A Thesis/Project/Dissertation Report

on

HOUSE PRICE PREDICTION USING MACHINE LEARNING

Fall 2021 – 2022

Submitted in partial fulfilment

of the requirement for the award of the degree of

B. TECH CSE

GALGOTIAS
UNIVERSITY

(Established under Galgotias University Uttar Pradesh Act No. 14 of 2011)

Under The Supervision

Ms. Indrakumari

Assistant Professor.

Reviewer

Mr. P. Raja Kumar

Group ID: BT3253

S. No	Enrolment Number	Admission Number	Student Name	Degree / Branch	Sem
1	19021180018	19SCSE1010913	RUDRANSH SOLANKI	B.tech/CSE	5
2	19021050105	19SCSE1050016	RUDRA PRATAP SINGH BISHT	B.tech/CSE	5

SCHOOL OF COMPUTING SCIENCE AND ENGINEERING

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

GALGOTIAS UNIVERSITY, GREATER NOIDA

DECEMBER, 2021

SCHOOL OF COMPUTING SCIENCE AND ENGINEERING GALGOTIAS UNIVERSITY, GREATER NOIDA

Candidate's declaration

I/We hereby certify that the work which is being presented in the project, entitled **"HOUSE PRICE PREDICTION USING MACHINE LEARNING"** in partial fulfilment of the requirements for the award of the B.tech CSE submitted in the School of Computing Science and Engineering of Galgotias University, Greater Noida, is an original work carried out during the period of month, Year to Month and Year, under the supervision of Name… Designation, Department of Computer Science and Engineering/Computer Application and Information and Science, of School of Computing Science and Engineering , Galgotias University, Greater Noida

The matter presented in the project has not been submitted by me/us for the award of any other degree of this or any other places.

Rudransh Solanki, 19SCSE1010913

Rudra Pratap Singh Bisht, 19SCSE1050016

This is to certify that the above statement made by the candidates is correct to the best of my knowledge.

Supervisor Name

Designation

Certificate

The Final Project Viva-Voce examination of Name: RUDRANSH SOLANKI and RUDRA PRATAP SINGH BISHT Admission No: 19SCSE1010913 has been held on 26/10/2021 and his/her work is recommended for the award of B.tech CSE.

Signature of Examiner(s) **Signature of Supervisor(s)**

Signature of Project Coordinator **Signature of Dean**

Date: November, 2013

Place: Greater Noida

TABLE OF CONTENTS

Abstract

Machine learning is a branch of Artificial Intelligence which is used to analyse the data more smartly. It automates the process using certain algorithms to minimize human intervention in the process. This machine learning project, we are going to predict the house price using python. This project will help the sellers and buyers to have an overview of the situation so that they can act accordingly. 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. House Price Index (HPI) is commonly used to estimate the changes in housing price. Since housing price is strongly correlated to other factors such as location, area, population, it requires other information apart from HPI to predict individual housing price. There has been a considerably large number of papers adopting traditional machine learning approaches

to predict housing prices accurately, but they rarely concern about the performance of individual models and neglect the less popular yet complex models. As a result, to explore various impacts of features on prediction methods, this paper will apply both traditional and advanced machine learning approaches to investigate the difference among several advanced models. This paper will also comprehensively validate multiple techniques in model implementation on regression and provide an optimistic result for housing price prediction. Real estate is the least transparent industry in our ecosystem. Housing prices keep changing day in and day out and sometimes are hyped rather than being based on valuation. Predicting housing prices with real factors is the main crux of our research project. Here we aim to make our evaluations based on every basic parameter that is considered while determining the price. We use various regression techniques in this pathway, and our results are not sole determination of one technique rather it is the weighted mean of various techniques to give most accurate results. The results proved that this approach yields minimum error and maximum accuracy than individual algorithms applied. We also propose to use real-time neighborhood details using Google maps to get exact real-world valuations. Predictive models for determining the sale price of houses in cities like Bengaluru is still re maining as more challenging and tricky task. The sale price of properties in cities like Benga luru depends on a nu mber of interdependent factors. Key factors that might affect the price include area of the property, location of the property and its amenities. In this research work, an analytical study has been carried out by considering the data set that remains open to the public by illustrating the available housing properties in machine hackathon plat form. The data set has nine features. In this study, an attempt has been made to construct a predictive model for evaluating the price based on the factors that affect the price. Modeling explorations apply some regression techniques such as multiple linear regression (Least Squares), Lasso and Ridge regression models, support vector regression, and boosting algorithms such as Extreme Gradient Boost Regression (XG Boost). Such models are used to build a predictive model, and to pick the best performing model by performing a comparative analysis on the predictive errors obtained between these models. Here, the attempt is to construct a predictive model for evaluating the price based on factors that affects the price.

Keywords

Machine learning, Regression Technique, Classification Technique, Cross validation Technique, K-means

1.Machine learning
Machine learning is the science of getting computers to act without being explicitly programmed. In the past decade, machine learning has given us self-driving cars, practical speech recognition, effective web search, and a vastly improved understanding of the human genome. Machine learning is so pervasive today that you probably use it dozens of times a day without knowing it. Many researchers also think it is the best way to make progress towards human-level AI. In this class, you will learn about the most effective machine learning techniques, and gain practice implementing them and getting them to work for yourself. More importantly, you'll learn about not only the theoretical underpinnings of learning, but also gain the practical know-how needed to quickly and powerfully apply these techniques to new problems. Finally, you'll learn about some of Silicon Valley's best practices in innovation as it pertains to machine learning and AI.
This course provides a broad introduction to machine learning, datamining, and statistical pattern recognition. Topics include: (i) Supervised learning (parametric/non-parametric algorithms, support vector machines, kernels, neural networks). (ii) Unsupervised learning (clustering, dimensionality reduction, recommender systems, deep learning). (iii) Best practices in machine learning (bias/variance theory; innovation process in machine learning and AI). The course will also draw from numerous case studies and applications, so that you'll also learn how to apply learning algorithms to building smart robots (perception, control), text understanding (web search, anti-spam), computer vision, medical informatics, audio, database mining, and other areas.

2.Regression Technique
Regression techniques consist of finding a mathematical relationship between measurements of two variables, *y* and *x*, such that the value of variable *y* can be predicted from a measurement of the other variable, *x*. However, regression techniques should not be regarded as a magic formula that can fit a good relationship to measurement data in all circumstances, as the characteristics of data must satisfy certain conditions. In determining the suitability of measurement data for the application of regression techniques, it is recommended practice to draw an approximate graph of the measured data points, as this is often the best means of detecting aspects of data that make it unsuitable for regression analysis. Drawing a graph of data will indicate, for example, whether any data points appear to be erroneous. This may indicate that human mistakes or instrument malfunctions have affected the erroneous data points, and it is assumed that any such data points will be checked for correctness.
Regression techniques cannot be applied successfully if the deviation of any particular data point from the line to be fitted is greater than the maximum possible error calculated for the measured variable (i.e., the predicted sum of all systematic and random errors). The nature of some measurement data sets is such that this criterion cannot be satisfied, and any attempt to apply regression techniques is doomed to failure. In that event, the only valid course of action is to express the measurements in tabular form. This can then be used as an $x-y$ look-up table, from which values of the variable *y* corresponding to particular values of *x* can be read off. In many cases, this problem of large errors in some data points only becomes apparent during the process of attempting to fit a relationship by regression.

A further check that must be made before attempting to fit a line or curve to measurements of two variables, x and y, is to examine data and look for any evidence that both variables are subject to random errors. It is a clear condition for the validity of regression techniques that only one of the measured variables is subject to random errors, with no error in the other variable. If random errors do exist in both measured variables, regression techniques cannot be applied and recourse must be made instead to correlation analysis (covered later in this chapter). Simple examples of a situation where both variables in a measurement data set are subject to random errors are measurements of human height and weight, and no attempt should be made to fit a relationship between them by regression.

Having determined that the technique is valid, the regression procedure is simplest if a straight-line relationship exists between the variables, which allows a relationship of the form $y = a + bx$ to be estimated by linear least-squares regression. Unfortunately, in many cases, a straight-line relationship between points does not exist, which is shown readily by plotting raw data points on a graph. However, knowledge of physical laws governing data can often suggest a suitable alternative form of relationship between the two sets of variable measurements, such as a quadratic relationship or a higher order polynomial relationship. Also, in some cases, the measured variables can be transformed into a form where a linear relationship exists. For example, suppose that two variables, y and x, are related according to $y = ax^c$. A linear relationship from this can be derived, using a logarithmic transformation, as $\log(y) = \log(a) + c\log(x)$.

Thus, if a graph is constructed of $\log(y)$ plotted against $\log(x)$, the parameters of a straight-line relationship can be estimated by linear least-squares regression.

All quadratic and higher order relationships relating one variable, y, to another variable, x, can be represented by a power series of the form:

y=a0+a1x+a2x2+…+apxp.

Estimation of the parameters $a_0 \ldots a_p$ is very difficult if p has a large value. Fortunately, a relationship where p only has a small value can be fitted to most data sets. Quadratic least-squares regression is used to estimate parameters where p has a value of two; for larger values of p, polynomial least-squares regression is used for parameter estimation.

