VISVESVARAYA TECHNOLOGICAL UNIVERSITY

Jnana Sangama, Belagavi-590018

An Internship Report

on

"HOUSE RENT PRICE PREDICTION USING MACHINE LEARNING "

Submitted in partial fulfillment of the requirements for the award of degree

of

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE AND ENGINEERING

By

AVINASH KUMAR SINGH **1EP16CS017**

Internal Guide	**External Guide**
Mrs. Rashmi T.V	**Mr. Ruthvik S**
Assistant Professor	**Technical Facilitator**
Dept. of CSE, EPCET	**YJ Infotech Solutions**

EAST POINT COLLEGE OF ENGINEERING & TECHNOLOGY

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

Jnana Prabha, Bidarahalli, Virgo Nagar Post, Bengaluru, Karnataka 560049

2019-2020

EAST POINT COLLEGE OF ENGINEERING & TECHNOLOGY

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

Jnana Prabha, Bidarahalli, Virgo Nagar Post, Bengaluru, Karnataka 560049

CERTIFICATE

This is to certify that the Internship work entitled “**HOUSE RENT PRICE PREDICTION USING MACHINE LEARNING**” is a bonafide work carried out by **AVINASH KUMAR SINGH [1EP16CS017],** in partial fulfillment of the requirements of **BACHELOR OF ENGINEERING** in **COMPUTER SCIENCE AND ENGINEERING** in **VISVESVARAYA TECHNOLOGICAL UNIVERSITY, Belgaum,** during the year 2019-2020. It is certified that corrections/suggestions recommended have been incorporated in the report.

Internal Guide
Mrs. Rashmi T.V
Assistant Prof, Dept. of CSE
EPCET, Bangalore

External Guide
Mr. Ruthvik S
Technical Facilitator
YJ Infotech Solutions

Signature of HOD
Prof. Nityananda C R
HOD, Dept. of CSE,
EPCET, Bangalore

Signature of Principal
Dr. Prakash S
Principal,
EPCET, Bangalore

GSTIN : 29AAACY9186E1ZT CIN : U72900KA2017PTC105673

YJ INFOTECH SOLUTIONS PVT. LTD.

56A, 3rd Floor, 24th Main, 3rd Cross, J.P. Nagar, 2nd Phase, Bangalore - 560 078.
Website : www.yjinfotech.co.in | Ph : 080-48909790 Mob : 8892062683

CERTIFICATE OF MERIT

This is certify that **Mr. AVINASH KUMAR SINGH (Reg No. 1EP16CS017)** has successfully completed the internship in **DATA SCEINCE AND MACHINE LEARNING** Application Development in Our concern from **11th Jul 2019** to **11th Aug 2019.**

During the internship period, the performance of the intern was found to be **GOOD.**

YJ INFOTECH SOLUTIONS PVT. LTD. 080-48909790

Rutwik
Program Coordinator

HR Head

ACKNOWLEDGEMENT

I thank the **Management and Principal of East Point College of Engineering and Technology, Bangalore** for providing me an opportunity to work on this internship project.

I would like to express my heartfelt thanks to **Mr. Nithyananda C R**, Professor and Head of Department of Computer Science and Engineering, EPCET for his valuable advice and encouragement to me in completing this internship.

I am obliged to **Mrs. Rashmi T.V** , Assistant Professor, Dept. of CSE, who rendered valuable assistance as the Internship Guide.

I would like to thank **Mr. Ruthvik S** Technical Facilitator, Azure Skynet Solutions Pvt. Ltd., for providing with technical support and knowledge.

I would like to thank my **Parents** and **Friends** for their support and encouragement during the course of my Internship. Finally, I offer my regards to all the faculty members of the CSE Department and all those who supported me in any respect during the internship.

AVINASH KUMAR SINGH [1EP16CS017]

ABSTRACT

House Rent prices can be hard to understand due to the lack of information and the complexities of the real estate market. Airbnb is a platform for marketing and renting properties for short-term stays. Airbnb listings contain a lot of information that can help predict the price associated with that list. The relationship between Airbnb listing prices and average real estate prices in the vicinity of these listings is explored. Multiple Machine Learning algorithms are considered: Linear Regression, Support Vector Machines, K-Nearest Neighbor, Decision Tree, Naive Bayes, and Stochastic Gradient Descent. Finally, each of these algorithms are ranked by effectiveness and considered both with and without real estate estimates as features.

CONTENTS

LIST OF FIGURES

CHAPTER 1

ORGANIZATION INFORMATION

YJ INFOTECH Solutions Pvt. Ltd. is an executive and technical search firm that emphasizes strategic placement of experienced professionals, team leaders and support staff. Our mission is to help progressive companies acquire top-notch talent in IT Industry, Non-IT Industry, Engineering Industry, etc.

Fig: 1.1 Company Logo.

SERVICES

The services provided by the company are-

TRAINING: They evolve and conduct the training in the holistic customized and tailor made modules catering to the need of trainees.
RECRUITMENT/PLACEMENTS: They extend their professional services to the career aspirant's at all high levels of the organization.

HR CONSULTING: 15+ years of fruitful, ethical, professional and sincere service in handling the Human Resources.

COMPLIANCE MANAGEMENT SERVICES: They provide Digital Solutions and Automated Manage Services in IT Technologies like:

- Web application development
- Hadoop- Big data
- Raspberry pie- IOT
- Python – Big data
- R programming – machine learning
- Cloud computing – Big data

CHAPTER 2

DEPARTMENT PROFILE

2.1 Machine Learning:

Machine learning (ML) is the study of computer algorithms that improve automatically through experience. It is seen as a subset of artificial intelligence. Machine learning algorithms build a mathematical model based on sample data, known as "training data", in order to make predictions or decisions without being explicitly programmed to do so. Machine learning algorithms are used in a wide variety of applications, such as email filtering and computer vision, where it is difficult or infeasible to develop conventional algorithms to perform the needed tasks.

Machine learning is closely related to computational statistics, which focuses on making predictions using computers. The study of mathematical optimization delivers methods, theory and application domains to the field of machine learning. Data mining is a related field of study, focusing on exploratory data analysis through unsupervised learning. In its application across business problems, machine learning is also referred to as predictive analytics.

Supervised Learning and Unsupervised Learning:

In **Supervised learning**, you train the machine using data which is well labelled. It means some data is already tagged with the correct answer. It can be compared to learning which takes place in the presence of a supervisor or a teacher.

A **supervised learning** algorithm learns from labelled training data, helps you to predict outcomes for unforeseen data. Successfully building, scaling, and deploying accurate supervised machine learning Data science model takes time and technical expertise from a team of highly skilled data scientists. Moreover, Data scientist must rebuild models to make sure the insights given remains true until its data changes.

Supervised learning is a machine learning technique, where you do not need to supervise the model. Instead, you need to allow the model to work on its own to discover information. It mainly deals with the unlabelled data.

Unsupervised learning algorithms allow you to perform more complex processing tasks compared to supervised learning. Although, unsupervised learning can be more unpredictable compared with other natural learning deep learning and reinforcement learning methods.

CHAPTER 3

METHODS AND TECHNOLOGIES LEARNT

3.1 Technologies learnt:

Python

Python is an interpreted, high-level, general-purpose programming language. Created by Guido van Rossum and first released in 1991, Python's design philosophy emphasizes code readability with its notable use of significant whitespace. Its language constructs and 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. Python is often described as a "batteries included" language due to its comprehensive standard library.

Python was conceived in the late 1980s as a successor to the ABC language. Python 2.0, released in 2000, introduced features like list comprehensions and a garbage collection system capable of collecting reference cycles. Python 3.0, released in 2008, was a major revision of the language that is not completely backward-compatible, and much Python 2 code does not run unmodified on Python 3.

