HOURLY BIKE RENTAL DEMAND PREDICTION

by

REHANA KHATOON – 18013440063

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

R.V.S. COLLEGE OF ENGINEERING AND TECHNOLOGY

UNDER JUT

2018-2022

A Project Report

On

HOURLY BIKE RENTAL DEMAND PREDICTION

by

REHANA KHATOON – 18013440063

Department of Computer Science & Engineering

R.V.S. College of Engineering and Technology

Jamshedpur – 831012

2018 - 2022

CERTIFICATE

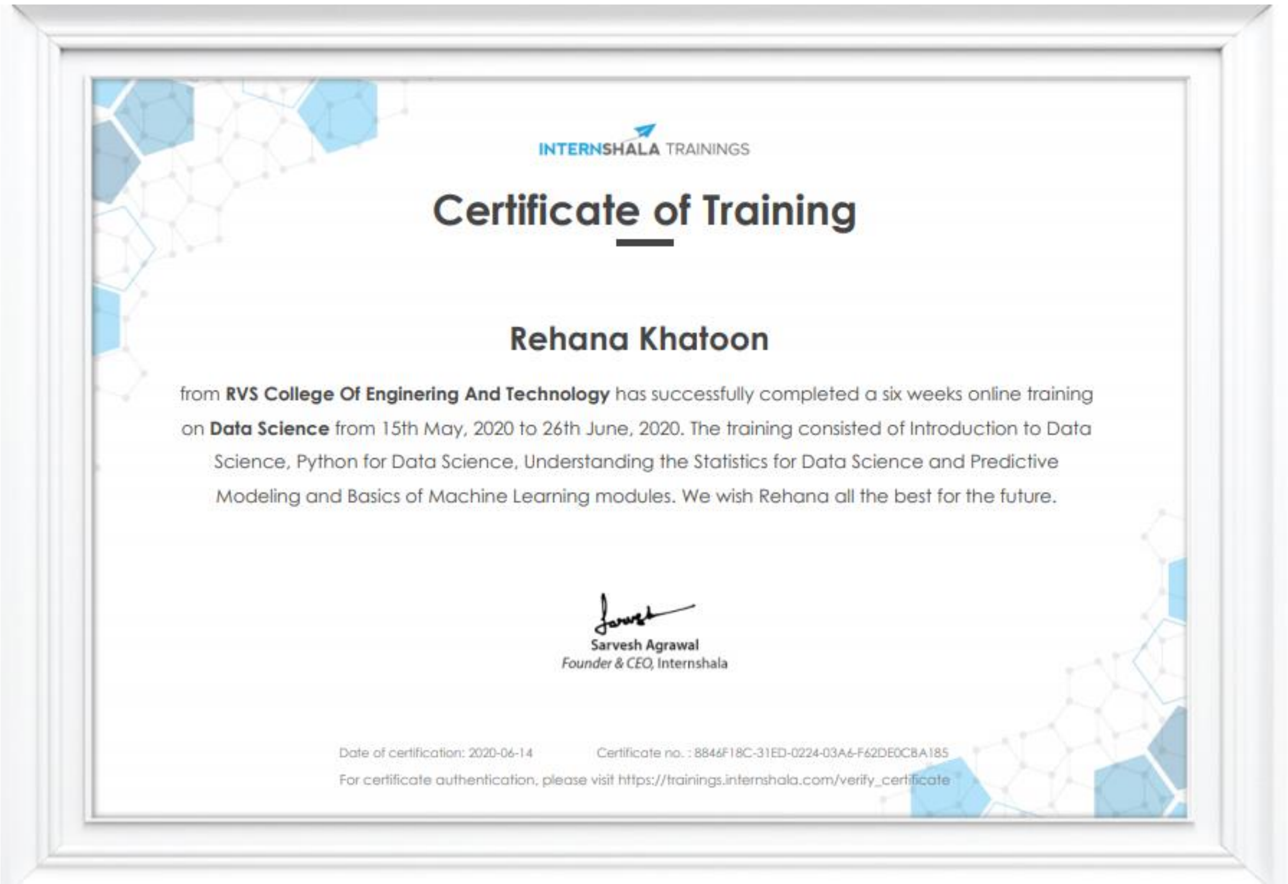
This is to certify that the project work entitled **“Hourly bike rental demand prediction”** submitted by **Rehana Khatoon** is approved for the 4th semester internship project. This work is submitted to the department as a part of evaluation as 4th semester Internship Project under the supervision of **Prof. Manjeet Singh.**

Prof. Manjeet Singh
Asst. Professor
Dept. of CSE

(Supervisor)

Prof. Jeevan Kumar
Asst. Professor & HOD
Dept. of CSE
(Head of the Department)

CERTIFICATE FROM INTERNSHALA TRAINING



INTERNSHALA TRAININGS

Certificate of Training

Rehana Khatoon

from **RVS College Of Enginering And Technology** has successfully completed a six weeks online training on **Data Science** from 15th May, 2020 to 26th June, 2020. The training consisted of Introduction to Data Science, Python for Data Science, Understanding the Statistics for Data Science and Predictive Modeling and Basics of Machine Learning modules. We wish Rehana all the best for the future.

Sarvesh Agrawal
Founder & CEO, Internshala

Date of certification: 2020-06-14

Certificate no. : 8846F18C-31ED-0224-03A6-F62DE0C8A185

For certificate authentication, please visit https://trainings.internshala.com/verify_certificate

Image 1. Certificate of internship

ACKNOWLEDGEMENT

I avail this opportunity to express my profound sense of gratitude to Asst. Prof. Manjeet Singh sir for his constant supervision, guidance and encouragement right from the beginning till the completion of the project.

We wish to thank all the faculty members and supporting staffs of Dept. of Computer Science and Engineering for their valuable suggestions, timely advice and encouragement.

REHANA KHATOON -CSE/084/18

ABSTRACT

Land use is a primary factor affecting the demand for public bicycle rentals. Demand for public bicycle rentals during different periods of time were predicted using the following procedures. First, walking distances from the rental stations where riders returned the public bicycles to the final destinations were obtained by field investigation, and the 85th percentile statistical values were used as the scopes of influence of those stations. Then, a relationship model among the rental demands for public bicycles and the features of land use inside the influence scope of the rental station was established based on a linear regression model. Finally, considering the public bicycle system in the old urban region of Indian city, the newly established prediction model for rental demand was tested. Results shows that the model can predict the daily hourly bike rental demand, rental demand during the morning peak, returns during the morning peak, rental demands during the evening peak, and returns during the evening peak. The demand prediction model can provide a significant theoretical basis for preparing the layout stations, operation and management strategies, and vehicle scheduling in the public bicycle system.

CONTENTS

1. Introduction……………………………………………………………………….1
   1. DATA SCIENCE…………………………………………......2
   2. PYTHON…………………………………………………5
   3. STATISTICS……………………………………………...7
   4. MACHINE LEARNING……………………………………12
   5. PREDICTIVE MODELLING……………………………….14
   6. IDE (JUPYTER NOTEBOOK) ……………………….…..16
2. Codes & screenshot………………………………………………………...17
3. Conclusion ……………………………………………..…………………….25
4. Bibliography………………………...……………………………….………26
5. References…………………………………………………………………….27

INTRODUCTION

This report is about “Hourly bike rental demand prediction” which has been implemented to predict whether there will be demand for rent or not in the next coming hours.

Since 2009, the public bicycle system has rapid development in India. As the conventional type, the public bicycle system has fixed rental stations. But since 2016, a large number of shared bicycles have appeared in many cities in India, namely, public bicycles without fixed rental stations based on the internet, and related scholars have studied the operation of shared bicycles. However, the development prospects of shared bicycles in India are now taking place in a massive scale.

So, for this very purpose I decided to make a project on hourly bike rental demand so that we will be able to predict the demand and then can make our decisions in accordance with that.

DATA SCIENCE

Data science is an inter-disciplinary field that uses scientific methods, processes, algorithms and systems to extract knowledge and insights from many structural and unstructured data.[1][2] Data science is related to data mining, deep learning and big data.

Data science is a "concept to unify statistics, data analysis, machine learning, domain knowledge and their related methods" in order to "understand and analyze actual phenomena" with data.[3] It uses techniques and theories drawn from many fields within the context of mathematics, statistics, computer science, domain knowledge and information science. Turing award winner Jim Gray imagined data science as a "fourth paradigm" of science (empirical, theoretical, computational and now data-driven) and asserted that "everything about science is changing because of the impact of information technology" and the data deluge.