Where the appropriate form of relationship between variables in measurement data sets is not obvious either from visual inspection or from consideration of physical laws, a method that is effectively a trial and error one has to be applied. This consists of estimating the parameters of successively higher order relationships between y and x until a curve is found that fits data sufficiently closely. What level of closeness is acceptable is considered later in the section on confidence tests.

3.Classification Technique

Classification techniques have shown recently their usefulness for complex process diagnosis. Besides the fact that no physical model for the process is required, they enable to study the problem of sensor location. Preliminary studies made previously in the domain of chemical process diagnosis have been the initial key point to extend its application to the medical diagnosis framework. Despite the behavioral difference, both domains exhibit many common practices.

However, medical diagnosis recently has brought serious challenges such as high dimensionality (gene expression profiling) and heterogeneity of data (symbolic histo-pathological factors). We show here that both challenges can be overcome and used in return to improve complex process diagnosis.

4.Cross validation Technique

Cross-validation is a statistical method used to estimate the skill of machine learning models.

It is commonly used in applied machine learning to compare and select a model for a given predictive modeling problem because it is easy to understand, easy to implement, and results in skill estimates that generally have a lower bias than other methods.

In this tutorial, you will discover a gentle introduction to the k-fold cross-validation procedure for estimating the skill of machine learning models.

After completing this tutorial, you will know:

1.That k-fold cross validation is a procedure used to estimate the skill of the model on new data.

2.There are common tactics that you can use to select the value of k for your dataset.

3.There are commonly used variations on cross-validation such as stratified and repeated that are available in scikit-learn.

5.K-means

K-means clustering is one of the simplest and popular unsupervised machine learning algorithms. Typically, unsupervised algorithms make inferences from datasets using only input vectors without referring to known, or labelled, outcomes.

AndreyBu, who has more than 5 years of machine learning experience and currently teaches people his skills, says that “the objective of K-means is simple: group similar data points together and discover underlying patterns. To achieve this objective, K-means looks for a fixed number (k) of clusters in a dataset.”

A cluster refers to a collection of data points aggregated together because of certain similarities.

You’ll define a target number k, which refers to the number of centroids you need in the dataset.

A centroid is the imaginary or real location representing the center of the cluster.

Every data point is allocated to each of the clusters through reducing the in-cluster sum of squares.

In other words, the K-means algorithm identifies k number of centroids, and then allocates every data point to the nearest cluster, while keeping the centroids as small as possible.
The ‘means’ in the K-means refers to averaging of the data; that is, finding the centroid.

CHAPTER 1

Introduction

Keep tormenting data until it starts revealing its hidden secrets. Yes, it can be done but there's a way around it. Making predictions using Machine Learning isn't just about grabbing the data and feeding it to algorithms. The algorithm might spit out some prediction but that's not what you are aiming for. The difference between good data science professionals and naive data science aspirants is that the former set follows this process religiously. The process is as follows: 1. Understand the problem: Before getting the data, we need to understand the problem we are trying to solve. If you know the domain, think of which factors could play an epic role in solving the problem. If you don't know the domain, read about it. 2. Hypothesis Generation: This is quite important, yet it is often forgotten. In simple words, hypothesis generation refers to creating a set of features which could influence the target variable given a confidence interval (taken as 95% all the time). We can do this before looking at the data to avoid biased thoughts. This step often helps in creating new features. 3. Get Data: Now, we download the data and look at it. Determine which features are available and which aren't, how many features we generated in hypothesis generation hit the mark, and which ones could be created. Answering these questions will set us on the right track. 4. Data Exploration: We can't determine everything by just looking at the data. We need to dig deeper. This step helps us understand the nature of variables (skewed, missing, zero variance feature) so that they can be treated properly. It involves creating charts, graphs (univariate and bivariate analysis), and cross-tables to understand the behavior of features. 5. *Data Preprocessing: *Here, we impute missing values and clean string variables (remove space, irregular tabs, data time format) and anything that shouldn't be there. This step is usually followed along with the data exploration stage. 6. Feature Engineering: Now, we create and add new features to the data set. Most of the ideas for these features come during the hypothesis generation stage. 7. Model Training: Using a suitable algorithm, we train the model on the given data set. 8. Model Evaluation: Once the model is trained, we evaluate the model's performance using a suitable error metric. Here, we also look for variable importance, i.e., which variables have proved to be significant in determining the target variable. And, accordingly we can shortlist the best variables and train the model again. 9. Model Testing: Finally, we test the model on the unseen data (test data) set. Estimating the sale prices of houses is one of the basic projects to have on your Data Science CV. By finishing this article, you will be able to predict continuous variables using various types of linear regression algorithm.

Why linear regression? Linear regression is an algorithm used to predict values that are continuous in nature. It became more popular because it is the best algorithm to start with if you are a newbie to ML.

To predict the sale prices we are going to use the following linear regression algorithms: Ordinal Least Square (OLS) algorithm, Ridge regression algorithm, Lasso regression algorithm, Bayesian regression algorithm, and lastly Elastic Net regression algorithm. These algorithms can be feasibly implemented in python with the use of the scikit-learn package.

Finally, we conclude which model is best suitable for the given case by evaluating each of them using the evaluation metrics provided by the scikit-learn package.

Problem Formulation

Housing prices are an important reflection of the economy, and housing price ranges are of great interest for both buyers and sellers. Ask a home buyer to describe their dream house, and they probably won't begin with the height of the basement ceiling or the proximity to an east-west railroad. But this playground competition's data-set proves that much more influences price negotiations than the number of bedrooms or a white-picket fence.

The goal of this statistical analysis is to help us understand the relationship between house features and how these variables are used to predict house price.

Objective

1.Predict the house price

2.Using two different models in terms of minimizing the difference between predicted and actual rating

Fig 1

With 79 explanatory variables describing (almost) every aspect of residential homes in Ames, Iowa, this competition challenges you to predict the final price of each home.

Required tools

- **PYTHON**
- **TENSORFLOW**
- **Anaconda**
- **Libraries - pandas, NumPy, matplotlib**

PYTHON

Python is an interpreted high-level general-purpose programming language. Its design philosophy emphasizes code readability with its use of significant indentation. Its language constructs as well as its object-oriented approach aim to help programmers write clear, logical code for small and large-scale projects.

Python is dynamically-typed and garbage-collected. It supports multiple programming paradigms, including structured (particularly, procedural), object-oriented and functional programming. It is often described as a "batteries included" language due to its comprehensive standard library.

Guido van Rossum began working on Python in the late 1980s, as a successor to the ABC programming language, and first released it in 1991 as Python 0.9.0. Python 2.0 was released in 2000 and introduced new features, such as list comprehensions and a cycle-detecting garbage collection system (in addition to reference counting). Python 3.0 was released in 2008 and was a major revision of the language that is not completely backward-compatible. Python 2 was discontinued with version 2.7.18 in 2020.
Python consistently ranks as one of the most popular programming languages.

TENSORFLOW

Machine learning is a complex discipline. But implementing machine learning models is far less daunting and difficult than it used to be, thanks to machine learning frameworks—such as Google's TensorFlow—that ease the process of acquiring data, training models, serving predictions, and refining future results.

Created by the Google Brain team, TensorFlow is an open source library for numerical computation and large-scale machine learning. TensorFlow bundles together a slew of machine learning and deep learning (aka neural networking) models and algorithms and makes them useful by way of a common metaphor. It uses Python to provide a convenient front-end API for building applications with the framework, while executing those applications in high-performance C++.
TensorFlow can train and run deep neural networks for handwritten digit classification, image recognition, word embeddings, recurrent neural networks, sequence-to-sequence models for machine translation, natural language processing, and PDE (partial differential equation) based simulations. Best of all, TensorFlow supports production prediction at scale, with the same models used for training.

How TensorFlow works

TensorFlow allows developers to create dataflow graphs—structures that describe how data moves through a graph, or a series of processing nodes. Each node in the graph represents a mathematical operation, and each connection or edge between nodes is a multidimensional data array, or tensor.

TensorFlow provides all of this for the programmer by way of the Python language. Python is easy to learn and work with, and provides convenient ways to express how high-level abstractions can be coupled together. Nodes and tensors in TensorFlow are Python objects, and TensorFlow applications are themselves Python applications.

The actual math operations, however, are not performed in Python. The libraries of transformations that are available through TensorFlow are written as high-performance C++ binaries. Python just directs traffic between the pieces, and provides high-level programming abstractions to hook them together.

TensorFlow applications can be run on most any target that's convenient: a local machine, a cluster in the cloud, iOS and Android devices, CPUs or GPUs. If you use Google's own cloud, you can run TensorFlow on Google's custom TensorFlow Processing Unit (TPU) silicon for further acceleration. The resulting models created by TensorFlow, though, can be deployed on most any device where they will be used to serve predictions.

ANACONDA

Anaconda is a 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. The distribution includes data-science packages suitable for Windows, Linux, and macOS. It is developed and maintained by Anaconda, Inc., which was founded by Peter Wang and Travis Oliphant in 2012. As an Anaconda, Inc. product, it is also known as Anaconda Distribution or Anaconda Individual Edition, while other products from the company are Anaconda Team Edition and Anaconda Enterprise Edition, both of which are not free.