The Python 2 language, i.e. Python 2.7.x, was officially discontinued on 1 January 2020 (first planned for 2015) after which security patches and other improvements will not be released for it. With Python 2's end-of-life, only Python 3.5.xand later are supported.

Python interpreters are available for many operating systems. A global community of programmers develops and maintains CPython, an source reference. A non-profit organization, the Python Software Foundation, manages and directs resources for Python and CPython development.

Features of python:

Easy to code

Python is high level programming language. Python is very easy to learn language as compared to other language like c, c#, java script, java etc. It is very easy to code in python language and anybody can learn python basic in few hours or days. It is also developer-friendly language.

Free and Open Source

Python language is freely available at official website. Since, it is open-source, this means that source code is also available to the public. So, you can download it as, use it as well as share it.

Object oriented Language

One of the key features of python is Object-Oriented programming. Python supports object-oriented language and concepts of classes, objects encapsulation etc.

GUI Programming Support

Graphical Users interfaces can be made using a module such as PyQt5, PyQt4, python or Tk in python. PyQt5 is the most popular option for creating graphical apps with Python.

High-Level Language

Python is a high-level language. When we write programs in python, we do not need to remember the system architecture, nor do we need to manage the memory.

Extensible feature

Python is an Extensible language. we can write our python code into C or C++ language and also, we can compile that code in C/C++ language.

Python is Portable language

Python language is also a portable language. For example, if we have python code for windows and if we want to run this code on other platform such as Linux, UNIX and Mac then we do not need to change it, we can run this code on any platform.

Python is integrated language

Python is also an integrated language because we can easily integrated python with other language like C, C++ etc.

Interpreted Language

Python is an Interpreted Language. because python code is executed line by line at a time. Like other language C, C++, java etc there is no need to compile python code this makes it easier to debug our code. The source code of python is converted into an immediate form called byte code.

Large Standard Library

Python has a large standard library which provides rich set of module and functions so you do not have to write your own code for every single thing. There are many libraries present in python for such as regular expressions, unit-testing, web browsers etc.

Dynamically Typed Language

Python is dynamically-typed language. That means the type (for example- int, double, long etc) for a variable is decided at run time not in advance. Because of this feature we don't need to specify the type of variable.

Anaconda (Python Distribution):

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 includes data-science packages suitable for Windows, Linux, and mac OS.

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.

Anaconda Cloud

Anaconda Cloud is a package management service by Anaconda where you can find, access, store and share public and private notebooks, environments, and conda and PyPI packages. Cloud hosts useful Python packages, notebooks and environments for a wide variety of applications. You do not need to log in or to have a Cloud account, to search for public packages, download and install them.

You can build new packages using the Anaconda Client command line interface (CLI), then manually or automatically upload the packages to Cloud.

The following applications are available by default in Navigator

- JupyterLab
- Jupyter Notebook
- QtConsole
- Spyder
- Glue
- Orange
- RStudio
- Visual Studio Code

Python used in Machine Learning:

Simple and consistent

Python offers concise and readable code. While complex algorithms and versatile workflows stand behind machine learning and AI, Python's simplicity allows developers to write reliable systems. Developers get to put all their effort into solving an ML problem instead of focusing on the technical nuances of the language.

Additionally, Python is appealing to many developers as it's easy to learn. Python code is understandable by humans, which makes it easier to build models for machine learning.

Extensive selection of libraries and frameworks

Implementing AI and ML algorithms can be tricky and requires a lot of time. It's vital to have a well-structured and well-tested environment to enable developers to come up with the best coding solutions.

To reduce development time, programmers turn to a number of Python frameworks and libraries. A software library is pre-written code that developers use to solve common programming tasks. Python, with its rich technology stack, has an extensive set of libraries for artificial intelligence and machine learning. Here are some of them:

- Keras, TensorFlow, and Scikit-learn for machine learning
- NumPy for high-performance scientific computing and data analysis
- SciPy for advanced computing
- Pandas for general-purpose data analysis
- Matplotlib for data visualization
- Pydotplus for Graph/Tree visualization

Platform independence

Platform independence refers to a programming language or framework allowing developers to implement things on one machine and use them on another machine without any (or with only minimal) changes. One key to Python's popularity is that it's a platform independent language. Python is supported by many platforms including Linux, Windows, and macOS. Python code can be used to create standalone executable programs for most common operating systems, which means that Python software can be easily distributed and used on those operating systems without a Python interpreter.

What's more, developers usually use services such as Google or Amazon for their computing needs. However, you can often find companies and data scientists who use their own machines with powerful Graphics Processing Units (GPUs) to train their ML models. And the fact that Python is platform independent makes this training a lot cheaper and easier.

3.2 Non-Technical Outcomes:

3.2.1 Soft Skills Development

Soft skills are a combination of people skills, social skills, communication skills, character or personality traits, attitudes, career attributes, social intelligence and emotional intelligence quotients, among others, that enable people to navigate their environment, work well with others, perform well, and achieve their goals with complementing hard skills.

The Collins English Dictionary defines the term "soft skills" as "desirable qualities for certain forms of employment that do not depend on acquired knowledge: they include common sense, the ability to deal with people, and a positive flexible attitude."

3.2.2 Basic Grammar

English grammar is the way in which meanings are encoded into wordings in the English language. This includes the structure of words, phrases, clauses, and sentences, right up to the structure of whole texts.

3.2.3 Presentation Skills

Presentation skills are the skills we need in delivering effective and engaging presentations to a variety of audiences. These skills cover a variety of areas such as the structure of your presentation, the design of our slides, the tone of our voice and the body language we convey.

3.2.4 Reading Skills

Reading skills is the ability of an individual to read, comprehend and interpret written words on a page of an article or any other reading material. The possession of a good reading skill will enable the individual to be able to assimilate a written work within a short period while reading. If an individual develops a reading skill, it is a lifelong activity. And while reading at any given time the individual is expected to also think critically on the particular topic or subject to understand the point of the writer.

3.2.5 Communication Skills

Communication skills are abilities we use when giving and receiving different kinds of information. Some examples include communicating ideas, feelings or what's happening around us. Communication skills involve listening, speaking, observing and empathizing. It is also helpful to understand the differences in how to communicate through face-to-face interactions, phone conversations and digital communications, like email and social media.

3.2.6 Personality Development

Personal development covers activities that improve awareness and identity, develop talents and potential, build human capital and facilitate employability, enhance the quality of life and contribute to the realization of dreams and aspirations. Personal development takes place over the course of a person's entire life. Not limited to self-help, the concept involves formal and informal activities for developing others in roles such as teacher, guide, counselor, manager, life coach or mentor. When personal development takes place in the context of institutions, it refers to the methods, programs, tools, techniques, and assessment systems that support human development at the individual level in organizations.
Among other things, personal development includes the following activities:

Improving self-awareness
Improving self-knowledge
Improving skills and/or learning new ones
Building or renewing identity/self-esteem
Developing strengths or talents
Improving a career
Identifying or improving potential
Building employability or (alternatively) human capital

3.2.7 Team Building Activities

Team building activities that require coworkers to work together to solve problems can improve the ability to think rationally and strategically. Overall, team building in the workplace enables better communication, better relationships and ultimately increases productivity.

CHAPTER 4

OUTCOMES OF INTERNSHIP

The Project collect our input data from two sources: Inside Airbnb and Zillow. From Inside Airbnb, we get multiple datasets, in the form of .csv files, where each row is a different Airbnb listing. After processing the datasets, we are left with 59,932 listings, each with id, neighborhood, city, state, property type, room type, number of accommodates, number of bathrooms, number of bedrooms, number of beds, bed type and price.