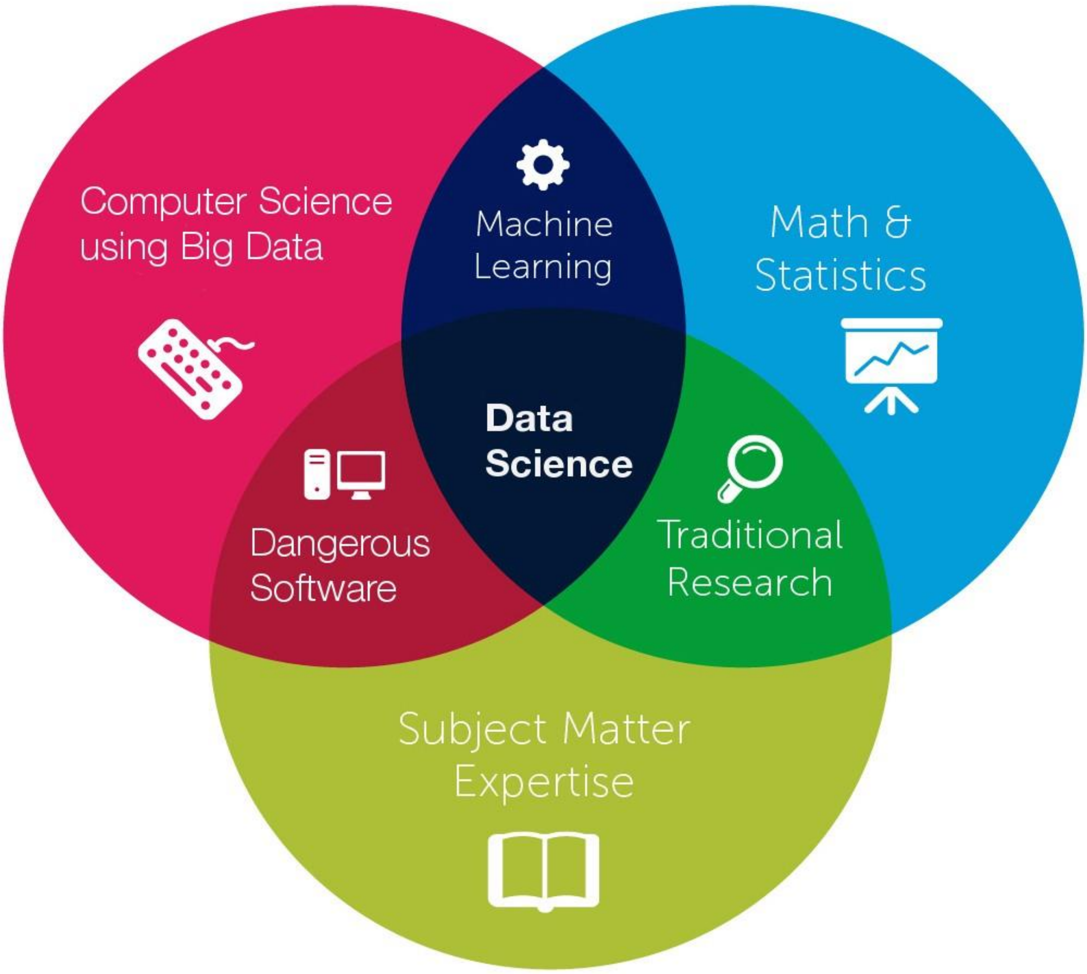


Image 2. Data Science Uses

APPLICATIONS OF DATA SCIENCE

1. **Data Science in Finance Industry-** Finance has always been about data. As a matter of fact, data science and finance go hand in hand. Even before the term data science was devised, Finance was using it. Data Science widely used in areas like risk analytics, customer management, fraud detection, and algorithmic trading.

2. **Data Science in Retail Industry —** Retail industries have always been hugely benefited by data science. The retail sector performs several tasks with the help of data science tools/algorithms starting from Recommendation engines, Customer sentiment analysis, Market basket analysis, Price optimisation, Fraud detection, etc. "According to IBM, 62% of Retailers say the use of data science techniques gives them a serious competitive advantage".

3. **Data Science in Ecommerce-** Every eCommerce needs data science to improve the ability to target the audience.

4. **Digital marketing with Data Science-** Data science has the capability of revolutionising your digital marketing strategies and enhancing your customer's experience.

5. **Data Science in Healthcare-** With data science the healthcare sector is moving to a whole new level. The healthcare receives great benefits from data science, below are the top 7 data science use cases in healthcare-

- Virtual assistance for patients and customer support

- Medical image analysis

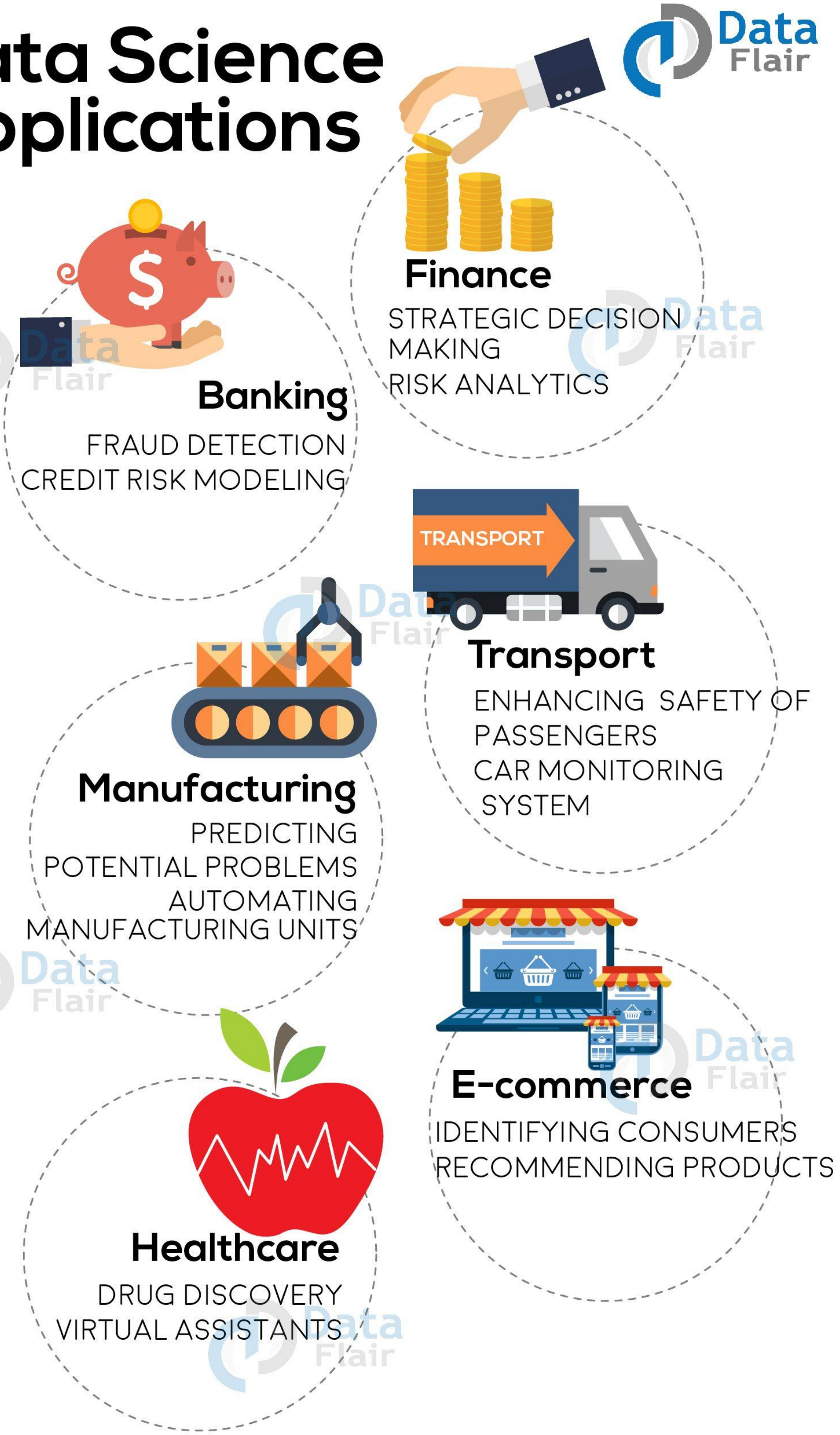
- Predictive medicine: prognosis and diagnostic accuracy

- Managing customer data

6. **Data Science in Education-** Data science is a powerful tool for shaping academic skills as well as non-academic skills like social-emotional skills, Measuring instructor performance, Monitoring student requirements and universities are using it for Innovating the curriculum. Data science plays a diverse role in the field of education.

7. **Data science in Banking-** The banking industries are using data science to improve their products, services, and security. All the banks across the globe are analysing data to provide better experiences to their customers.