Package versions in Anaconda are managed by the package management system conda.This package manager was spun out as a separate open-source package as it ended up being useful on its own and for things other than Python.There is also a small, bootstrap version of Anaconda called Miniconda, which includes only conda, Python, the packages they depend on, and a small number of other packages.

PANDAS

Python Pandas Tutorial

Pandas is an open-source, BSD-licensed Python library providing high-performance, easy-to-use data structures and data analysis tools for the Python programming language. Python with Pandas is used in a wide range of fields including academic and commercial domains including finance, economics, Statistics, analytics, etc. In this tutorial, we will learn the various features of Python Pandas and how to use them in practice.

Audience

This tutorial has been prepared for those who seek to learn the basics and various functions of Pandas. It will be specifically useful for people working with data cleansing and analysis. After completing this tutorial, you will find yourself at a moderate level of expertise from where you can take yourself to higher levels of expertise.

Prerequisites
You should have a basic understanding of Computer Programming terminologies. A basic understanding of any of the programming languages is a plus. Pandas library uses most of the functionalities of NumPy. It is suggested that you go through our tutorial on NumPy before proceeding with this tutorial. You can access it from − NumPy Tutorial

NUMPY
NumPy (Numerical Python) is an open source Python library that's used in almost every field of science and engineering. It's the universal standard for working with numerical data in Python, and it's at the core of the scientific Python and PyData ecosystems. NumPy users include everyone from beginning coders to experienced researchers doing state-of-the-art scientific and industrial research and development. The NumPy API is used extensively in Pandas, SciPy, Matplotlib, scikit-learn, scikit-image and most other data science and scientific Python packages.

The NumPy library contains multidimensional array and matrix data structures (you'll find more information about this in later sections). It provides ndarray, a homogeneous n-dimensional array object, with methods to efficiently operate on it. NumPy can be used to perform a wide variety of mathematical operations on arrays. It adds powerful data structures to Python that guarantee efficient calculations with arrays and matrices and it supplies an enormous library of high-level mathematical functions that operate on these arrays and matrices.

MATPLOT
Matplotlib is one of the most popular Python packages used for data visualization. It is a cross-platform library for making 2D plots from data in arrays. It provides an object-oriented API that helps in embedding plots in applications using Python GUI toolkits such as PyQt, WxPythonotTkinter. It can be used in Python and IPython shells, Jupyter notebook and web application servers also.

Audience
This tutorial is designed for those learners who wish to acquire knowledge on the basics of data visualization.

Prerequisites
Matplotlib is written in Python and makes use of NumPy, the numerical mathematics extension of Python. We assume that the readers of this tutorial have basic knowledge of Python.

CHAPTER 2 Literature Survey/Project Design

Problems

Mean Absolute Error (MAE) is the mean of the absolute value of the errors:

$$\frac{1}{n}\sum_{i = 1}^{n} |y_i - \hat{y}_i|$$

Mean Squared Error (MSE) is the mean of the squared errors.

$$\frac{1}{n}\sum_{i = 1}^{n} (y_i - \hat{y}_i)^2$$

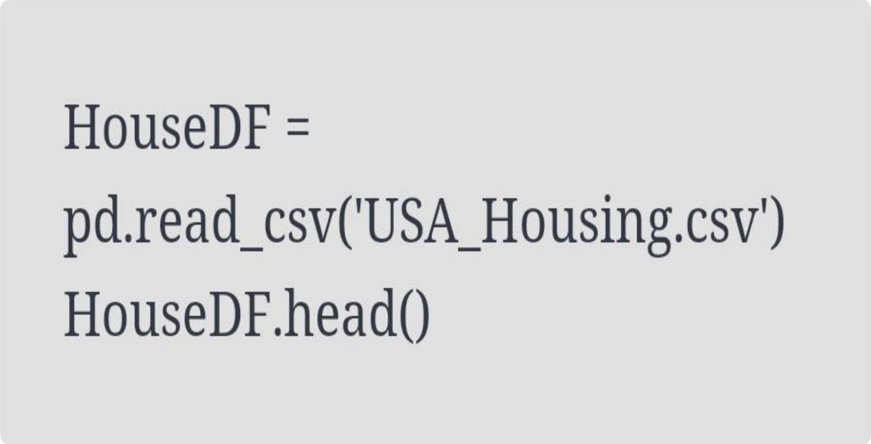
Root Mean Squared Error (RMSE) is the square root of the mean of the squared errors:

$$\sqrt{\frac{1}{n}\sum_{i = 1}^{n} (y_i - \hat{y}_i)^2}$$

Implementation and Description of Project Modules -

Step1.

Importing Data and Checking out As data is in the CSV file, we will read the CSV using pandas read_csv function and check the first 5 rows of the data frame using head().

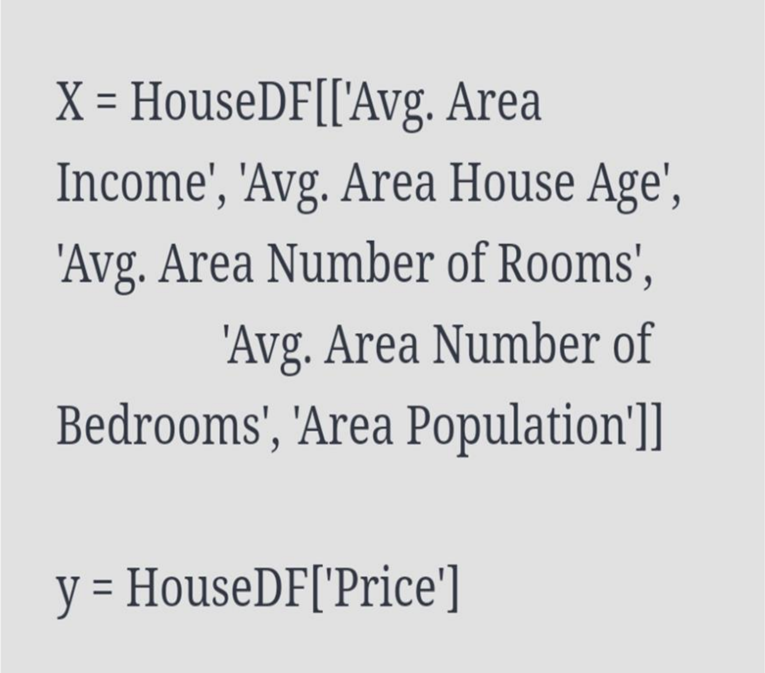


```
HouseDF =
pd.read_csv('USA_Housing.csv')
HouseDF.head()
```

Fig. 4

Step 2.

Get Data Ready for Training a Linear Regression Model Let’s now begin to train out the regression model. We will need to first split up our data into an X list that contains the features to train on, and a y list with the target variable, in this case, the Price column. We will ignore the Address column.

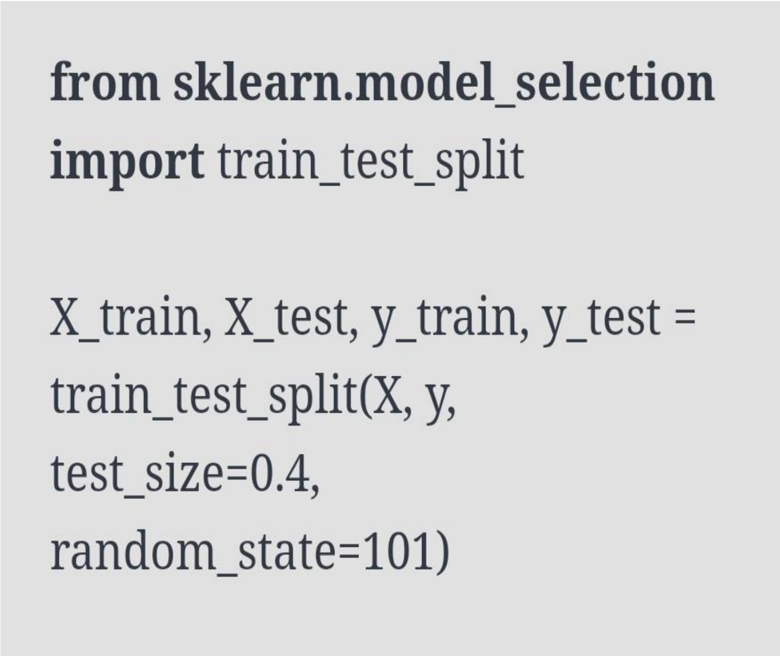


```
X = HouseDF[['Avg. Area
Income', 'Avg. Area House Age',
'Avg. Area Number of Rooms',
            'Avg. Area Number of
Bedrooms', 'Area Population']]

y = HouseDF['Price']
```

Fig. 5

Step 3.

Split Data into Train, Test Now we will split our dataset into a training set and testing set using sklearn train_test_split(). the training set will be going to use for training the model and testing set for testing the model. We are creating a split of 40% training data and 60% of the training set. .X_train and y_train contain data for the training model. X_test and y_test contain data for the testing model. X and y are features and target variable name.