From Zillow, we get a dataset containing a time series estimates for the average rent price for a Single Family Residence for most zip codes in the US. Yet, for this research, we don't need an entire time series worth of estimates. Instead, we need a single value per zip code.

In order to get that single value, we consider the upload date of each Airbnb dataset and use that upload date as the canonical date for all of the listings in that dataset. For each zip code, we either extract the value at that date or, if missing, consider the closest value to that date. In full, in order to model the "price" target feature, we consider the following features:

- Property Type (categorical)
- Room Type (categorical)
- Number of Accommodates (continuous)
- Number of Bathrooms (continuous)
- Number of Bedrooms (continuous)
- Number of Beds (continuous)
- Bed Type (categorical)
- Average Rent (continuous)

We believe that the application of machine learning algorithms on these features will yield improved results.

Packages used in the project:

Matplotlib: Matplotlib is a plotting library for the Python programming language and its numerical mathematics extension NumPy. It provides an object-oriented API for embedding plots into application using general-purpose GUI toolkits.

Pandas: It is a high-level data manipulation tool developed by Wes McKinney. It is built on the Numpy package and its key data structure is called the Data Frame. Data Frames allow you to store and manipulate tabular data in rows of observations and columns of variables.

Sklearn: Sklearn features various classification, clustering and regression algorithms. Scikit-learn is largely written in Python, and uses numpy extensively for high-performance linear algebra and array operations. Scikit-Learn contains a very large list of machine learning algorithms which can be easily interchanged.

The Machine Learning algorithms require two phases: a training phase and a testing phase.

Training Phase : The algorithms are fed with data points containing all of the aforementioned features including the target feature. After this process, each algorithm "learns" or bases its internal constants or processes on the data it was fed.

Testing Phase : It feed different data points, containing all of the aforementioned features omitting the target feature, to the trained algorithms in order to produce a prediction for the target feature. We then collect all of the predictions made by all of the algorithms and compare these predictions to the actual values.

Finally, we rank all of the algorithms based on how "close" their predictions are to the real value.

Furthermore, as regards the ranking of Machine Learning algorithms, we consider multiple regression metrics:

- Mean Absolute Error
- Mean Squared Error
- R2 Error

4.1 IMPLEMENTATION

The project has been fully implemented in Python using the popular Scikit-Learn library.

Implementation consists of three parts:

- Processing and preparing the data
- Training and storing the models and related output .
- Demo :The demo is a simple Python script that imports all the trained models and gets demo data from 'demo_data.csv' and uses this data to make predictions. This demo can have actual pragmatic uses as it can be used to generate a nightly rental price for a home/apartment that would like to go on Airbnb.

The following flow Diagram describes the process succinctly :

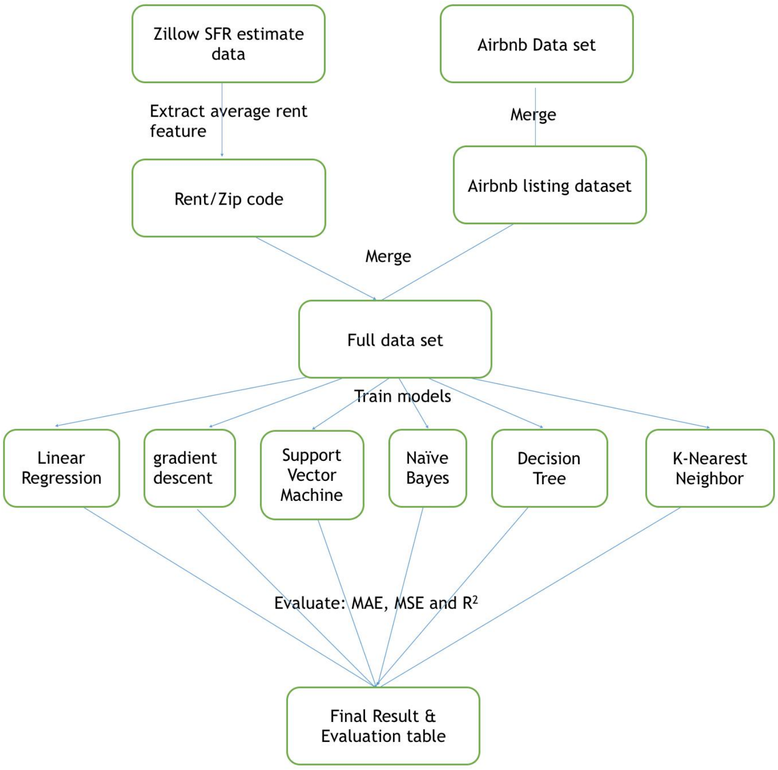
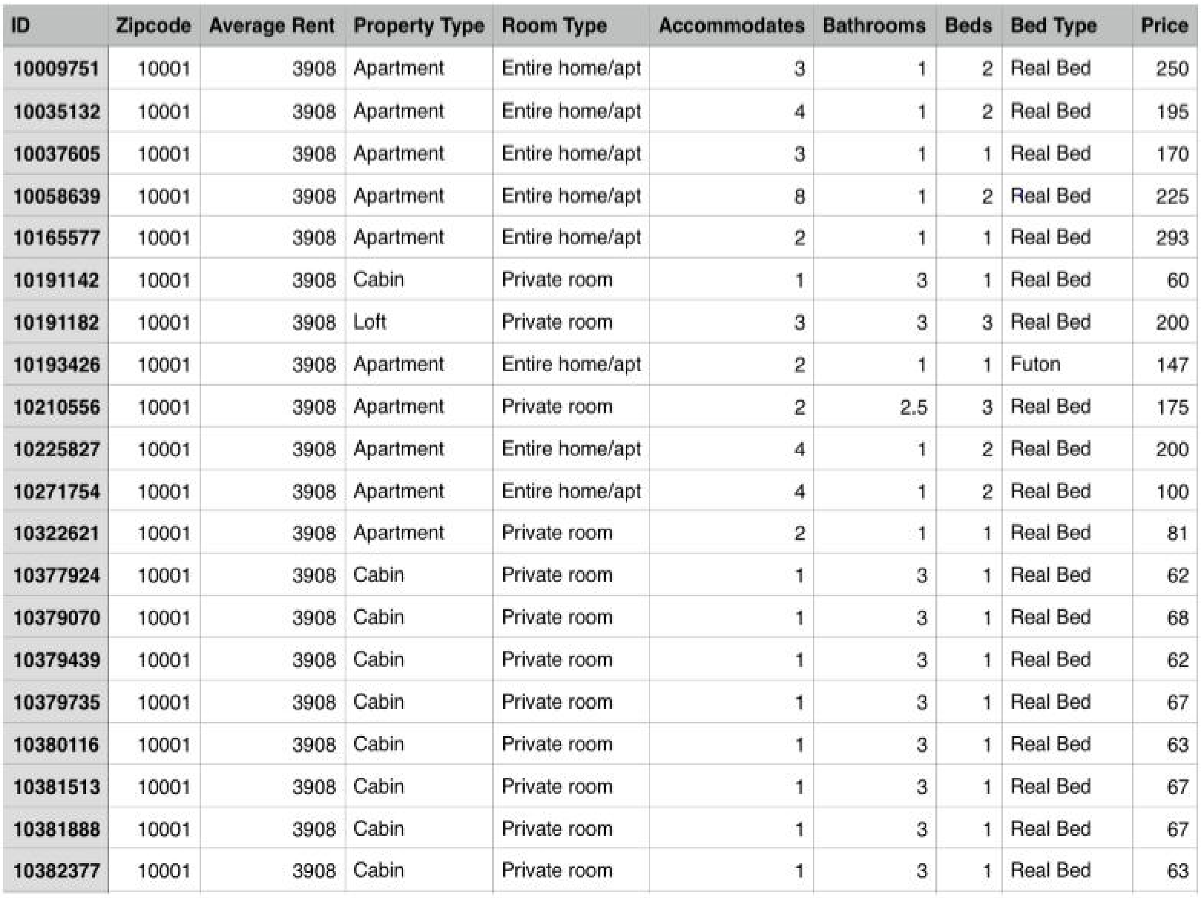


Fig 4.1 Flow Diagram

4.2 PROJECT EXECUTION:

4.2.1 Dataset:

It have 2 different datasets in the form of .csv files.