Data Science Applications



Finance

STRATEGIC DECISION MAKING
RISK ANALYTICS

Banking

FRAUD DETECTION
CREDIT RISK MODELING

Transport

ENHANCING SAFETY OF PASSENGERS
CAR MONITORING SYSTEM

Manufacturing

PREDICTING POTENTIAL PROBLEMS
AUTOMATING MANUFACTURING UNITS

E-commerce

IDENTIFYING CONSUMERS
RECOMMENDING PRODUCTS

Healthcare

DRUG DISCOVERY
VIRTUAL ASSISTANTS

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 with reference counting.

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 was officially discontinued in 2020 (first planned for 2015), and "Python 2.7.18 is the last Python 2.7 release and therefore the last Python 2 release." No more security patches or other improvements will be released for it. With Python 2's end-of-life, only Python 3.5.x and later are supported.

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

WHY PYTHON?

1. Web development:

Python can be used to make web-applications at a rapid rate. Why is that? It is because of the frameworks Python uses to create these applications. There is *common-backend logic* that goes into making these frameworks and a number of libraries that can help integrate protocols such as HTTPS, FTP, SSL etc. and even help in the processing of JSON, XML, E-Mail and so much more.

2. Game Development

Python is also used in the development of interactive games. There are libraries such as PySoy which is a 3D game engine supporting Python 3, PyGame which provides functionality and a library for game development. Games such as Civilization-IV, Disney’s Toontown Online, Vega Strike etc. have been built using Python.

3. Machine Learning & Artificial Intelligence

Machine Learning and Artificial Intelligence are the talks of the town as they yield the most promising careers for the future. We make the computer learn based on past experiences through the data stored or better yet, create algorithms which makes the computer learn by itself. The programming language that mostly everyone chooses? It's Python. Why? Support for these domains with the **libraries** that exist already such as Pandas, Scikit-Learn, NumPy and so many more.

4. Data Science & Data visualisation

Data is money if you know how to extract relevant information which can help you take calculated risks and increase profits. You study the data you have, perform operations and extract the information required. Libraries such as Pandas, NumPy help you in extracting information.

5. Desktop GUI

Python can be used to program **desktop applications**. It provides the Tkinter library that can be used to develop user interfaces. There are some other useful toolkits such as the wxWidgets, Kivy, PYQT that can be used to create applications on several platforms.

6. Web Scrapping Application

Python can be used to pull a large amount of data from websites which can then be helpful in various real-world processes such as price comparison, job listings, research and development and much more.

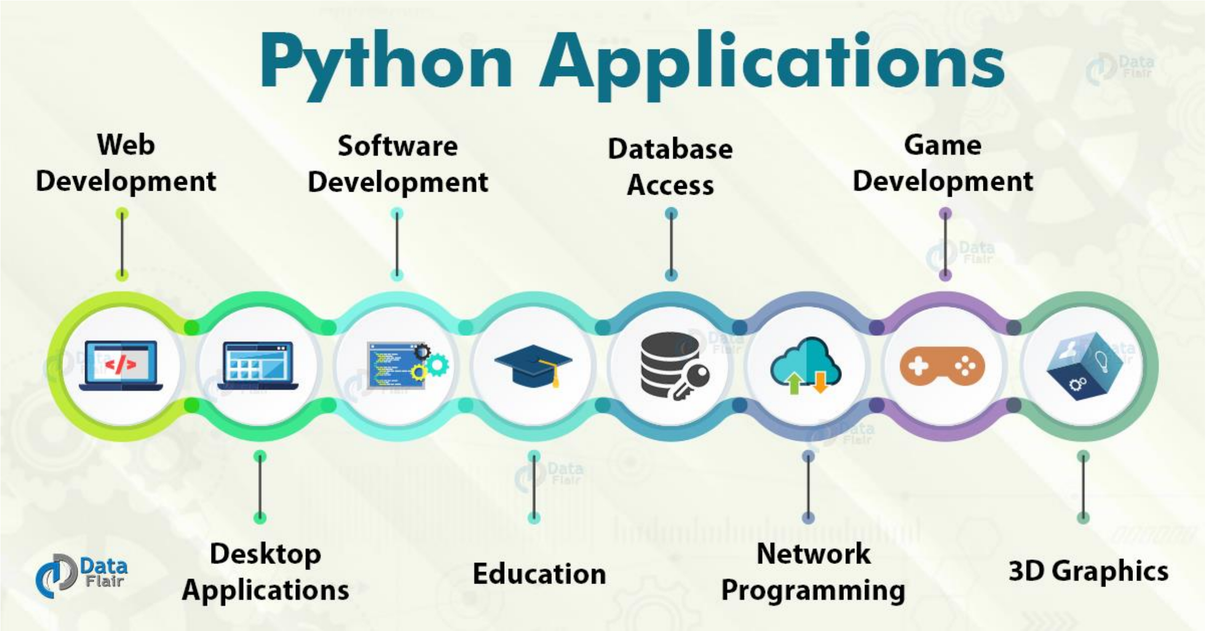


Image 3. Applications of Python

STATISTICS

Statistics is used to process complex problems in the real world so that Data Scientists and Analysts can look for meaningful trends and changes in Data. In simple words, Statistics can be used to derive meaningful insights from data by performing mathematical computations on it. Several Statistical functions, principles and algorithms are implemented to analyse raw data, build a Statistical Model and infer or predict the result.

The field of Statistics has an influence over all domains of life, the Stock market, life sciences, weather, retail, insurance and education are but to name a few.

Moving ahead. let's discuss the basic terminologies in Statistics.

Terminologies in Statistics – Statistics for Data Science. One should be aware of a few key statistical terminologies while dealing with Statistics for Data Science. I've discussed these terminologies below:

- Population is the set of sources from which data has to be collected.
- A Sample is a subset of the Population
- A Variable is any characteristics, number, or quantity that can be measured or counted. A variable may also be called a data item.
- Also known as a statistical model, A statistical Parameter or population parameter is a quantity that indexes a family of probability distributions. For example, the mean, median, etc of a population.

Before we move any further and discuss the categories of Statistics, let's look at the types of analysis.

Types of Analysis:
An analysis of any event can be done in one of two ways:

1. Quantitative Analysis: Quantitative Analysis or the Statistical Analysis is the science of collecting and interpreting data with numbers and graphs to identify patterns and trends.
2. Qualitative Analysis: Qualitative or Non-Statistical Analysis gives generic information and uses text, sound and other forms of media to do so.

Categories in Statistics

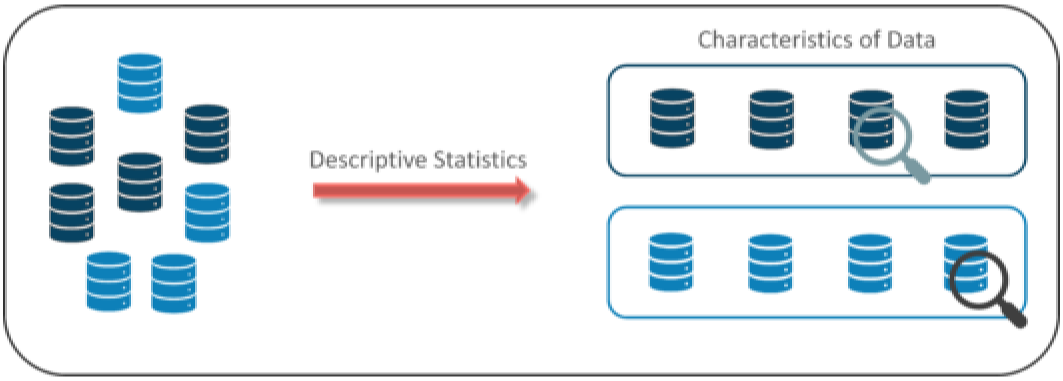
There are two main categories in Statistics, namely:

1. Descriptive Statistics
2. Inferential Statistics

Descriptive Statistics

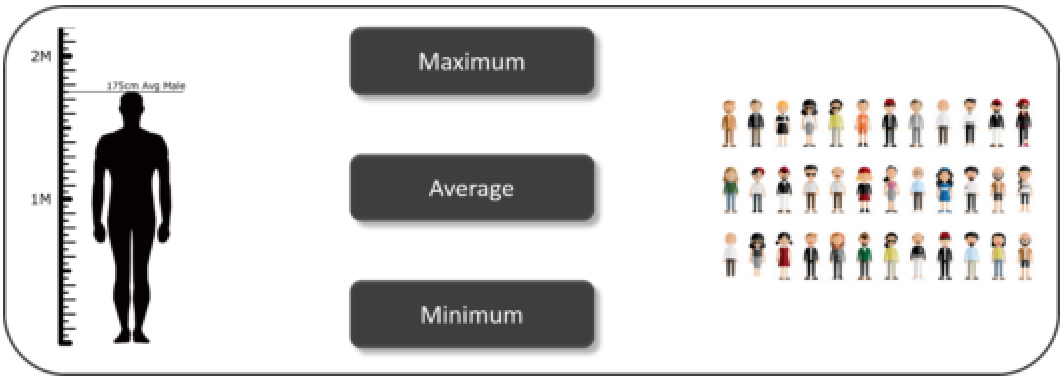
Descriptive Statistics uses the data to provide descriptions of the population, either through numerical calculations or graphs or tables.