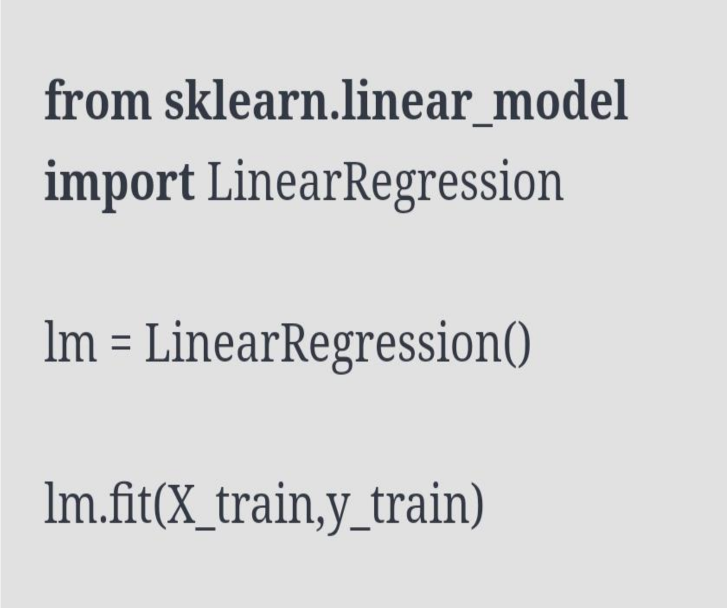


```
from sklearn.model_selection
import train_test_split

X_train, X_test, y_train, y_test =
train_test_split(X, y,
test_size=0.4,
random_state=101)
```

Fig. 6

Step 4.

Creating and Training the Linear Regression Model. We will import and create sklearn linearmodel LinearRegression object and fit the training dataset in it.

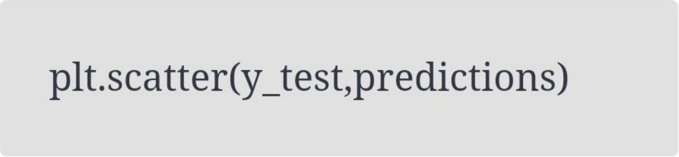


```
from sklearn.linear_model
import LinearRegression

lm = LinearRegression()

lm.fit(X_train,y_train)
```

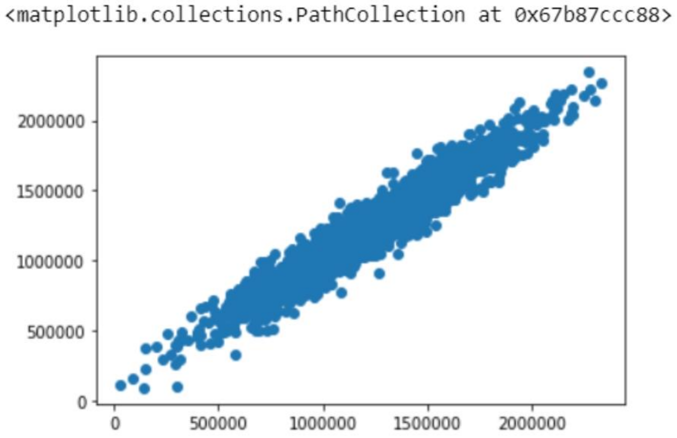
Fig. 7

Step 5.

Predictions from our Linear Regression Model Let’s find out the predictions of our test set and see how well it perform.



```
plt.scatter(y_test,predictions)
```

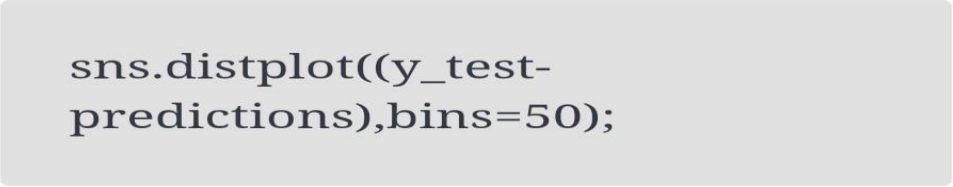


```
<matplotlib.collections.PathCollection at 0x67b87ccc88>
```

Fig. 8

Step 6.

In the above scatter plot, we see data is in a line form, which means our model has done good predictions.



```
sns.distplot((y_test-
predictions),bins=50);
```

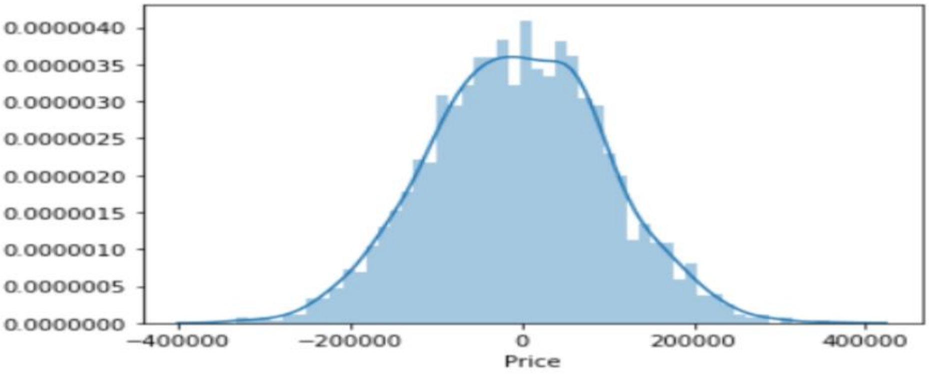


Fig. 9

In the above histogram plot, we see data is in bell shape (Normally Distributed), which means our model has done good predictions. Regression Evaluation Metrics Here are three common evaluation metrics for regression.

CHAPTER 3 Functionality/Working of Project

Methodology

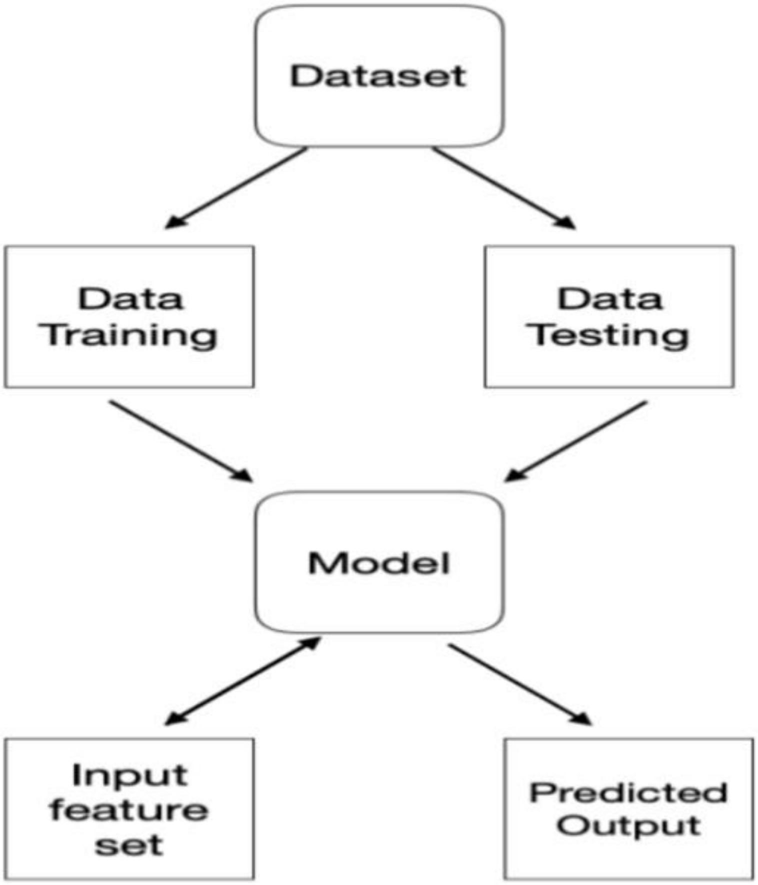


Fig 3

Datasets
A data set is a collection of related, discrete items of related data that may be accessed individually or in combination or managed as a whole entity.

A data set is organized into some type of data structure. In a database, for example, a data set might contain a collection of business data (names, salaries, contact information, sales figures, and so forth). The database itself can be considered a data set, as can bodies of data within it related to a particular type of information, such as sales data for a particular corporate department.

The term data set originated with IBM, where its meaning was similar to that of file. In an IBM mainframe operating system, a data set s a named collection of data that contains individual data units organized (formatted) in a specific, IBM-prescribed way and accessed by a specific access

method based on the data set organization. Types of data set organization include sequential, relative sequential, indexed sequential, and partitioned. Access methods include the Virtual Sequential Access Method (VSAM) and the Indexed Sequential Access Method (ISAM).
A data set is also an older and now deprecated term for modem.

Data Testing

There is a lot of attention for testing methods like security testing, performance testing or regression testing. Testing agile and test automation are also hot topics these days. But how to handle the data (automated or not) which you need for testing software is addressed less often. That is actually quite strange since software development and testing would stand or fall on carefully prepared data cases. You can't use just some data or just a random test case. In order to test a software application effectively, you'll need good and representative data set. The ideal test set identifies all the application errors with a smallest possible data set. In short, you need a relatively small (test) data set that is realistic, valid and versatile.