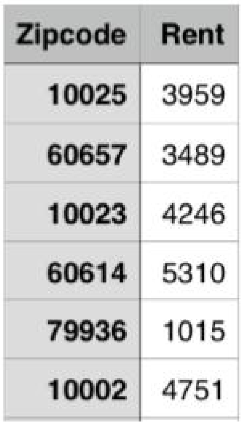


ID	Zipcode	Average Rent	Property Type	Room Type	Accommodates	Bathrooms	Beds	Bed Type	Price
10009751	10001	3908	Apartment	Entire home/apt	3	1	2	Real Bed	250
10035132	10001	3908	Apartment	Entire home/apt	4	1	2	Real Bed	195
10037605	10001	3908	Apartment	Entire home/apt	3	1	1	Real Bed	170
10058639	10001	3908	Apartment	Entire home/apt	8	1	2	Real Bed	225
10165577	10001	3908	Apartment	Entire home/apt	2	1	1	Real Bed	293
10191142	10001	3908	Cabin	Private room	1	3	1	Real Bed	60
10191182	10001	3908	Loft	Private room	3	3	3	Real Bed	200
10193426	10001	3908	Apartment	Entire home/apt	2	1	1	Futon	147
10210556	10001	3908	Apartment	Private room	2	2.5	3	Real Bed	175
10225827	10001	3908	Apartment	Entire home/apt	4	1	2	Real Bed	200
10271754	10001	3908	Apartment	Entire home/apt	4	1	2	Real Bed	100
10322621	10001	3908	Apartment	Private room	2	1	1	Real Bed	81
10377924	10001	3908	Cabin	Private room	1	3	1	Real Bed	62
10379070	10001	3908	Cabin	Private room	1	3	1	Real Bed	68
10379439	10001	3908	Cabin	Private room	1	3	1	Real Bed	62
10379735	10001	3908	Cabin	Private room	1	3	1	Real Bed	67
10380116	10001	3908	Cabin	Private room	1	3	1	Real Bed	63
10381513	10001	3908	Cabin	Private room	1	3	1	Real Bed	67
10381888	10001	3908	Cabin	Private room	1	3	1	Real Bed	67
10382377	10001	3908	Cabin	Private room	1	3	1	Real Bed	63

Fig 4.2.1 Dataset 1

The above figure is the snippet of the Full_Table dataset used for the prediction of House Rent price.This contains all the information necessary to train the models. This is a table containing every Airbnb listing in our set with 10 different columns all non-empty:

- ID
- Zipcode
- Average Rent (From Zillow)
- Property Type
- Room Type
- Accommodates
- Bathrooms
- Beds
- Bed Type



Zipcode	Rent
10025	3959
60657	3489
10023	4246
60614	5310
79936	1015
10002	4751

Figure 4.2.2 Dataset 2

The Dataset 2, ‘full_zipcode_rent.csv’, contains a table mapping almost all zip codes in the US to the average rent price for a Single Family Residence in that zip code.

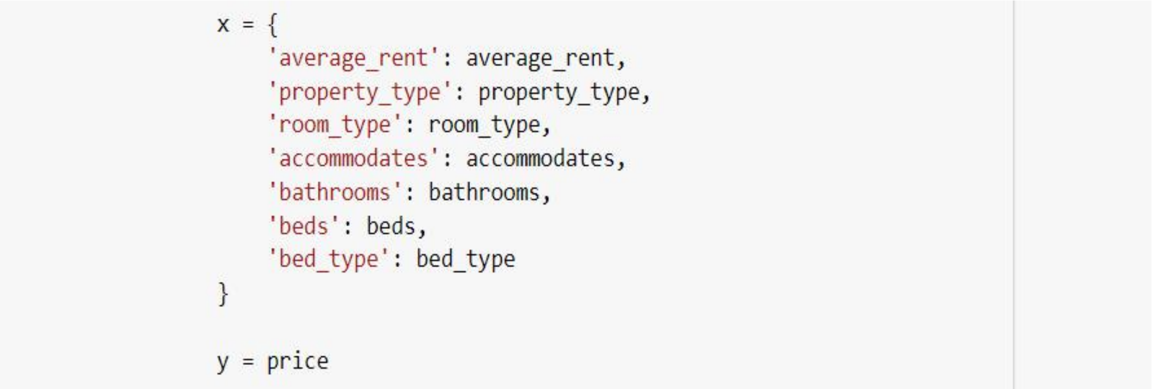
This is loaded on the jupyter notebook using the read_csv method,
data = pd.read_csv("Full_Table.csv")

The dataset is not having any string data and is already in the right format. No redundant or null values were found. Therefore, we proceed to use the data without any preprocessing. With the data processing done, the next part is the Machine Learning.

In order to generate these datasets, we start with a number of datasets that have to be joined together. First, we start with the datasets containing the Airbnb listings for each city/region in the set. In total, we have 16 datasets representing 16 different city/regions, each containing multiple zip codes. These datasets had to be programmatically merged into a single dataset.

This new, merged dataset is almost ‘full_table.csv’ except that it misses the Average Rent feature from Zillow.

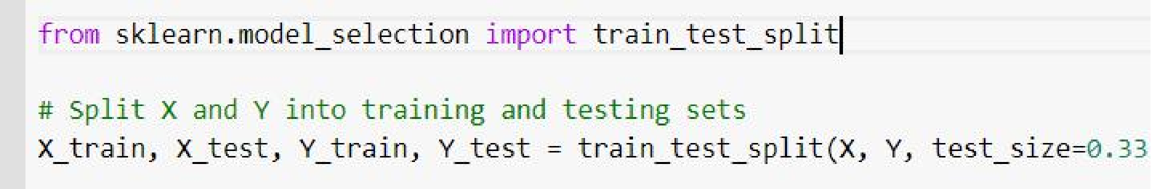
4.2.2 Shuffle and Split Data



```
x = {
    'average_rent': average_rent,
    'property_type': property_type,
    'room_type': room_type,
    'accommodates': accommodates,
    'bathrooms': bathrooms,
    'beds': beds,
    'bed_type': bed_type
}

y = price
```

Fig 4.3.1 Assigning X and Y

Now that we have the data fully gathered, we can now train each model. To do this, we first go through the 'full_table.csv' dataset and gather the data into two different lists 'X' and 'Y'. 'X' is a list containing all of the descriptive features whereas 'Y' is a list containing the target features.