Descriptive Statistics helps organize data and focuses on the characteristics of data providing parameters.



Descriptive Statistics – Math and Statistics for Data Science – Edureka

Suppose you want to study the average height of students in a classroom, in descriptive statistics you would record the heights of all students in the class and then you would find out the maximum, minimum and average height of the class.

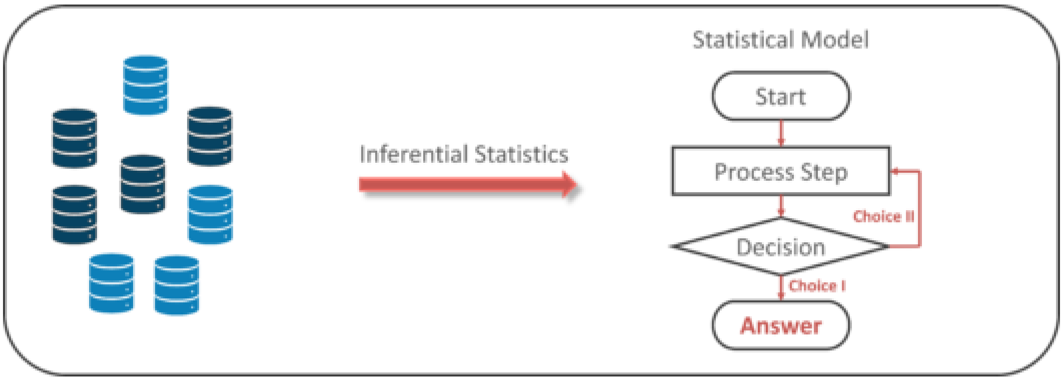


Descriptive Statistics Example – Math and Statistics for Data Science – Edureka

Inferential Statistics

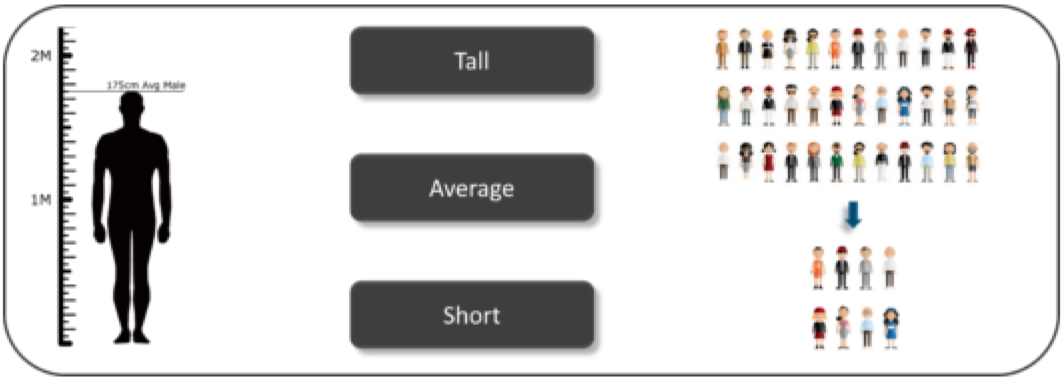
Inferential Statistics makes inferences and predictions about a population based on a sample of data taken from the population in question.

Inferential statistics generalizes a large data set and applies probability to arrive at a conclusion. It allows you to infer parameters of the population based on sample stats and build models on it.



Inferential Statistics – Math and Statistics for Data Science – Edureka

So, if we consider the same example of finding the average height of students in a class, in Inferential Statistics, you will take a sample set of the class, which is basically a few people from the entire class. You already have had grouped the class into tall, average and short. In this method, you basically build a statistical model and expand it for the entire population in the class.

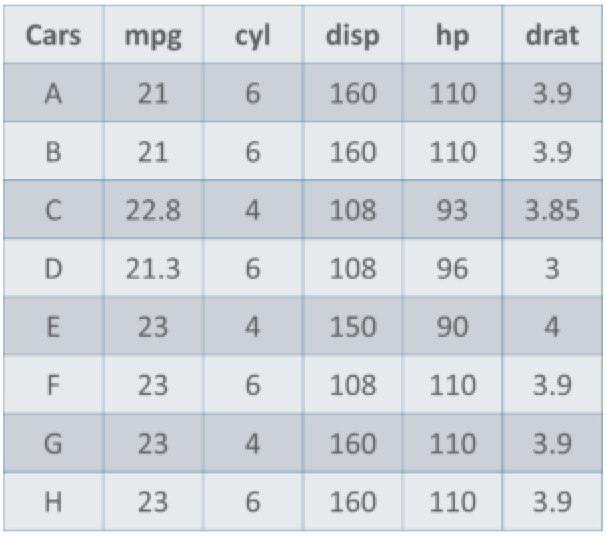


Inferential Statistics Example – Maths and Statistics for Data Science – Edureka

Now let’s focus our attention on Descriptive Statistics and see how it can be used to solve analytical problems.

Understanding Descriptive Analysis

When we try to represent data in the form of graphs, like histograms, line plots, etc. the data is represented based on some kind of central tendency. Central tendency measures like, mean, median, or measures of the spread, etc are used for statistical analysis. To better understand Statistics lets discuss the different measures in Statistics with the help of an example.



Cars	mpg	cyl	disp	hp	drat
A	21	6	160	110	3.9
B	21	6	160	110	3.9
C	22.8	4	108	93	3.85
D	21.3	6	108	96	3
E	23	4	150	90	4
F	23	6	108	110	3.9
G	23	4	160	110	3.9
H	23	6	160	110	3.9

Cars Data Set – Math and Statistics for Data Science – Edureka

Here is a sample data set of cars containing the variables:

1. Cars
2. Mileage per Gallon (mpg)
3. Cylinder Type (cyl)
4. Displacement (disp)
5. Horse Power (hp)
6. Real Axle Ratio (drat).

Before we move any further, let's define the main Measures of the Center or Measures of Central tendency.

Measures of The Center

1. Mean: Measure of average of all the values in a sample is called Mean.
2. Median: Measure of the central value of the sample set is called Median.
3. Mode: The value most recurrent in the sample set is known as Mode.

Using descriptive Analysis, you can analyse each of the variables in the sample data set for mean, standard deviation, minimum and maximum.

- If we want to find out the mean or average horsepower of the cars among the population of cars, we will check and calculate the average of all values. In this case, we'll take the sum of the Horse Power of each car, divided by the total number of cars:

Mean = (110+110+93+96+90+110+110+110)/8 = 103.625

- If we want to find out the center value of mpg among the population of cars, we will arrange the mpg values in ascending or descending order and choose the middle value. In this case, we have 8 values which is an even entry. Hence, we must take the average of the two middle values.

The mpg for 8 cars: 21,21,21.3,22.8,23,23,23,23
Median = (22.8+23)/2 = 22.9

- If we want to find out the most common type of cylinder among the population of cars, we will check the value which is repeated most number of times. Here we can see that the cylinders come in two values, 4 and 6. Take a look at the data set, you can see that the most recurring value is 6. Hence 6 is our Mode.

Measures of The Spread

Just like the measure of center, we also have measures of the spread, which comprises of the following measures:

1. Range: It is the given measure of how spread apart the values in a data set are.
2. Inter Quartile Range (IQR): It is the measure of variability, based on dividing a data set into quartiles.
3. Variance: It describes how much a random variable differs from its expected value. It entails computing squares of deviations.
   1. Deviation is the difference between each element from the mean.
   2. Population Variance is the average of squared deviations
   3. Sample Variance is the average of squared differences from the mean
4. Standard Deviation: It is the measure of the dispersion of a set of data from its mean.

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.[3] 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.

Machine learning approaches

Machine learning approaches are traditionally divided into three broad categories, depending on the nature of the "signal" or "feedback" available to the learning system:

- Supervised learning: The computer is presented with example inputs and their desired outputs, given by a "teacher", and the goal is to learn a general rule that maps inputs to outputs.
- Unsupervised learning: No labels are given to the learning algorithm, leaving it on its own to find structure in its input. Unsupervised learning can be a goal in itself (discovering hidden patterns in data) or a means towards an end (feature learning).
- Reinforcement learning: A computer program interacts with a dynamic environment in which it must perform a certain goal (such as driving a vehicle or playing a game against an opponent). As it navigates its problem space, the program is provided feedback that's analogous to rewards, which it tries to maximise.