HOW TO CREATE TEST DATA

Data can be created 1) manually, 2) by using data generation tools or 3) it can be retrieved from existing production environment. The data set can consist of synthetic (fake) data, but preferably it consists of representative (real) data (for security reasons this data should of course be masked) with good coverage of the test cases. This will provide the best software quality and that is what we all want ultimately.

"The ideal test data identifies all the application errors with a smallest possible data set."

So beware with dummy data, generated by a random name generator or a credit card number generator for example. These generators provide you with sample data that offers no challenges to the software being tested. Of course synthetic data can be used to enrich and/or mask your test database.

TEST DATA PREPARATION IN SOFTWARE TESTING

The preparation of data for testing is a very time-consuming phase in software testing. Various researches show that 30-60% of the tester's time is spent on searching, maintaining and generating data for testing and development. The main reasons for this are the following:

Testing teams do not have access to the data sources
Delay in giving production data access to the testers by developers
Large volumes of data
Data dependencies/combinations
Long refreshment times

1. Testing teams do not have access to the data sources

Especially with the GDPR, PCI, HIPAA and other data security regulations in place, access to data sources is limited. As a result only a few employees are able to access the data sources. The advantage of this policy is that the chance of a data breach is reduced. The disadvantage is that test teams are dependent on others and that long waiting times arise.

2. Delay in giving production data access to the testers by developers

Agile is not yet being used everywhere. In many organizations multiple teams and users work on the same project and thus on the same databases. Besides that it causes conflicts, the data set often changes and doesn’t contain the right (up to date) data when it’s the next team’s turn to test the application.

3. Large volumes of data
Compiling data from a production database is like searching for a pin in a haystack. You need the special cases to perform good tests and they are hard to find when you have to dig in dozens of terabytes.

4. Data dependencies/combinations
Most data values are dependent on other data values in order to get recognized. When preparing the cases, these dependencies make it a lot more complex and therefore time-consuming.

5. Long refreshment times
Most testing teams do not have the facility to self-refresh the test database. That means that they have to go to the DBA to ask for a refreshment. Some teams have to wait for days or even weeks before this refresh is done.

Data Training
What Does Training Data Mean?
Training data is an extremely large dataset that is used to teach a machine learning model. For supervised ML models, the training data is labeled. The data used to train unsupervised ML models is not labeled.

The idea of using training data in machine learning programs is a simple concept, but it is also very foundational to the way that these technologies work. The training data is an initial set of data used to help a program understand how to apply technologies like neural networks to learn and produce sophisticated results. It may be complemented by subsequent sets of data called validation and testing sets.

Training data is also known as a training set, training dataset or learning set.

Techopedia Explains Training Data
The training set is the material through which the computer learns how to process information. Machine learning uses algorithms – it mimics the abilities of the human brain to take in diverse inputs and weigh them, in order to produce activations in the brain, in the individual neurons. Artificial neurons replicate a lot of this process with software – machine learning and neural network programs that provide highly detailed models of how our human thought processes work.

With that in mind, training data can be structured in different ways. For sequential decision trees and those types of algorithms, it would be a set of raw text or alphanumerical data that gets classified or otherwise manipulated. On the other hand, for convolutional neural networks that have to do with image processing and computer vision, the training set is often composed of large numbers of images. The idea is that because the machine learning program is so complex and so sophisticated, it uses iterative training on each of those images to eventually be able to recognize features, shapes and even subjects such as people or animals. The training data is absolutely essential to the process – it can be thought of as the “food” the system uses to operate.

Models

Algorithms used in machine learning fall roughly into three categories: supervised, unsupervised, and reinforcement learning. Supervised learning involves feedback to indicate when a prediction is right or wrong, whereas unsupervised learning involves no response: The algorithm simply tries to categorize data based on its hidden structure. Reinforcement learning is similar to supervised learning in that it receives feedback, but it’s not necessarily for each input or state. This tutorial explores the ideas behind these learning models and some key algorithms used for each.

Machine-learning algorithms continue to grow and evolve. In most cases, however, algorithms tend to settle into one of three models for learning. The models exist to adjust automatically in some way to improve their operation or behavior.

In supervised learning, a data set includes its desired outputs (or labels) such that a function can calculate an error for a given prediction. The supervision comes when a prediction is made and an error produced (actual vs. desired) to alter the function and learn the mapping.

In unsupervised learning, a data set doesn’t include a desired output; therefore, there’s no way to supervise the function. Instead, the function attempts to segment the data set into “classes” so that each class contains a portion of the data set with common features.

Finally, in reinforcement learning, the algorithm attempts to learn actions for a given set of states that lead to a goal state. An error is provided not after each example (as is the case for supervised learning) but instead on receipt of a reinforcement signal (such as reaching the goal state). This behavior is similar to human learning, where feedback isn’t necessarily provided for all actions but when a reward is warranted.

Supervised learning

Supervised learning is the simplest of the learning models to understand. Learning in the supervised model entails creating a function that can be trained by using a training data set, then applied to unseen data to meet some predictive performance. The goal is to build the function so that it generalizes well over data it has never seen.

You build and test a mapping function with supervised learning in two phases (see image below). In the first phase, you segment a data set into two types of samples: training data and test data. Both training and test data contain a test vector (the inputs) and one or more known desired output values. You train the mapping function with the training data set until it meets some level of performance (a metric for how accurately the mapping function maps the training data to the associated desired output). In the context of supervised learning, this occurs with each training sample, where you use the error (actual vs. desired output) to alter the mapping function. In the next phase, you test the trained mapping function against the test data. The test data represents data that has not been used for training and provides a good measure for how well the mapping function generalizes to unseen data.

Neural networks

A neural network processes an input vector to a resulting output vector through a model inspired by neurons and their connectivity in the brain. The model consists of layers of neurons interconnected through weights that alter the importance of certain inputs over others. Each neuron includes an activation function that determines the output of the neuron (as a function of its input vector multiplied by its weight vector). The output is computed by applying the input vector to the input layer of the network, then computing the outputs of each neuron through the network (in a feed-forward fashion).

Decision trees

A decision tree is a supervised learning method for classification. Algorithms of this variety create trees that predict the result of an input vector based on decision rules inferred from the features present in the data. Decision trees are useful because they're easy to visualize so you can understand the factors that lead to a result.

Unsupervised learning

Unsupervised learning is also a relatively simple learning model, but as the name suggests, it lacks a critic and has no way to measure its performance. The goal is to build a mapping function that categorizes the data into classes based on features hidden within the data.

As with supervised learning, you use unsupervised learning in two phases. In the first phase, the mapping function segments a data set into classes. Each input vector becomes part of a class, but the algorithm cannot apply labels to those classes.The segmentation of the data into classes may be the result (from which you can then draw conclusions about the resulting classes), but you can use these classes further depending on the application. One such application is a recommendation system, where the input vector may represent the characteristics or purchases of a user, and users within a class represent those with similar interests who can then be used for marketing or product recommendations.

K-means clustering

k-means clustering is a simple and popular clustering algorithm that originated in signal processing. The goal of the algorithm is to partition examples from a data set into k clusters. Each example is a numerical vector that allows the distance between vectors to be calculated as a Euclidean distance.

The simple example below visualizes the partitioning of data into k = 2 clusters, where the Euclidean distance between examples is smallest to the centroid (center) of the cluster, which indicates its membership.The k-means algorithm is extremely simple to understand and implement. You begin by randomly assigning each example from the data set into a cluster, calculate the centroid of the clusters as the mean of all member examples, then iterate the data set to determine whether an example is closer to the member cluster or the alternate cluster (given that k = 2). If the member is closer to the alternate cluster, the example is moved to the new cluster and its centroid recalculated. This process continues until no example moves to the alternate cluster.

Adaptive resonance theory

Adaptive resonance theory (ART) is a family of algorithms that provide pattern recognition and prediction capabilities. You can divide ART along unsupervised and supervised models, but I focus here on the unsupervised side. ART is a self-organizing neural network architecture. The approach allows learning new mappings while maintaining existing knowledge.

Like k-means, you can use ART1 for clustering, but it has a key advantage in that rather than defining k at runtime, ART1 can alter the number of clusters based on the data.

ART1 includes three key features: a comparison field (used to determine how a new feature vector fits within the existing categories), a recognition field (contains neurons that represent the active clusters), and a reset module. When the input vector is applied, the comparison field identifies the cluster in which it most closely fits. If the input vector matches in the recognition field above a vigilance parameter, then the connections to the neuron in the recognition field are updated to account for this new vector. Otherwise, a new neuron is created in the recognition field to account for a new cluster. When a new neuron is created, the existing neuron weights are not updated, allowing them to retain the existing knowledge. All examples of the data set are applied in this way until no example input vector changes cluster. At this point, training is complete.