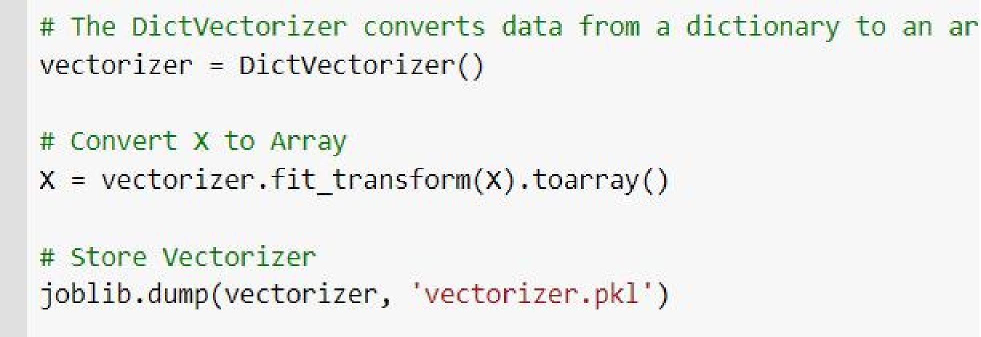
```
from sklearn.model_selection import train_test_split

# Split X and Y into training and testing sets
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.33
```

Fig 4.3.2 Splitting Dataset

The above figure we will take the dataset and split the data into training and testing subsets. Typically, the data is also shuffled into a random order when creating the training and testing subsets to remove any bias in the ordering of the dataset.

The benefit to splitting a dataset into some ratio of training and testing subsets for a learning algorithm is it is useful to evaluate our model once it is trained. To know if it has learned properly from a training split of the data.

4.2.3 Vectorize and Normalize the Dataset

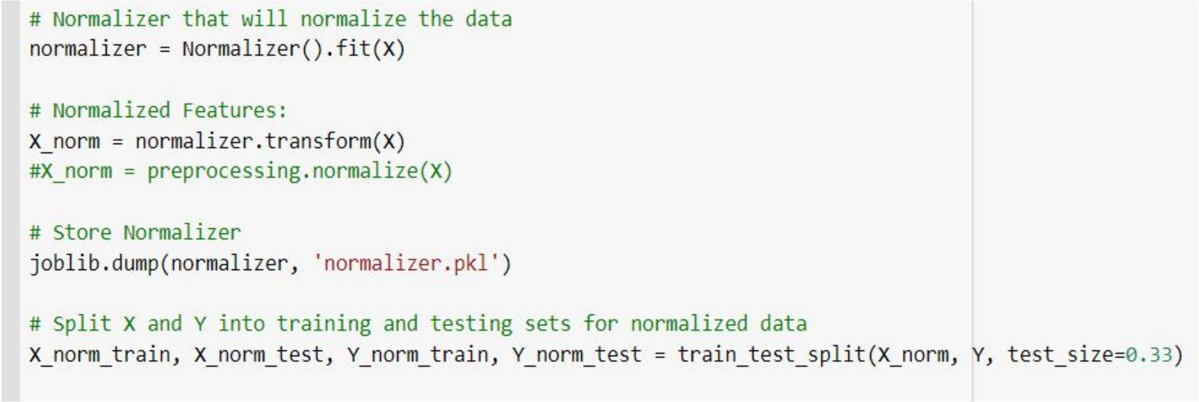


```
# The DictVectorizer converts data from a dictionary to an ar
vectorizer = DictVectorizer()

# Convert X to Array
X = vectorizer.fit_transform(X).toarray()

# Store Vectorizer
joblib.dump(vectorizer, 'vectorizer.pkl')
```

Fig 4.4.1 Vectorizer

Furthermore, the descriptive features are of different types—some are continuous and some are categorical—and, in order for the models to appropriately process this data, all of the data must be converted to continuous data.In the above figure, the code snippet for applying,

we use Scikit-Learn’s DictVectorizer class, which converts dictionaries into lists and back and also performs a one-hot encoding of the categorical features.



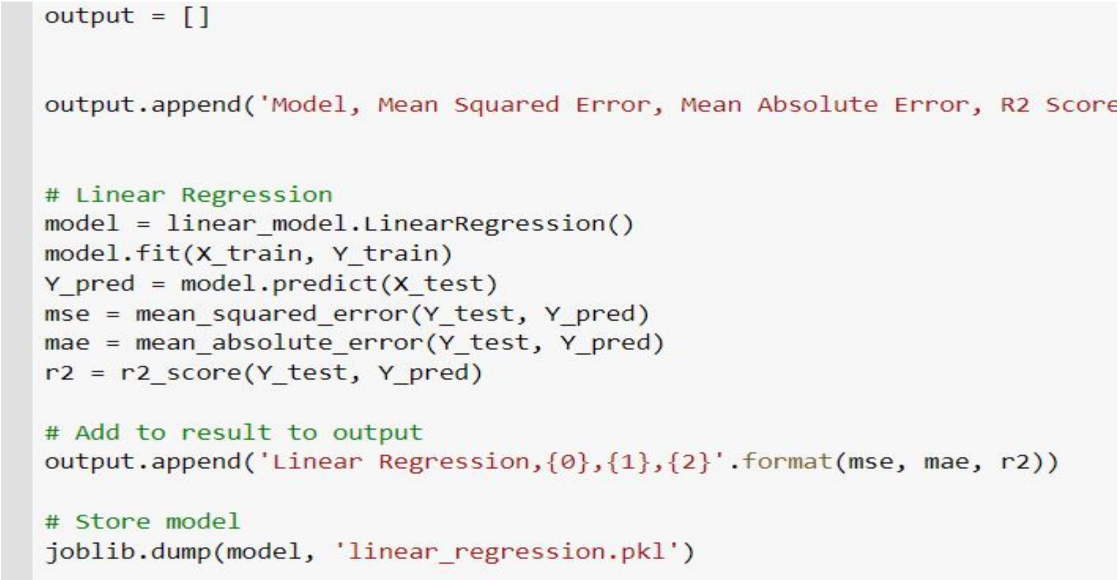
```
# Normalizer that will normalize the data
normalizer = Normalizer().fit(X)

# Normalized Features:
X_norm = normalizer.transform(X)
#X_norm = preprocessing.normalize(X)

# Store Normalizer
joblib.dump(normalizer, 'normalizer.pkl')

# Split X and Y into training and testing sets for normalized data
X_norm_train, X_norm_test, Y_norm_train, Y_norm_test = train_test_split(X_norm, Y, test_size=0.33)
```

Fig 4.4.2 Normalizing the data

The above code snippet we use Scikit-Learn’s Normalizer class to normalize the descriptive features. Normalization is only required for Gradient Descent and as such, we create a separate list called ‘X_norm’ that will hold the data normalized.

4.2.4 Applying multiple Algorithms

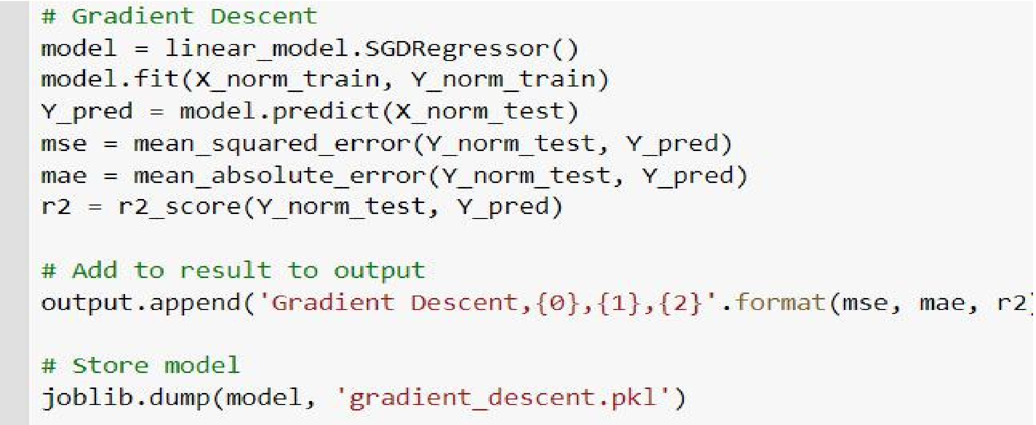
In our study, we decide to use multiple algorithms:



```
output = []


output.append('Model, Mean Squared Error, Mean Absolute Error, R2 Score


# Linear Regression
model = linear_model.LinearRegression()
model.fit(X_train, Y_train)
Y_pred = model.predict(X_test)
mse = mean_squared_error(Y_test, Y_pred)
mae = mean_absolute_error(Y_test, Y_pred)
r2 = r2_score(Y_test, Y_pred)

# Add to result to output
output.append('Linear Regression,{0},{1},{2}'.format(mse, mae, r2))

# Store model
joblib.dump(model, 'linear_regression.pkl')
```

Fig 4.5.1 Linear Regression

In the above snippet we are applying Linear Regression ,the goal is to minimize the sum of the squared errors to fit a straight line to a set of data points.And the results of Mean squarred error , mean Absolute error and R2 score with stored in Output.