Other approaches have been developed which don't fit neatly into this three-fold categorisation, and sometimes more than one is used by the same machine learning system. For example topic modelling, dimensionality reduction or meta learning.

As of 2020, deep learning has become the dominant approach for much ongoing work in the field of machine learning.

Types of machine learning

Classical machine learning is often categorized by how an algorithm learns to become more accurate in its predictions. There are four basic
approaches: supervised learning, unsupervised learning, semi-supervised learning and reinforcement learning. The type of algorithm a data scientist chooses to use depends on what type of data they want to predict.

- Supervised learning. In this type of machine learning, data scientists supply algorithms with labeled training data and define the variables they want the algorithm to assess for correlations. Both the input and the output of the algorithm is specified.
- Unsupervised learning. This type of machine learning involves algorithms that train on unlabeled data. The algorithm scans through data sets looking for any meaningful connection. Both the data algorithms train on and the predictions or recommendations they output are predetermined.
- Semi-supervised learning. This approach to machine learning involves a mix of the two preceding types. Data scientists may feed an algorithm mostly labeled training data, but the model is free to explore the data on its own and develop its own understanding of the data set.
- Reinforcement learning. Reinforcement learning is typically used to teach a machine to complete a multi-step process for which there are clearly defined rules. Data scientists program an algorithm to complete a task and give it positive or negative cues as it works out how to complete a task. But for the most part, the algorithm decides on its own what steps to take along the way.

PREDICTIVE MODELLING

Predictive modelling is a process that uses data and statistics to predict outcomes with data models. These models can be used to predict anything from sports outcomes and TV ratings to technological advances and corporate earnings.

Predictive modelling is also often referred to as:

Predictive analytics

Predictive analysis

Machine learning

These synonyms are often used interchangeably. However, predictive analytics most often refers to commercial applications of predictive modelling, while predictive modelling is used more generally or academically. Of the terms, predictive modelling is used more frequently, which is illustrated in the Google Trends chart below. Machine learning is also distinct from predictive modelling and is defined as the use of statistical techniques to allow a computer to construct predictive models. In practice, machine learning and predictive modelling are often used interchangeably. However, machine learning is a branch of artificial intelligence, which refers to intelligence displayed by machines.

What are the types of predictive models?

Broadly speaking, predictive models fall into two camps: parametric and non-parametric. Although these terms might seem like technical jargon, the essential difference is that parametric models make more assumptions and more specific assumptions about the characteristics of the population used in creating the model. Specifically, some of the different types of predictive models are:

- Ordinary Least Squares
- Generalized Linear Models (GLM)
- Logistic Regression
- Random Forests
- Decision Trees
- Neural Networks
- Multivariate Adaptive Regression Splines (MARS)

Each of these types has a particular use and answers a specific question or uses a certain type of dataset. Despite the methodological and mathematical differences among the model types, the overall goal of each is similar: to predict future or unknown outcomes based on data about past outcomes.

The Future of Predictive Modelling

The future of predictive modelling is, undoubtedly, closely tied to artificial intelligence. As computing power continues to increase, data collection rises exponentially, and new technologies and methods are born, computers will bear the brunt of the load when it comes to creating models. The global management consulting firm McKinsey and Co. recently studied future trends, some of which are detailed below.

Technological Advancements

Partially due to recent advancements in computing power and data quantities, predictive modelling technologies have improved the impact of regular newsworthy breakthroughs. Predictive algorithms are becoming extremely sophisticated in many fields, notably computer vision, complex games, and natural language.

Changes in Work

With more intelligent computers, the work of predictive modelling professionals, much like with other occupations, will change to adapt to newly available predictive technology. People who work in predictive modelling will not likely become obsolete, but their roles will shift in a way that complements new predictive technological features and abilities, and they will need to acquire new skills to excel in these new roles.

Risk Mitigation

Advances in predictive technology are extremely promising in terms of commercial and scientific value creation, but they do require risk mitigation as well. Some of these risks center on data privacy and security. With exponential increases in data volume, the importance of protecting data from hackers and mitigating other privacy concerns increase as well. Additionally, researchers point out the risk of hard wiring overt and unconscious societal biases into predictive models and algorithms, an issue that will be of great importance to policymakers and big technology companies.

IDE USED - JUPYTER NOTREBOOK

LANGUAGE USED – PYTHON

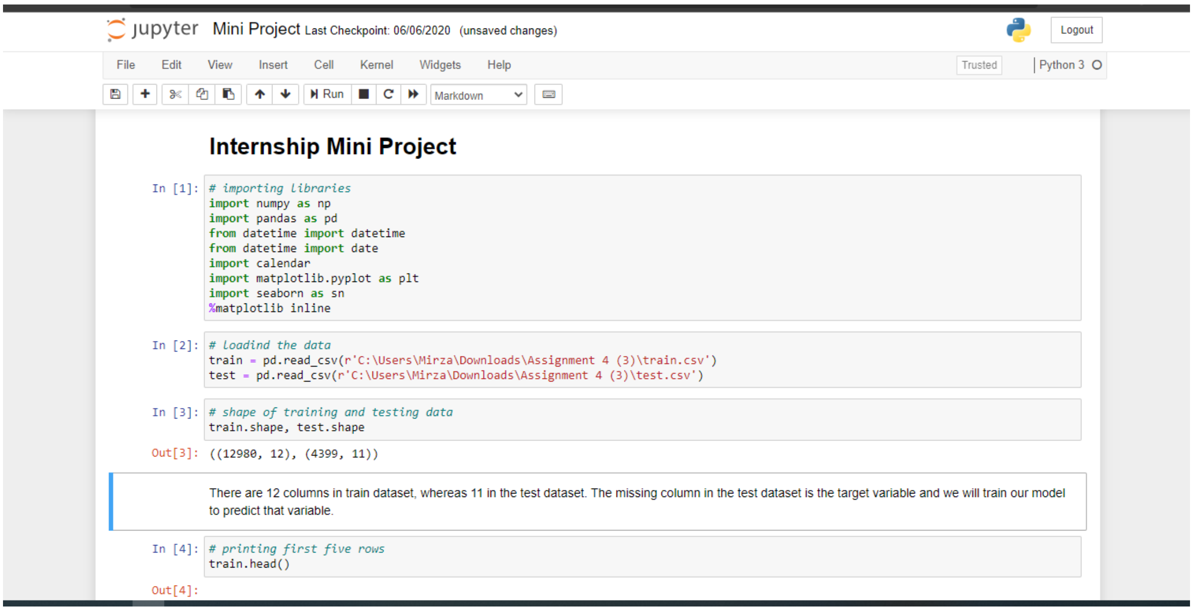
About Jupyter Notebook

These notebooks are popular among data scientists and are used both internally and externally to share information about data exploration, model training, and model deployment. Jupyter Notebook supports running both Python and R, two of the most common languages used by data scientists today.

How We Use It

We don’t think you should be limited to creating algorithms/models solely through our UI. Instead, to give data scientists a better and more comprehensive experience, we built an API endpoint that gives more control over your algorithms. You can create, update, and publish algorithms using our Python client, which lets data scientists deploy their models directly from their existing Jupyter Notebook workflows.

CODES & SCREENSHOTS


jupyter Mini Project Last Checkpoint: 06/06/2020 (unsaved changes)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3
Run
Markdown

Internship Mini Project

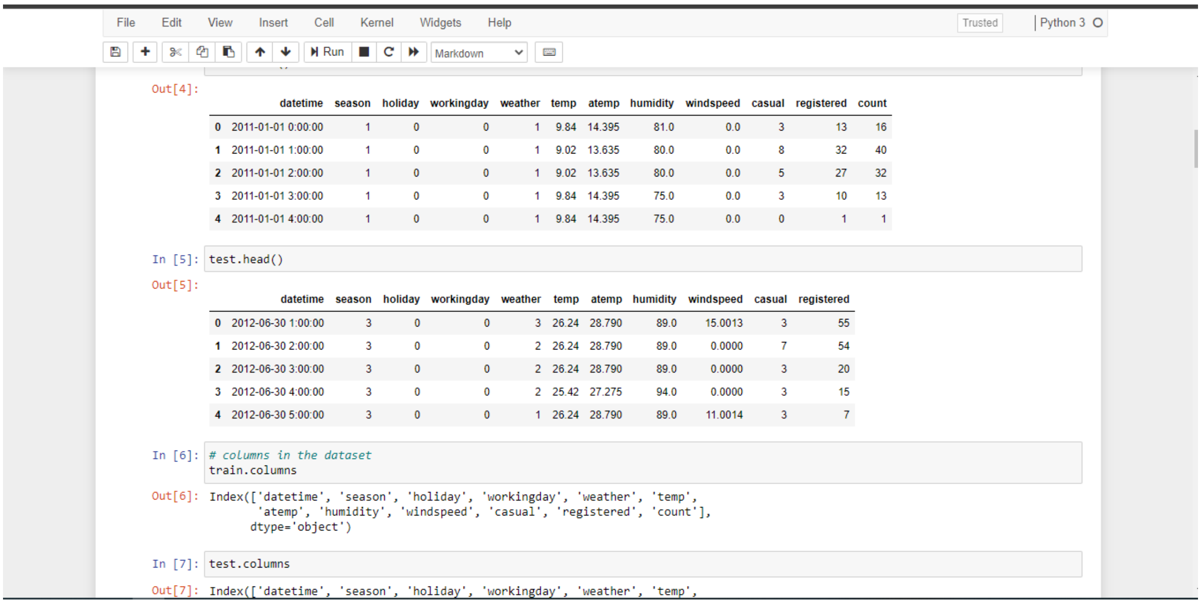
```
In [1]: # importing libraries
        import numpy as np
        import pandas as pd
        from datetime import datetime
        from datetime import date
        import calendar
        import matplotlib.pyplot as plt
        import seaborn as sn
        %matplotlib inline
```

```
In [2]: # loadind the data
        train = pd.read_csv(r'C:\Users\Mirza\Downloads\Assignment 4 (3)\train.csv')
        test = pd.read_csv(r'C:\Users\Mirza\Downloads\Assignment 4 (3)\test.csv')
```

```
In [3]: # shape of training and testing data
        train.shape, test.shape

Out[3]: ((12980, 12), (4399, 11))
```