Reinforcement learning

Reinforcement learning is an interesting learning model, with the ability not just to learn how to map an input to an output but to map a series of inputs to outputs with dependencies (Markov decision processes, for example). Reinforcement learning exists in the context of states in an environment and the actions possible at a given state. During the learning process, the algorithm randomly explores the state–action pairs within some environment (to build a state–action pair table), then in practice of the learned information exploits the state–action pair rewards to choose the best action for a given state that lead to some goal state. You can learn more about reinforcement learning in “Train a software agent to behave rationally with reinforcement learning.”Consider a simple agent that plays blackjack. The states represent the sum of the cards for the player. The actions represent what a blackjack-playing agent may do — in this case, hit or stand. Training an agent to play blackjack would involve many hands of poker, where reward for a given state–action nexus is given for winning or losing. For example, the value for a state of 10 would be 1.0 for hit and 0.0 for stand (indicating that hit is the optimal choice). For state 20, the learned reward would likely be 0.0 for hit and 1.0 for stand. For a less-straightforward hand, a state of 17 may have action values of 0.95 stand and 0.05 hit. This agent would then probabilistically stand 95 percent of the time and hit 5 percent of the time. These rewards would be leaned over many hands of poker, indicating the best choice for a given state (or hand).

Q-learning

Q-learning is one approach to reinforcement learning that incorporates Q values for each state–action pair that indicate the reward to following a given state path. The general algorithm for Q-learning is to learn rewards in an environment in stages. Each state encompasses taking actions for states until a goal state is reached. During learning, actions selected are done so probabilistically (as a function of the Q values), which allows exploration of the state-action space. When the goal state is reached, the process begins again, starting from some initial position.

Q values are updated for each state–action pair as an action is selected for a given state. The Q value for the state–action pair is updated with some reward provided by the move (may be nothing) along with the maximum Q value available for the new state reached by applying the action to the current state (discounted by a discount factor). This is further discounted by a learning rate that determines how valuable new information is over old. The discount factor indicates how important future rewards are over short-term rewards. Note that the environment may be filled with negative and positive rewards, or only the goal state may indicate a reward.This algorithm is performed over many epochs of reaching the goal state, permitting the Q values to be updated based on the probabilistic selection of actions for states. When complete, the Q values can be used greedily (use the action with the largest Q value for a given state) to exploit the knowledge gained so that the goal state is reached optimally.

Going further
Machine-learning benefits from a diverse set of algorithms that suit different needs. Supervised learning algorithms learn a mapping function for a data set with an existing classification, where unsupervised learning algorithms can categorize an unlabeled data set based on some hidden features in the data. Finally, reinforcement learning can learn policies for decision-making in an uncertain environment through iterative exploration of that environment.

Ready to go deeper? Take a look at the Get started with machine learning series to understand the principles of machine learning and get working knowledge of the different phases and tasks.

Input feature set
How to Choose a Feature Selection Method For Machine Learning

Feature selection is the process of reducing the number of input variables when developing a predictive model.

It is desirable to reduce the number of input variables to both reduce the computational cost of modeling and, in some cases, to improve the performance of the model.

Statistical-based feature selection methods involve evaluating the relationship between each input variable and the target variable using statistics and selecting those input variables that have the strongest relationship with the target variable. These methods can be fast and effective, although the choice of statistical measures depends on the data type of both the input and output variables.

As such, it can be challenging for a machine learning practitioner to select an appropriate statistical measure for a dataset when performing filter-based feature selection.

In this post, you will discover how to choose statistical measures for filter-based feature selection with numerical and categorical data.

After reading this post, you will know:

There are two main types of feature selection techniques: supervised and unsupervised, and supervised methods may be divided into wrapper, filter and intrinsic.
Filter-based feature selection methods use statistical measures to score the correlation or dependence between input variables that can be filtered to choose the most relevant features.
Statistical measures for feature selection must be carefully chosen based on the data type of the input variable and the output or response variable.
Kick-start your project with my new book Data Preparation for Machine Learning, including step-by-step tutorials and the Python source code files for all examples.

CHAPTER 4 Result and Discussion

Complete work plan layout

I. Import the dependencies and libraries

import pandas

import numpy

sklearn - it is the machine learning library for python

linear_model from sklearn

train_test_split from sklearn.model_selection

(it is a function that splits our data into training and testing sets)

II. Now load the dataset for the particular location which you want to analyze. Here we are going to use Boston housing dataset from sklearn.datasets. Now create a variable called boston and assign it to load_boston() function. Now print it using print(Boston)

III. Next step is to transform the dataset into the data frame. Create variable df_x and df_y.

IV. Now get some statistics from the data set, count, mean, etc.

V. Initialize the linear regression model - reg = linear_model.LinearRegrssion()

VI. Split the data into 67% as training and 33% as testing data.

VII. Now train the model with our training data

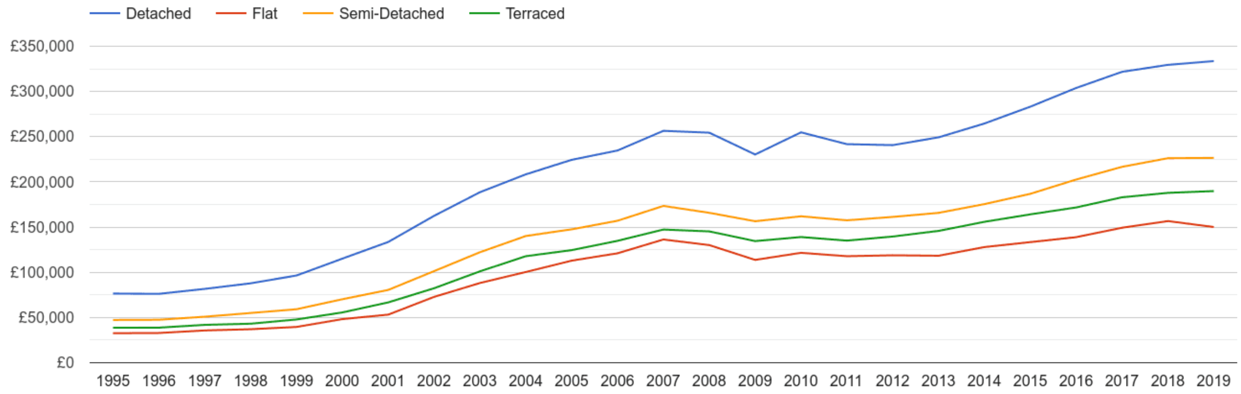


Fig 2

I. Print the coefficients/weights for each feature/column of our model - print(reg.coef)

II. Now print the predictions on our test data.

III. Print the actual values - print(y_test)

IV. Check the model performance using MSE (Mean Squared Error).

V. Now check the model performance using MSE and sklearn.metrics. Visualize the differences between the actual price and predicted price

VI. Similarly train the model using various models - Random Forest Regressor, XGBoost Regressor, SVM Regressor,

VII. Finally, evaluate and compare all the models to get proper output. (As far as with my experience XGBoost Regression works best for this dataset)

Similarly, you can predict the house price of various locations by importing the data of the particular place. (You can get the data from the real-estate websites like 99acres, airbnb, homes.com, trulia, realtor, etc.

pandas- It is an open-source library written for python to perform data analysis and manipulation.

Matplotlib - It is a plotting library for python program and its mathematics extension NumPy.

NumPy - It is a package for python for scientific computing to perform different operations.

sklearn/scikit-learn - It is a free machine learning library developed for python programming language under BSD license which is majorly used for data analysis and data mining. It also supports various machine learning algorithms such as SVM, random forests, k-neighbours, etc.

Predicted Output

Why are Predictions Important?

Machine learning model predictions allow businesses to make highly accurate guesses as to the likely outcomes of a question based on historical data, which can be about all kinds of things – customer churn likelihood, possible fraudulent activity, and more. These provide the business with insights that result in tangible business value. For example, if a model predicts a customer is likely to churn, the business can target them with specific communications and outreach that will prevent the loss of that customer.

DataRobot + Predictions

The DataRobot AI Cloud Platform allows users to easily develop models that make highly accurate predictions. It streamlines the data science process so that users get high-quality predictions in a fraction of the time it took using traditional methods, allowing them to more quickly implement those predictions and see the impact on their bottom line.

In order to start making predictions with DataRobot, you need to deploy the model into a production application. For more details, see the deployment wiki entry or the DataRobot model deployment briefing.

Data collection

Why is Data Collection Important?

Collecting data allows you to capture a record of past events so that we can use data analysis to find recurring patterns. From those patterns, you build predictive models using machine learning algorithms that look for trends and predict future changes.

Predictive models are only as good as the data from which they are built, so good data collection practices are crucial to developing high-performing models. The data need to be error-free (garbage in, garbage out) and contain relevant information for the task at hand. For example, a loan default model would not benefit from tiger population sizes but could benefit from gas prices over time.

Data Collection + DataRobot

DataRobot partners with several organizations that assist in collecting, storing, and transforming data to make it ready for predictive modeling. Once you've collected and prepared the appropriate data for your specific business problem, you can easily import it into the DataRobot AI Cloud Platform no matter where you've stored it. Then, DataRobot automatically creates new features and builds and evaluates hundreds of machine learning models which you can immediately deploy into production.

