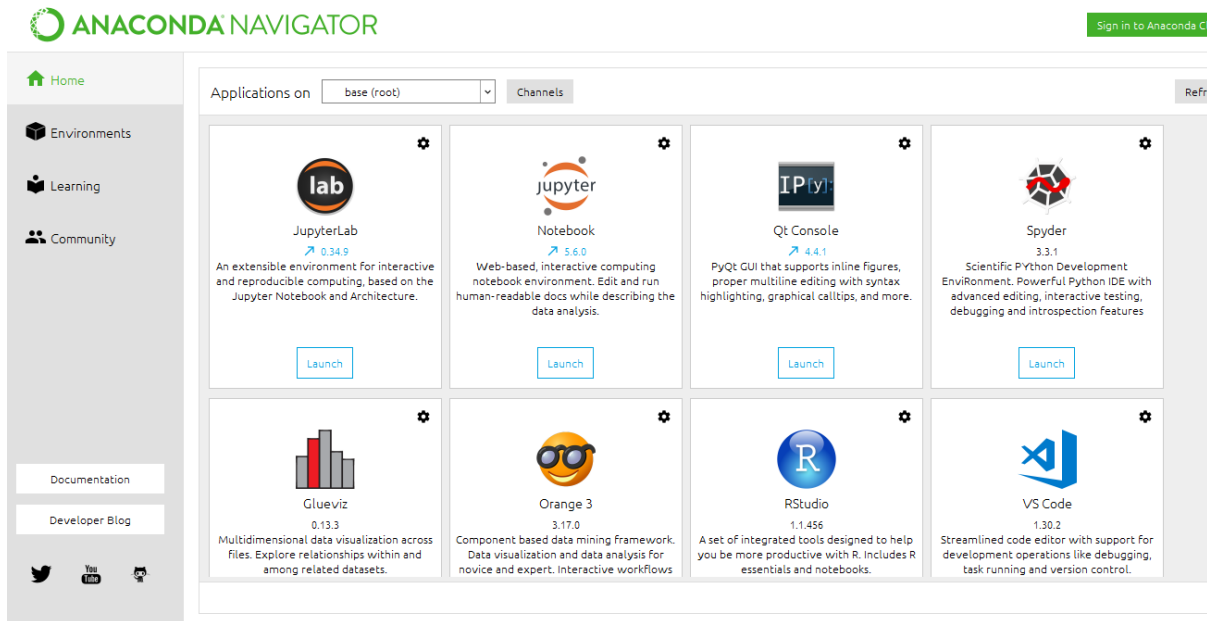


Fig: Open the anaconda navigator

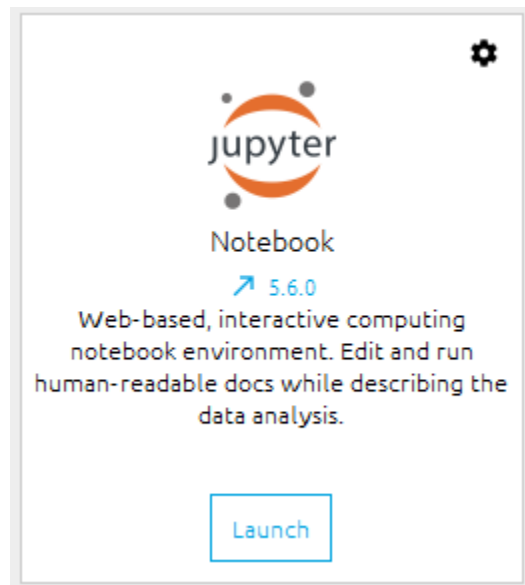


Fig: Launch the jupyter notebook platform

Conclusion:

The analytical process started from data cleaning and processing, missing value, exploratory analysis and finally model building and evaluation. The best accuracy on public test set is higher accuracy score will be find out for future with comparing accuracy calculations. This brings some of the following insights about house prize.

Future Work

- It working on large dataset would yield a better and real picture about the model
House price prediction wants to automate the detecting the good or not from eligibility process (real time).
- To automate this process by show the prediction result in web application or desktop application.
- To optimize the work to implement in Artificial Intelligence environment.