There are 12 columns in train dataset, whereas 11 in the test dataset. The missing column in the test dataset is the target variable and we will train our model to predict that variable.

```
In [4]: # printing first five rows
        train.head()

Out[4]:
```



Out[4]:

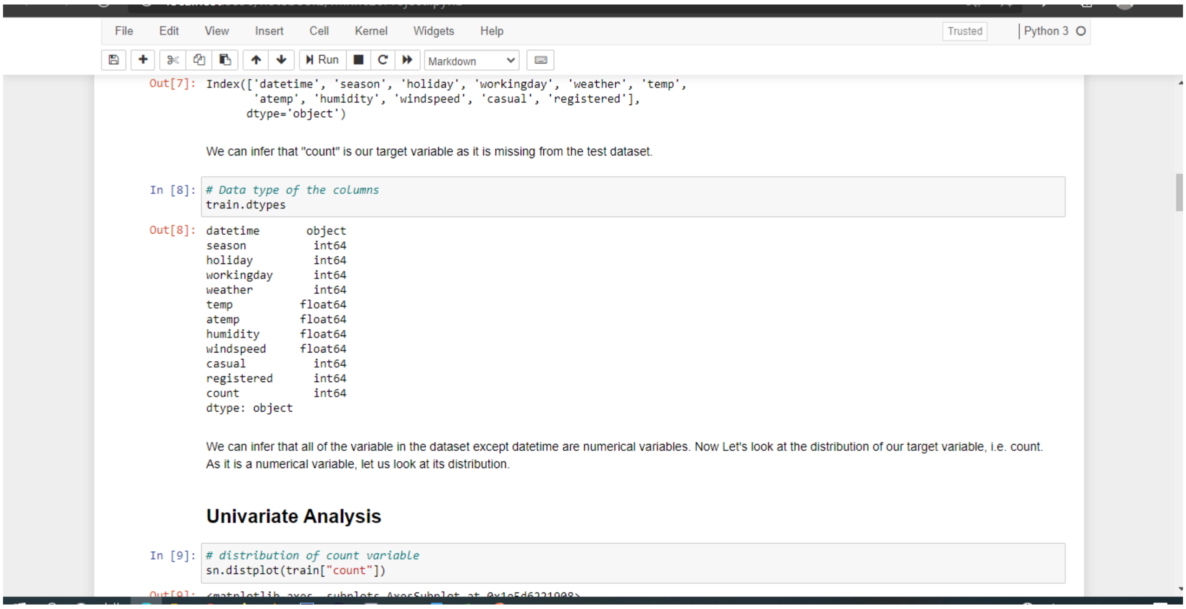
	datetime	season	holiday	workingday	weather	temp	atemp	humidity	windspeed	casual	registered	count
0	2011-01-01 0:00:00	1	0	0	1	9.84	14.395	81.0	0.0	3	13	16
1	2011-01-01 1:00:00	1	0	0	1	9.02	13.635	80.0	0.0	8	32	40
2	2011-01-01 2:00:00	1	0	0	1	9.02	13.635	80.0	0.0	5	27	32
3	2011-01-01 3:00:00	1	0	0	1	9.84	14.395	75.0	0.0	3	10	13
4	2011-01-01 4:00:00	1	0	0	1	9.84	14.395	75.0	0.0	0	1	1

```
In [5]: test.head()
```

Out[5]:

	datetime	season	holiday	workingday	weather	temp	atemp	humidity	windspeed	casual	registered
0	2012-06-30 1:00:00	3	0	0	3	26.24	28.790	89.0	15.0013	3	55
1	2012-06-30 2:00:00	3	0	0	2	26.24	28.790	89.0	0.0000	7	54
2	2012-06-30 3:00:00	3	0	0	2	26.24	28.790	89.0	0.0000	3	20
3	2012-06-30 4:00:00	3	0	0	2	25.42	27.275	94.0	0.0000	3	15
4	2012-06-30 5:00:00	3	0	0	1	26.24	28.790	89.0	11.0014	3	7

```
In [6]: # columns in the dataset
        train.columns

Out[6]: Index(['datetime', 'season', 'holiday', 'workingday', 'weather', 'temp',
               'atemp', 'humidity', 'windspeed', 'casual', 'registered', 'count'],
              dtype='object')

In [7]: test.columns

Out[7]: Index(['datetime', 'season', 'holiday', 'workingday', 'weather', 'temp',
```



```
Out[7]: Index(['datetime', 'season', 'holiday', 'workingday', 'weather', 'temp',
       'atemp', 'humidity', 'windspeed', 'casual', 'registered'],
      dtype='object')
```

We can infer that "count" is our target variable as it is missing from the test dataset.

```
In [8]: # Data type of the columns
        train.dtypes
```

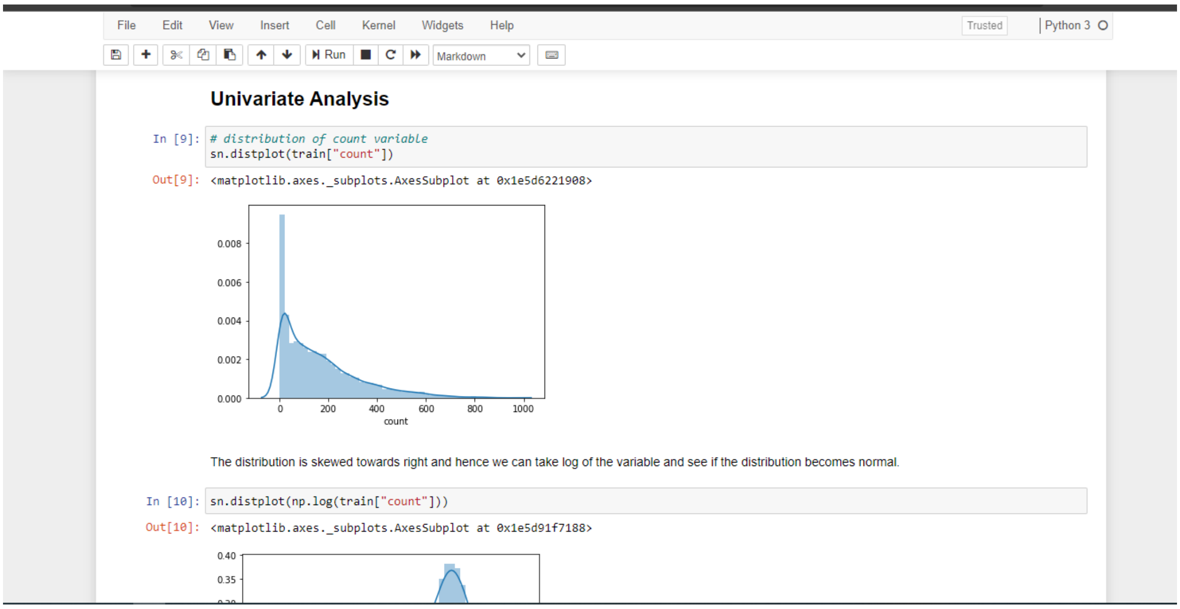
```
Out[8]: datetime       object
        season          int64
        holiday         int64
        workingday      int64
        weather         int64
        temp          float64
        atemp         float64
        humidity      float64
        windspeed     float64
        casual          int64
        registered      int64
        count           int64
        dtype: object
```

We can infer that all of the variable in the dataset except datetime are numerical variables. Now Let's look at the distribution of our target variable, i.e. count. As it is a numerical variable, let us look at its distribution.