Data collection is the process of gathering information on variables in a systematic manner. This helps in finding answers too many questions, hypothesis and evaluate outcomes.

What does Prediction mean in Machine Learning?

"Prediction" refers to the output of an algorithm after it has been trained on a historical dataset and applied to new data when forecasting the likelihood of a particular outcome, such as whether or not a customer will churn in 30 days. The algorithm will generate probable values for an unknown variable for each record in the new data, allowing the model builder to identify what that value will most likely be.

The word “prediction” can be misleading. In some cases, it really does mean that you are predicting a future outcome, such as when you’re using machine learning to determine the next best action in a marketing campaign. Other times, though, the “prediction” has to do with, for example, whether or not a transaction that already occurred was fraudulent. In that case, the transaction already happened, but you’re making an educated guess about whether or not it was legitimate, allowing you to take the appropriate action.

What is Data Collection?
As a society, we’re generating data at an unprecedented rate (see big data). These data can be numeric (temperature, loan amount, customer retention rate), categorical (gender, color, highest degree earned), or even free text (think doctor’s notes or opinion surveys). Data collection is the process of gathering and measuring information from countless different sources. In order to use the data we collect to develop practical artificial intelligence (AI) and machine learning solutions, it must be collected and stored in a way that makes sense for the business problem at hand.

Data collection and pre-processing techniques
Preparing data for use in machine learning models and deep learning
Whether they are new to deep learning or looking for a refresher, mobile app developers find that QDN blog posts are a good introduction to AI and machine learning (ML). Posts like Mobile AI Through Machine Learning Algorithms and AI Machine Learning Algorithms – How a Neural Network Works set the stage for using the Qualcomm® Neural Processing SDK for AI. You can find all the latest blogs on our Artificial Intelligence Get Started page.

The entry point to the development cycle of any ML project is the data preparation stage.

As shown in the primitive ML development pipeline below, data preparation precedes the training and learning stage, known as feature extraction, of any ML model. Hence, the importance of executing this stage correctly from the outset.

Learning Resources
Within the data preparation stage are the data collection and data pre-processing stages.

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

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

Pre-cleaned, freely available datasets. If the problem statement (for example, image classification, object recognition) aligns with a clean, pre-existing, properly formulated dataset, then take advantage of existing, open-source expertise.
Web crawling and scraping. Automated tools, bots and headless browsers can crawl and scrape websites for data.
Private data. ML engineers can create their own data. This is helpful when the amount of data required to train the model is small and the problem statement is too specific to generalize over an open-source dataset.
Custom data. Agencies can create or crowdsource the data for a fee.
Data pre-processing
Real-world raw data and images are often incomplete, inconsistent and lacking in certain behaviors or trends. They are also likely to contain many errors. So, once collected, they are pre-processed into a format the machine learning algorithm can use for the model.

Pre-processing includes a number of techniques and actions:

Data cleaning. These techniques, manual and automated, remove data incorrectly added or classified.
Data imputations. Most ML frameworks include methods and APIs for balancing or filling in missing data. Techniques generally include imputing missing values with standard deviation, mean, median and k-nearest neighbors (k-NN) of the data in the given field.
Oversampling. Bias or imbalance in the dataset can be corrected by generating more observations/samples with methods like repetition, bootstrapping or Synthetic Minority Over-Sampling Technique (SMOTE), and then adding them to the under-represented classes.
Data integration. Combining multiple datasets to get a large corpus can overcome incompleteness in a single dataset.
Data normalization. The size of a dataset affects the memory and processing required for iterations during training. Normalization reduces the size by reducing the order and magnitude of data.
Those techniques point to the types of machine learning available to mobile app developers.

Data visualization

Data Visualization is the pictorial or graphical representation of information..It enables to grasp difficult concepts or identify new patterns. Data Visualizationis seen by numerous orders as a cutting edge likeness visual correspondence. It includes the creation and investigation of the visual portrayal.
What is data visualization?
Data visualization is the process of translating large data sets and metrics into charts, graphs and other visuals. The resulting visual representation of data makes it easier to identify and share real-time trends, outliers, and new insights about the information represented in the data.

A dashboard is an information visualization tool. It helps you monitor events or activities at a glance by providing insights on one or more pages or screens. Unlike an infographic, which presents a static graphical representation, a dashboard conveys real-time information by pulling complex data points directly from large data sets. An interactive dashboard makes it easy to sort, filter, or drill into different types of data as needed. Data science techniques can be used to identify what is happening, why it's happening, and what will happen next at speed.

As the amount of big data increases, more people are using data visualization tools to access insights on their computer and on mobile devices. Dashboards are used by business people, data analysts, and data scientists to make data-driven business decisions.

5 Types of Data Visualization

We have data or sort of information in the form of numbers, or statistics, there is always a story lying behind numbers, visualizing that statistics brings creation in them.

Presenting and visualizing data accurately establish trust between you and your viewers, let’s have a gaze at how to select the most authentic and likeable approach to visualize data;

1. Bar Graphs:

If you want to analyze data over time or the data is assembled in multiple categories such as various industries, variety of food, the progress of a company in the past 5 years, etc, a Bar Graph is the best choice with some characteristics or some kinds of careful suggestions.

In order to make bar graph more effectively and easy to read, outline includes orders of the bars should be chronological, fix time frames label at one axis and label other quantities on other axes, data should not be placed from most to least or least to most but must be in chronology.

Bar graphs include data in the form of multiple categories, we can either make individual graphs for each and every category or keep it in a single form through including multiple bars as one for each category at each time label. These bars could be assigned side by side or accumulated on top of each other.

If the dataset is arranged into multiple categories but isn’t confined in time, we could use the bars’ order from most to least or least to most. This arrangement helps the viewers to get a conclusion easily.
various types of data visualization are provided in the image that are bar graphs, line graphs, pie graphs, quantograms, typography.

Types of data visualization

2. Line Chart:

Line graphs are also used for presenting data over time or classified data by category as bar graphs. The only difference is that line graphs allow for refinement.

If you want to present data over very long time periods or continuously changing data, the line graph could be a solid choice to consider.

Most of the time it happens, we clearly don't know how to fill data accurately in the time duration for which data is available, in that condition we are drawing nothing other than a straight line.

Though the rate of progress or decay between time duration is not linear up to a remarkable extent, so line graphs must be used very delicately to avoid malformation of data.

3. Pie chart:

It is a presentation of data visualization in the circular form or circular chart. It is one of the most popular forms of data visualization, it can only be used when a smart portion of data add up to a whole.

For example, 40 % of the marks are considered to pass in an exam, which could be displayed in the pie chart as it is indicating to 40 % out of the total 100 % of the marks.

We can convert the percentage to proportions or proportions to the percentage for this aim, additionally, circle charts cannot be used to show an increase or decrease on their own.

In case, if a pie chart could be used to present the data over time, there is a need to make a new chart for each time period and every measurement and display them together for comparison.

4. Quantagrams:

The repeated pictogram or icon representation to show quantity is termed as Quantagrams, such that

A very common example to show the multi-character quantities using Quantagrams is the number of people. You must have seen Quantagrams as classic male and female icons at the doors of the restroom. This technique is suitable for small numbers, small percentages or proportions.

If we talk about pictograms, they are so simple and feel sound or reductive if they get used for any severe issues or a large quantity. It would appear as minimized if a severe issue is represented with simple sorted icons. We can opt Typography if we need to visualize data for large statistics.

5. Typography:

It is limited to certain cases where it can be accepted as the best solution provider, it is not restricted to provide an old text-only solution, instead, it is intelligently used to achieve a successful and effective piece of content.

The data would be fit for typography if it is large or greater than 100, never be a percentage of a whole or increase or decrease in percentage, and can't be compared to another number.

In order to improve typography visualization, it can be combined with a pictogram or icon that gives the viewer a clear visual picture with the context of the subject matter of data and numbers.

Data pre-processing

It is the process of transforming data before feeding it into the algorithm. It is utilized to change over crude information into a clean data set. it is an information mining strategy that includes moving crude information into a justifiable organization.

Data preprocessing is the process of transforming raw data into an understandable format. It is also an important step in data mining as we cannot work with raw data. The quality of the data should be checked before applying machine learning or data mining algorithms.

Why is Data preprocessing important?
Preprocessing of data is mainly to check the data quality. The quality can be checked by the following

Accuracy: To check whether the data entered is correct or not.
Completeness: To check whether the data is available or not recorded.
Consistency: To check whether the same data is kept in all the places that do or do not match.
Timeliness: The data should be updated correctly.
Believability: The data should be trustable.
Interpretability: The understandability of the data.
Major Tasks in Data Preprocessing:
Data cleaning
Data integration
Data reduction
Data transformation
Data preprocessing

Data cleaning:
Data cleaning is the process to remove incorrect data, incomplete data and inaccurate data from the datasets, and it also replaces the missing values. There are some techniques in data cleaning

Handling missing values:
Standard values like “Not Available” or “NA” can be used to replace the missing values.
Missing values can also be filled manually but it is not recommended when that dataset is big.
The attribute’s mean value can be used to replace the missing value when the data is normally distributed
wherein in the case of non-normal distribution median value of the attribute can be used.
While using regression or decision tree algorithms the missing value can be replaced by the most probable
value.
Noisy:Noisy generally means random error or containing unnecessary data points. Here are some of the methods to handle noisy data.