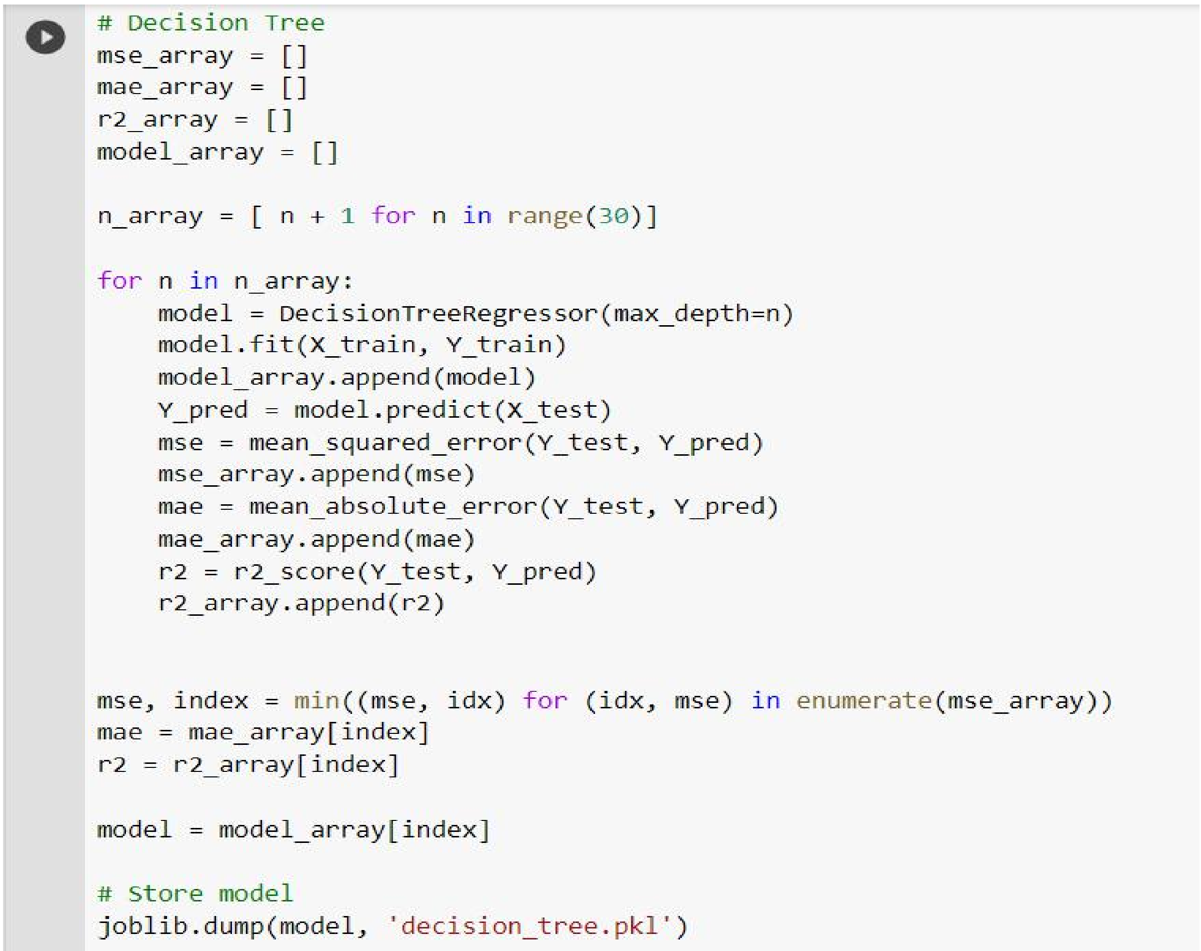


```
# Gradient Descent
model = linear_model.SGDRegressor()
model.fit(X_norm_train, Y_norm_train)
Y_pred = model.predict(X_norm_test)
mse = mean_squared_error(Y_norm_test, Y_pred)
mae = mean_absolute_error(Y_norm_test, Y_pred)
r2 = r2_score(Y_norm_test, Y_pred)

# Add to result to output
output.append('Gradient Descent,{0},{1},{2}'.format(mse, mae, r2

# Store model
joblib.dump(model, 'gradient_descent.pkl')
```

Fig 4.5.2 Gradient Descent

This above code snippet is using Stochastic gradient descent, also known as Incremental gradient descent algorithms . Stochastic gradient descent has been successfully applied to large-scale and sparse machine learning problems often encountered in Machine Learning. And the results of Mean squared error , mean Absolute error and R2 score with stored in Output.

Similarly we will repeat the same for Support Vector Machine and Naive Bayes Algorithms.



```
# Decision Tree
mse_array = []
mae_array = []
r2_array = []
model_array = []

n_array = [ n + 1 for n in range(30)]

for n in n_array:
    model = DecisionTreeRegressor(max_depth=n)
    model.fit(X_train, Y_train)
    model_array.append(model)
    Y_pred = model.predict(X_test)
    mse = mean_squared_error(Y_test, Y_pred)
    mse_array.append(mse)
    mae = mean_absolute_error(Y_test, Y_pred)
    mae_array.append(mae)
    r2 = r2_score(Y_test, Y_pred)
    r2_array.append(r2)


mse, index = min((mse, idx) for (idx, mse) in enumerate(mse_array))
mae = mae_array[index]
r2 = r2_array[index]

model = model_array[index]

# Store model
joblib.dump(model, 'decision_tree.pkl')
```