Univariate Analysis

```
In [9]: # distribution of count variable
        sn.distplot(train["count"])
```

```
Out[9]: <matplotlib.axes._subplots.AxesSubplot at 0x1e5d6221908>
```



Univariate Analysis

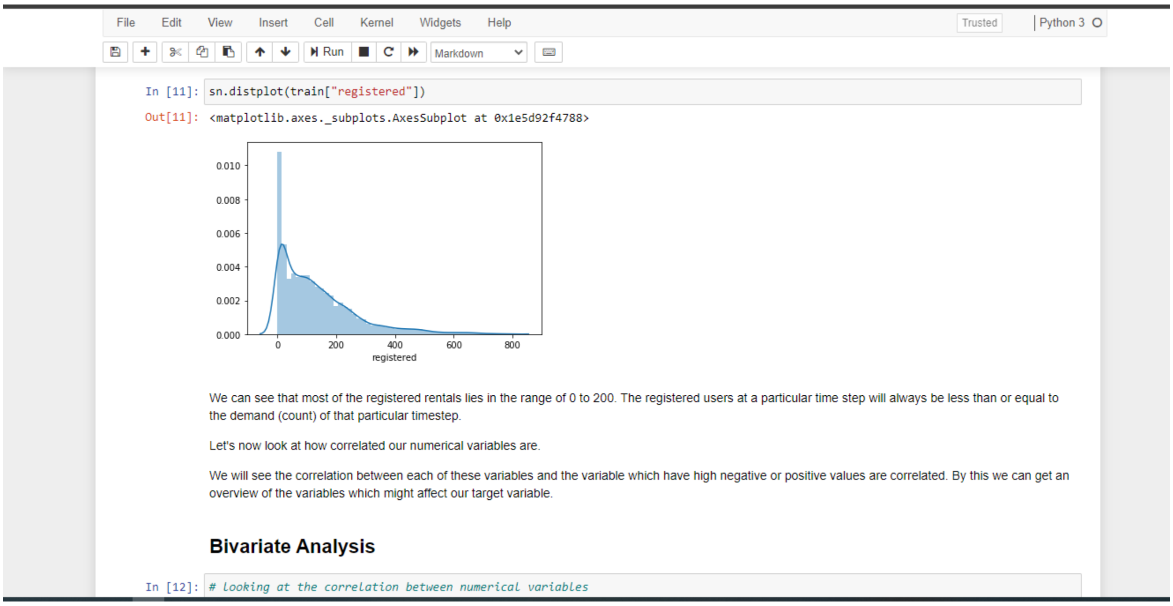
```
In [9]: # distribution of count variable
        sn.distplot(train["count"])
```

```
Out[9]: <matplotlib.axes._subplots.AxesSubplot at 0x1e5d6221908>
```

The distribution is skewed towards right and hence we can take log of the variable and see if the distribution becomes normal.

```
In [10]: sn.distplot(np.log(train["count"]))
```

```
Out[10]: <matplotlib.axes._subplots.AxesSubplot at 0x1e5d91f7188>
```



```
In [11]: sn.distplot(train["registered"])
Out[11]: <matplotlib.axes._subplots.AxesSubplot at 0x1e5d92f4788>
```

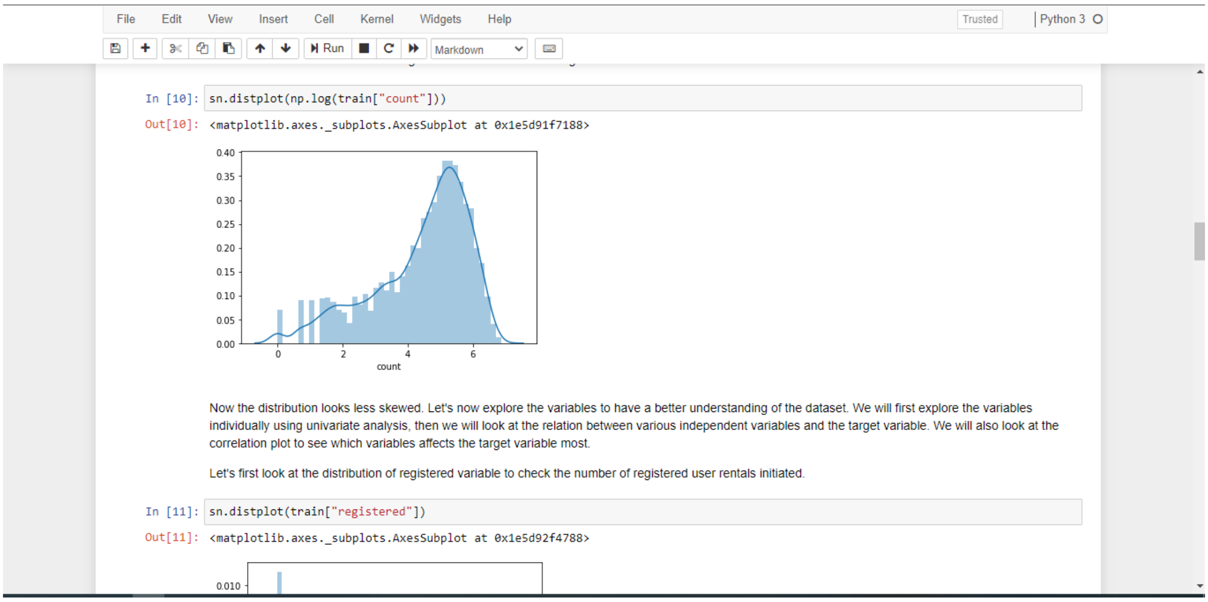
We can see that most of the registered rentals lies in the range of 0 to 200. The registered users at a particular time step will always be less than or equal to the demand (count) of that particular timestep.

Let's now look at how correlated our numerical variables are.

We will see the correlation between each of these variables and the variable which have high negative or positive values are correlated. By this we can get an overview of the variables which might affect our target variable.

Bivariate Analysis

```
In [12]: # looking at the correlation between numerical variables
```

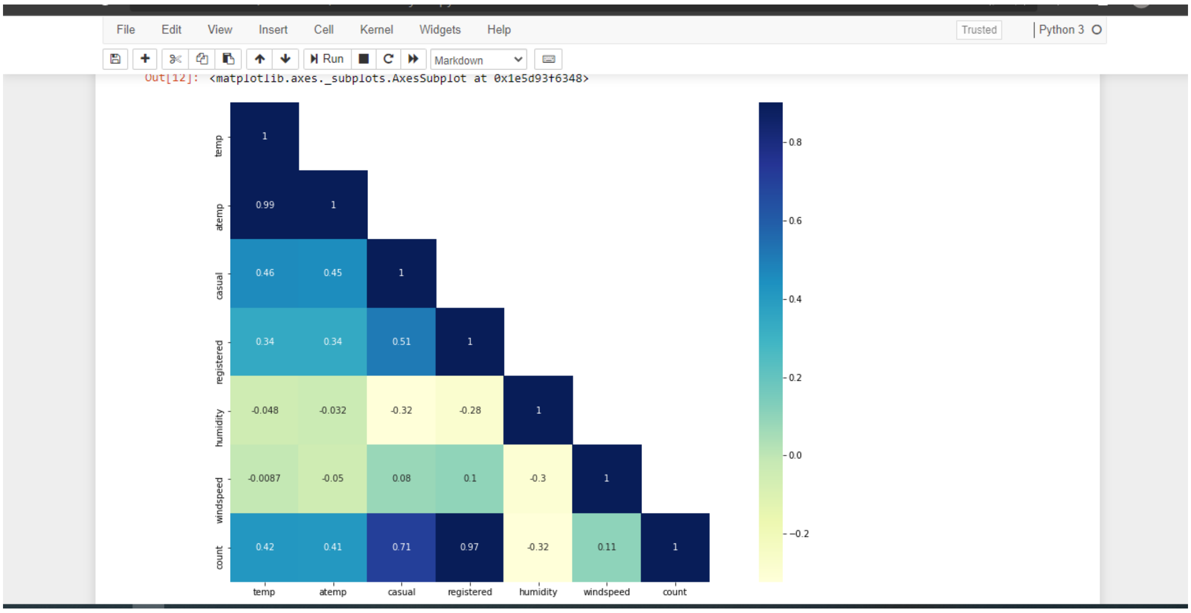


```
In [10]: sn.distplot(np.log(train["count"]))
Out[10]: <matplotlib.axes._subplots.AxesSubplot at 0x1e5d91f7188>
```