Binning: This method is to smooth or handle noisy data. First, the data is sorted then and then the sorted values are separated and stored in the form of bins. There are three methods for smoothing

data in the bin. Smoothing by bin mean method: In this method, the values in the bin are replaced by the mean value of the bin; Smoothing by bin median: In this method, the values in the bin are replaced by the median value; Smoothing by bin boundary: In this method, the using minimum and maximum values of the bin values are taken and the values are replaced by the closest boundary value.
Regression: This is used to smooth the data and will help to handle data when unnecessary data is present. For the analysis, purpose regression helps to decide the variable which is suitable for our analysis.

Clustering: This is used for finding the outliers and also in grouping the data. Clustering is generally used in unsupervised learning.

Data integration:The process of combining multiple sources into a single dataset. The Data integration process is one of the main components in data management. There are some problems to be considered during data integration.

Schema integration: Integrates metadata(a set of data that describes other data) from different sources.

Entity identification problem: Identifying entities from multiple databases. For example, the system or the use should know student _id of one database and student_name of another database belongs to the same entity.

Detecting and resolving data value concepts: The data taken from different databases while merging may differ. Like the attribute values from one database may differ from another database. For example, the date format may differ like “MM/DD/YYYY” or “DD/MM/YYYY”.

Data reduction:This process helps in the reduction of the volume of the data which makes the analysis easier yet produces the same or almost the same result. This reduction also helps to reduce storage space. There are some of the techniques in data reduction are Dimensionality reduction, Numerosity reduction, Data compression.

Dimensionality reduction: This process is necessary for real-world applications as the data size is big. In this process, the reduction of random variables or attributes is done so that the dimensionality of the data set can be reduced. Combining and merging the attributes of the data without losing its original characteristics. This also helps in the reduction of storage space and computation time is reduced. When the data is highly dimensional the problem called “Curse of Dimensionality” occurs.

Numerosity Reduction: In this method, the representation of the data is made smaller by reducing the volume. There will not be any loss of data in this reduction.
Data compression: The compressed form of data is called data compression. This compression can be lossless or lossy. When there is no loss of information during compression it is called lossless compression. Whereas lossy compression reduces information but it removes only the unnecessary information.

Data Transformation:The change made in the format or the structure of the data is called data transformation. This step can be simple or complex based on the requirements. There are some methods in data transformation.

Smoothing: With the help of algorithms, we can remove noise from the dataset and helps in knowing the important features of the dataset. By smoothing we can find even a simple change that helps in prediction.
Aggregation: In this method, the data is stored and presented in the form of a summary. The data set which is from multiple sources is integrated into with data analysis description. This is an important step since the accuracy of the data depends on the quantity and quality of the data. When the quality and the quantity of the data are good the results are more relevant.
Discretization: The continuous data here is split into intervals. Discretization reduces the data size. For example, rather than specifying the class time, we can set an interval like (3 pm-5 pm, 6 pm-8 pm).
Normalization: It is the method of scaling the data so that it can be represented in a smaller range. Example ranging from -1.0 to 1.0.

Result

To achieve the results, various data mining techniques are utilized in python language. Various factors which affects the house pricing are considered and further worked upon them. Machine learning has been considered to complete out the desired task. Firstly, data collection is performed. Then data cleaning is performed to remove all the errors from the data and make clean. Then data pre-processing is done.

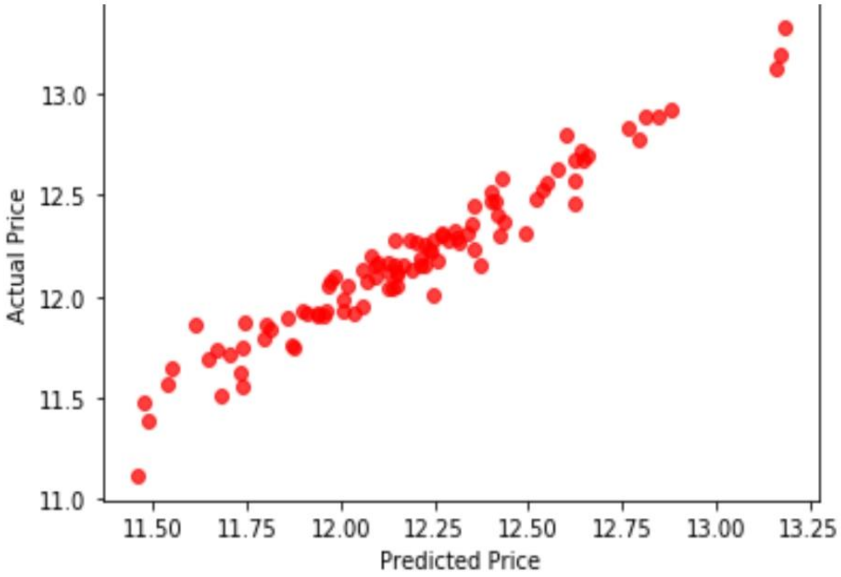


Fig. 10

Conclusion

The sales price for the houses are calculated using different algorithms. The sales prices have been calculated with better accuracy and precision. This would be of great help for the people. To achieve these results, various data mining techniques are utilized in python language. The various factors which affect the house pricing should be considered and work upon them. Machine learning has assisted to complete out task. Firstly, the data collection is performed. Then data cleaning is carried out to remove all the errors from the data and make it clean. Then the data pre-processing is done. Then with help of data visualization, different plots are created. This has depicted the distribution of data in different forms. Further, the preparation and testing of the model are performed. It has been found that some of the classification algorithms were applied on our dataset while some were not. So, those algorithms which were not being applied on our house pricing dataset are dropped and tried to improve the accuracy.

References

[1].Jain, N., Kalra, P., &Mehrotra, D. (2019). Analysis ofFactors AffectingInfant Mortality RateUsing Decision Treein RLanguage.In Soft Computing:Theories andApplications(pp.639-646).Springer,Singapore.

[2] R. A. Rahadi, S. K. Wiryono, D. P. Koesrindartotoor,and I.B.Syamwil, ―Factors influencingthepriceofhousing in Indonesia,‖Int. J.Hous.Mark.Anal.,vol.8,no.2,pp.169–188,2015

[3]V.Limsombunchai,―House price prediction: Hedonic price model vs.artificialneuralnetwork,‖Am.J....,2004

[4] Kadir, T.,&Gleeson, F. (2018). Lung cancer prediction using machine learning and advanced imagingtechniques.Translational Lung CancerResearch,7(3),304-312.

[5]Liu, J.,Ye,Y., Shen,C.,Wang,Y.,&Erdélyi,R.(2018). ANewTool for CMEArrival Time Prediction using MachineLearning Algorithms:CATPUMA.TheAstrophysical

[6] House Price Index. Federal Housing Finance Agency. https://www.fhfa.gov/ (accessed September 1, 2019).
Google Scholar

[7] Fan C, Cui Z, Zhong X. House Prices Prediction with Machine Learning Algorithms. Proceedings of the 2018 10th International Conference on Machine Learning and Computing ICMLC 2018. doi:10.1145/3195106.3195133.
Google Scholar

[8] Phan TD. Housing Price Prediction Using Machine Learning Algorithms: The Case of Melbourne City, Australia. 2018 International Conference on Machine Learning and Data Engineering (ICMLDE) 2018. doi:10.1109/icmlde.2018.00017.
Google Scholar

[9] Mu J, Wu F, Zhang A
Housing Value Forecasting Based on Machine Learning Methods
Abstract and Applied Analysis, 2014 (2014), pp. 1-7
doi:10.1155/2014/648047.
View PDFCrossRefGoogle Scholar

[10] Lu S, Li Z, Qin Z, Yang X, Goh RSM. A hybrid regression technique for house prices prediction. 2017 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM) 2017. doi:10.1109/ieem.2017.8289904.
Google Scholar

[11] Ivanov I. vecstack. GitHub 2016. https://github.com/vecxoz/vecstack (accessed June 1, 2019). [Accessed: 01-June-2019].
Google Scholar

[12] Wolpert D H
Stacked generalization

Neural Networks, 5 (1992), pp. 241-259
doi:10.1016/s0893-6080(05)80023-1.
ArticleDownload PDFView Record in ScopusGoogle Schola

[13] Qiu Q. Housing price in Beijing. Kaggle 2018. https://www.kaggle.com/ruiqurm/lianjia/ (accessed June 1, 2019).
Google Scholar

[14] Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, et al.
Scikit-learn: Machine Learning in Python
The Journal of Machine Learning Research, 12 (2011), pp. 2825-2830
View PDFView Record in ScopusGoogle Scholar

[15] Breiman L. Random Forests. SpringerLink. https://doi.org/10.1023/A:1010933404324 (accessed September 11, 2019).
Google Scholar

[16] Raschka S, Mirjalili V.
Python machine learning: Machine learning and deep learning with Python, scikit-learn, and TensorFlow (2nd ed.), Packt Publishing, Birmingham (2017)
Google Schol