Fig 4.5.3 Decision Tree

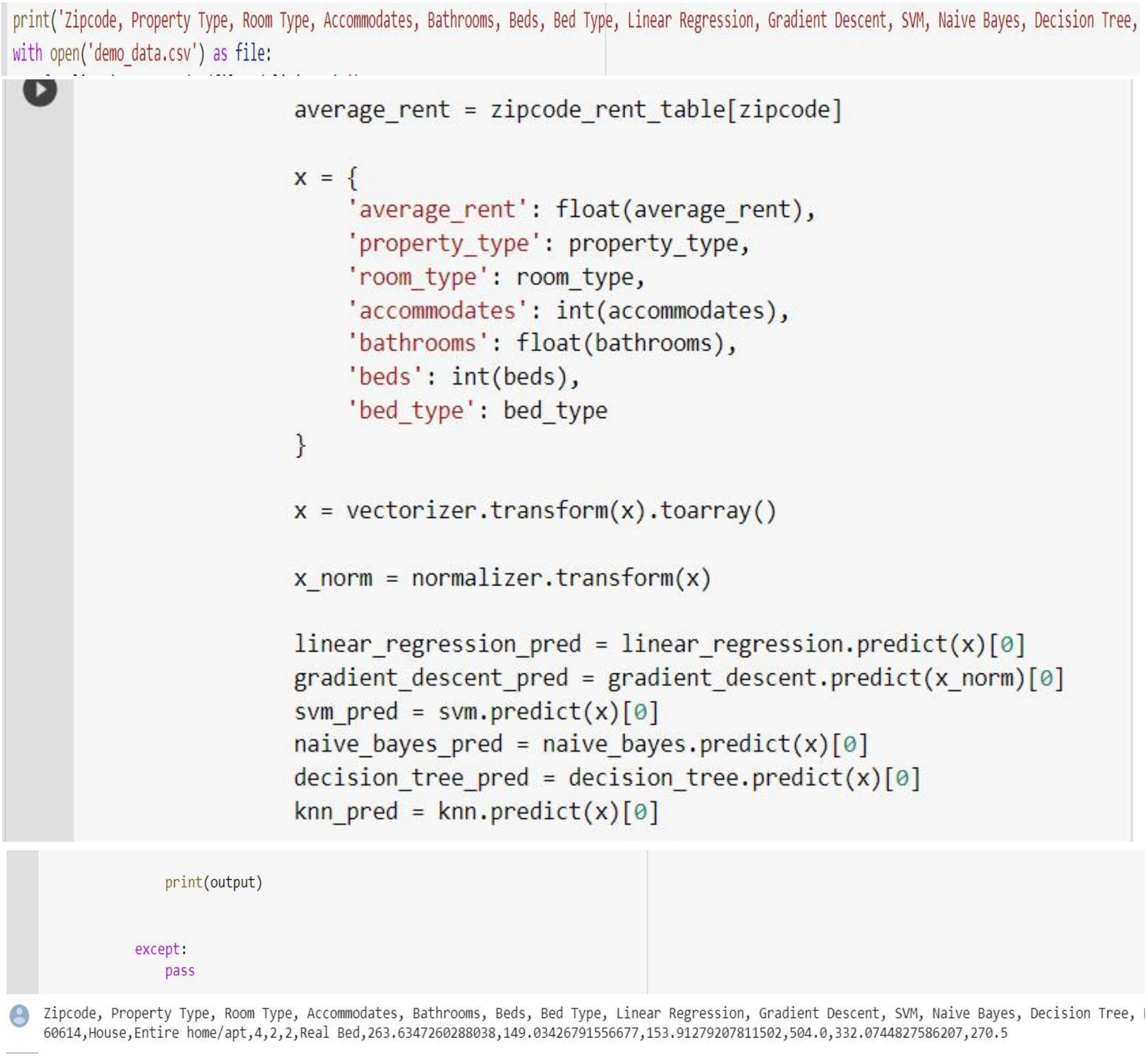
For the Decision Tree Model, we consider different models for different maxdepths of the decision tree, ranging from a max-depth of 1 to a max-depth of 30. We then select the model for the list that produces the smallest Mean Squared Error. We further decide to graph these relationships using Matplotlib.

The same is done for KNN, as we consider different KNN models for different number of neighbors, ranging from a 1 neighbor to 50 neighbors. Finally, we also output the decision tree as a graph using Pydotplus.

Furthermore, all the models, as well as the vectorizer and normalizer are stored on disk as '.pkl' files that Scikit-Learn can then invoke at will. This is very practical as this means that we can store a trained model on disk and not have to re-train it overtime it is used.

After all the models are processed, we output the results of their evaluation (i.e. MSE, MAE, and R2) as a .csv file. We then repeat the same process while omitting the the Average Rent feature. This will gives us two sets of evaluated models that we then may compare to assess the usefulness of the Average Rent feature

4.2.5 Prediction using inputs by user



```
print('Zipcode, Property Type, Room Type, Accommodates, Bathrooms, Beds, Bed Type, Linear Regression, Gradient Descent, SVM, Naive Bayes, Decision Tree,
with open('demo_data.csv') as file:
```

```
                average_rent = zipcode_rent_table[zipcode]

                x = {
                    'average_rent': float(average_rent),
                    'property_type': property_type,
                    'room_type': room_type,
                    'accommodates': int(accommodates),
                    'bathrooms': float(bathrooms),
                    'beds': int(beds),
                    'bed_type': bed_type
                }

                x = vectorizer.transform(x).toarray()

                x_norm = normalizer.transform(x)

                linear_regression_pred = linear_regression.predict(x)[0]
                gradient_descent_pred = gradient_descent.predict(x_norm)[0]
                svm_pred = svm.predict(x)[0]
                naive_bayes_pred = naive_bayes.predict(x)[0]
                decision_tree_pred = decision_tree.predict(x)[0]
                knn_pred = knn.predict(x)[0]
```

```
                print(output)


            except:
                pass
```

```
Zipcode, Property Type, Room Type, Accommodates, Bathrooms, Beds, Bed Type, Linear Regression, Gradient Descent, SVM, Naive Bayes, Decision Tree,
60614,House,Entire home/apt,4,2,2,Real Bed,263.6347260288038,149.03426791556677,153.91279207811502,504.0,332.0744827586207,270.5
```

Fig 4.6 Prediction using User inputs

The above snippet is demo.py which is a simple Python script that imports all the trained models and gets demo data from ‘demo_data.csv’ and uses this data to make predictions. This demo can have actual pragmatic uses as it can be used to generate a nightly rental price for a home/apartment that would like to go on Airbnb.

4.2.6 Data Visualization

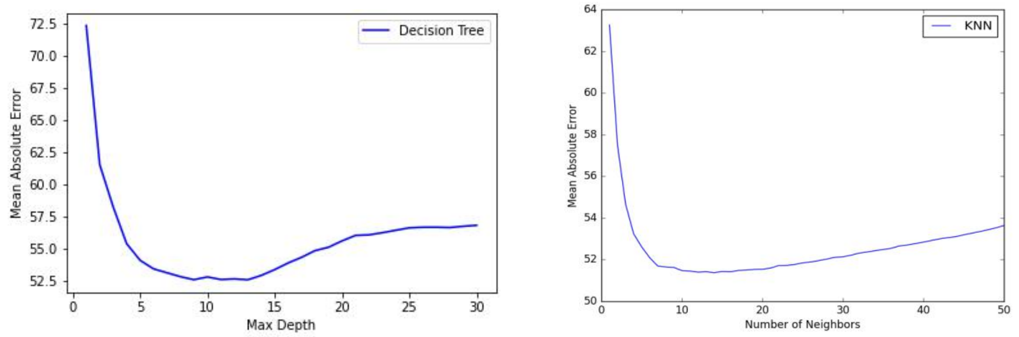


Fig 4.7.1 Mean Absolute error(Decision tree and KNN)

The above plotting represent Mean Absolute error in applied algorithm of Decision tree and KNN with average rent feature.

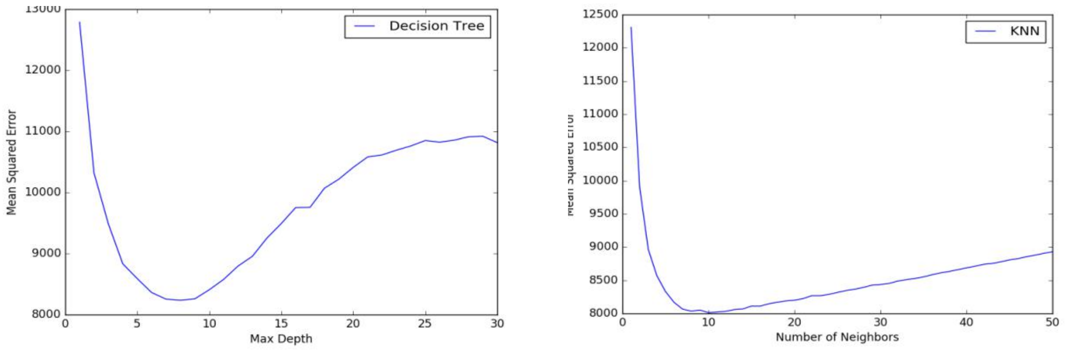


Fig 4.7.2 Mean Squared error(Decision tree and KNN)

The above plotting represent Mean Squared error in applied algorithm of Decision tree and KNN with average rent feature.

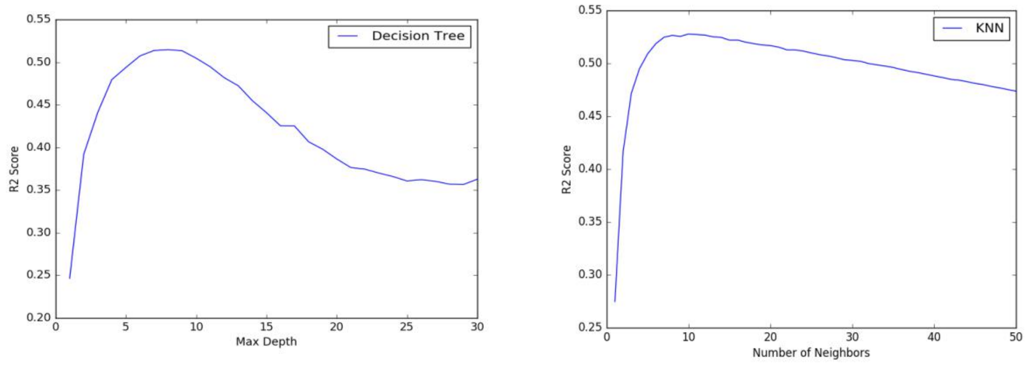
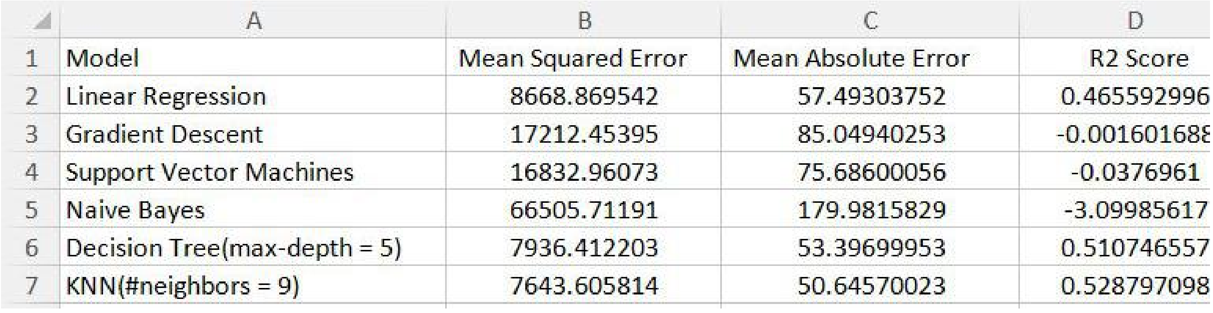


Fig 4.7.3 R2 Score (Decision Tree and KNN)

The above plotting represent Mean Squared error in applied algorithm of Decision tree and KNN with average rent feature.

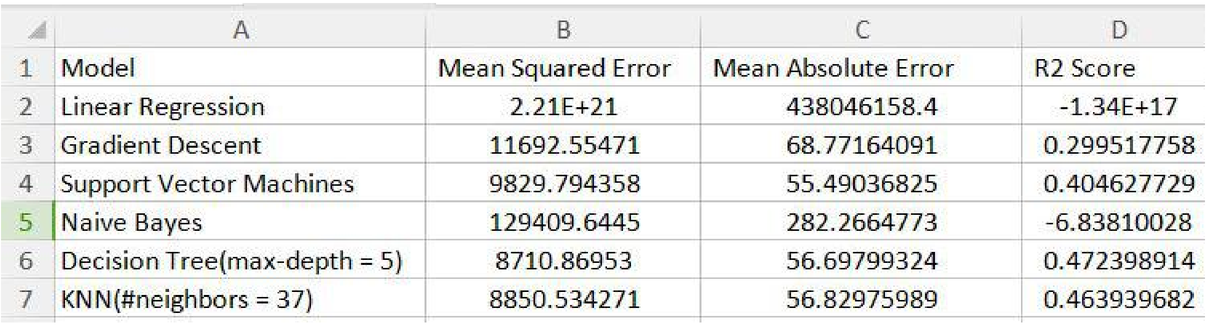
4.2.7 Data Analysis

We consider two different evaluation outputs: evaluation with average rent feature and without



	A	B	C	D
1	Model	Mean Squared Error	Mean Absolute Error	R2 Score
2	Linear Regression	8668.869542	57.49303752	0.465592996
3	Gradient Descent	17212.45395	85.04940253	-0.001601688
4	Support Vector Machines	16832.96073	75.68600056	-0.0376961
5	Naive Bayes	66505.71191	179.9815829	-3.09985617
6	Decision Tree(max-depth = 5)	7936.412203	53.39699953	0.510746557
7	KNN(#neighbors = 9)	7643.605814	50.64570023	0.528797098

Fig 4.8.1 Evaluation of Models with Average Rent Feature



	A	B	C	D
1	Model	Mean Squared Error	Mean Absolute Error	R2 Score
2	Linear Regression	2.21E+21	438046158.4	-1.34E+17
3	Gradient Descent	11692.55471	68.77164091	0.299517758
4	Support Vector Machines	9829.794358	55.49036825	0.404627729
5	Naive Bayes	129409.6445	282.2664773	-6.83810028
6	Decision Tree(max-depth = 5)	8710.86953	56.69799324	0.472398914
7	KNN(#neighbors = 37)	8850.534271	56.82975989	0.463939682

Fig 4.8.2 Evaluation of models without Average Rent Feature

Given these outputs, we can notice a number of things. First of all, in both scenarios, Decision Tree and KNN are the models that perform best. Second, we can notice an increase in the R2 scores and a decrease in the MSE and MAE for Decision Tree and KNN when the Average Rent feature is included. This therefore supports our hypothesis that the average rent prices in the area affect the price of an Airbnb listing.

Finally, when analyzing the evaluation outputs, we can conclude that the best way to model Airbnb prices is via a KNN regression model with 9 neighbors while including the average rent prices from Zillow as a feature. This makes intuitive sense as it means that the best way to price a listing on Airbnb is to look at similar listings. This is what people do intuitively.

CHAPTER 5

CONCLUSION

The Machine learning based internship has helped me into designing a machine learning application. Machine learning is all about data recollection and data processing. Through this internship, I have understood what exactly machine learning is, how it works and the effects it has and will have on society. As more information it is given to the model, the more it learns and the more accurate it becomes with the result. Therefore, it recognizes patterns and will give updates as relevant data is fed through. Machine Learning algorithms are created to handle and shift through large amounts of data to identify a pattern and produce a result based on the patterns. A human programmer would not be able to sit and go through every single point which would be extremely expensive and time-consuming.

Learning such technology with practical experience gives a big boost of confidence about my career in the field of machine learning by building real-time, useful and important applications to the industry such as the wine quality prediction project and motivates me to build more applications.

REFERENCES

[1] https://www.geeksforgeeks.org/machine-learning/

[2] https://anaconda.en.softonic.com/download

[3] https://www.kaggle.com/datasets

[4] https://www.kaggle.com/carrie1/ecommerce-data

[5] https://www.kaggle.com/dgomonov/new-york-city-airbnb-open-data

[6] https://www.simplifiedpython.net/python-chatbot/