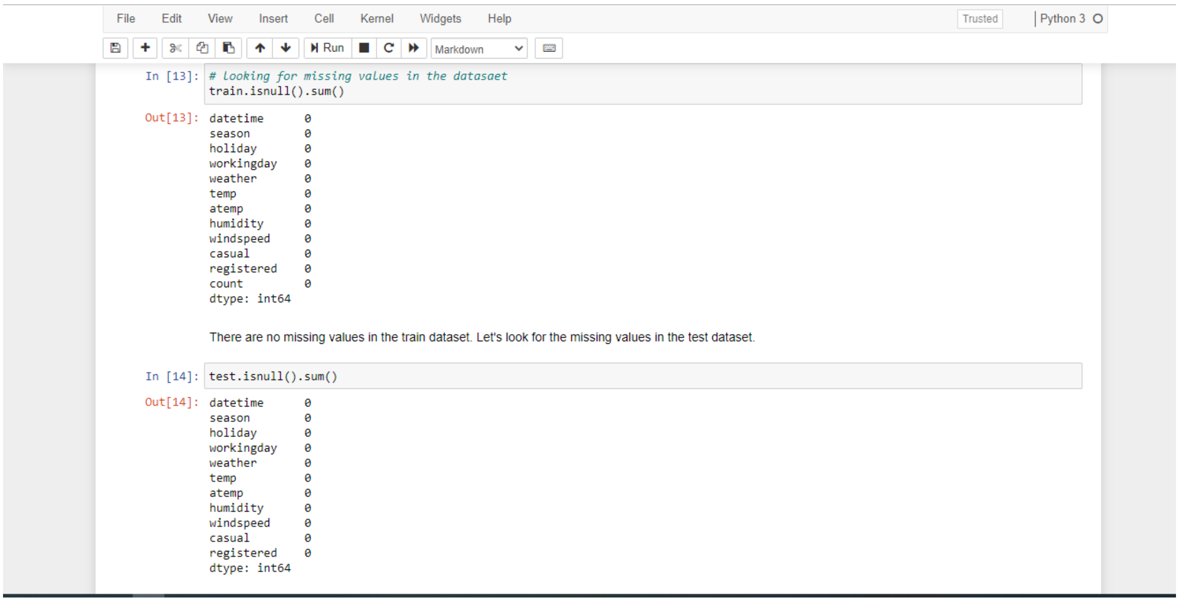
Now the distribution looks less skewed. Let's now explore the variables to have a better understanding of the dataset. We will first explore the variables individually using univariate analysis, then we will look at the relation between various independent variables and the target variable. We will also look at the correlation plot to see which variables affects the target variable most.

Let's first look at the distribution of registered variable to check the number of registered user rentals initiated.

```
In [11]: sn.distplot(train["registered"])
Out[11]: <matplotlib.axes._subplots.AxesSubplot at 0x1e5d92f4788>
```



```
Out[12]: <matplotlib.axes._subplots.AxesSubplot at 0x1e5d93f6348>
```



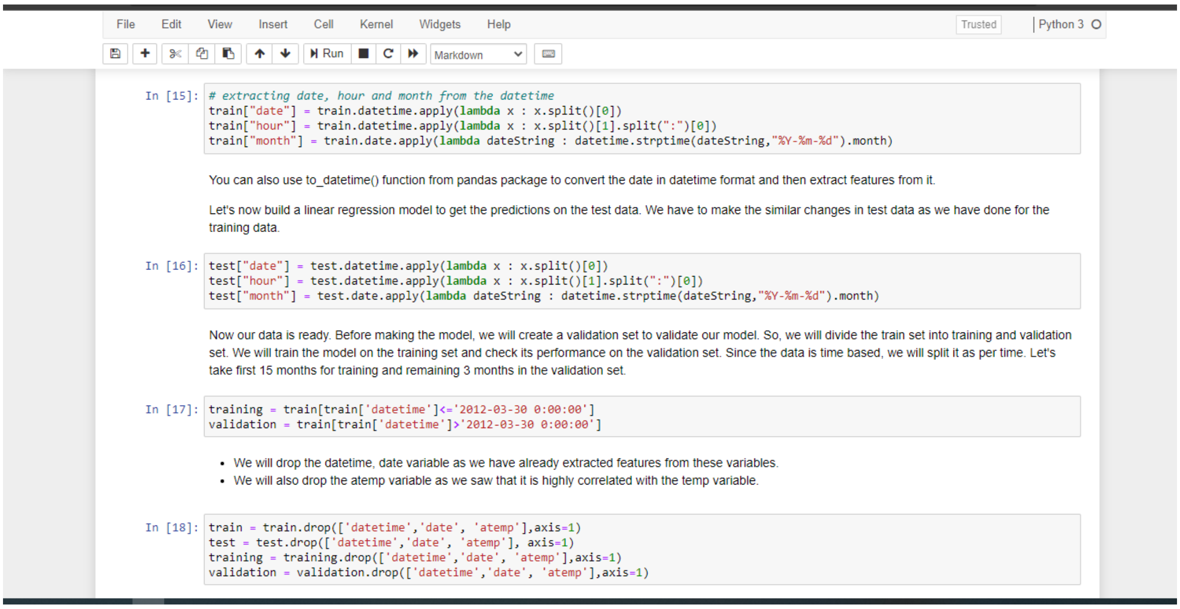
```
In [13]: # Looking for missing values in the datasaet
         train.isnull().sum()
```

```
Out[13]: datetime      0
         season        0
         holiday       0
         workingday    0
         weather       0
         temp          0
         atemp         0
         humidity      0
         windspeed     0
         casual        0
         registered    0
         count         0
         dtype: int64
```

There are no missing values in the train dataset. Let's look for the missing values in the test dataset.

```
In [14]: test.isnull().sum()
```

```
Out[14]: datetime      0
         season        0
         holiday       0
         workingday    0
         weather       0
         temp          0
         atemp         0
         humidity      0
         windspeed     0
         casual        0
         registered    0
         dtype: int64
```



```
In [15]: # extracting date, hour and month from the datetime
         train["date"] = train.datetime.apply(lambda x : x.split()[0])
         train["hour"] = train.datetime.apply(lambda x : x.split()[1].split(":")[0])
         train["month"] = train.date.apply(lambda dateString : datetime.strptime(dateString,"%Y-%m-%d").month)
```

You can also use to_datetime() function from pandas package to convert the date in datetime format and then extract features from it.

Let's now build a linear regression model to get the predictions on the test data. We have to make the similar changes in test data as we have done for the training data.

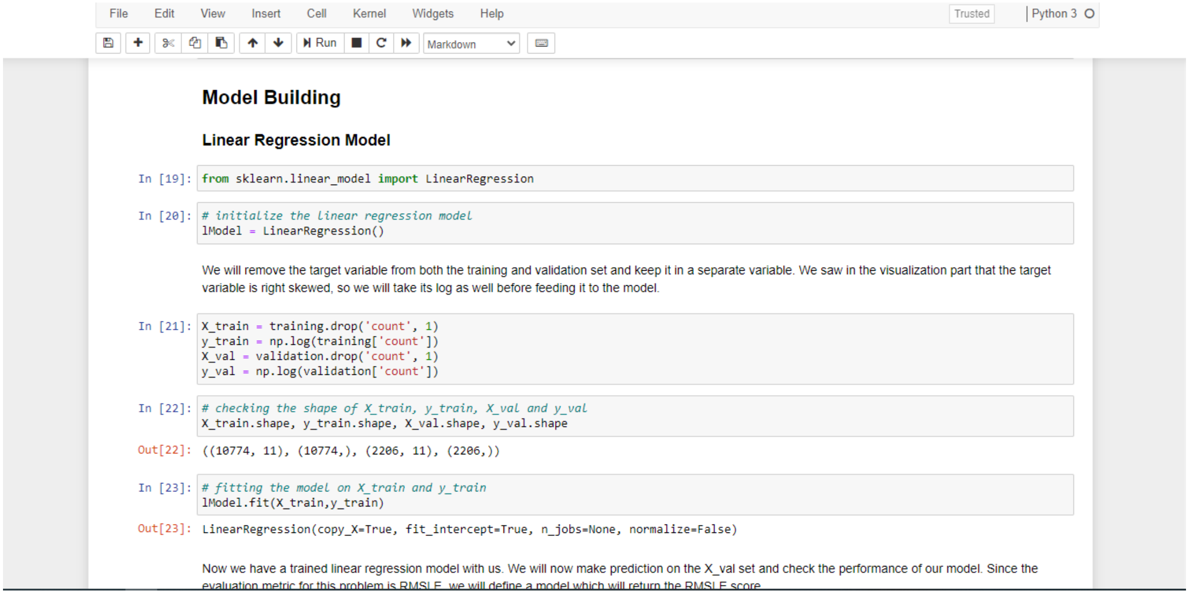
```
In [16]: test["date"] = test.datetime.apply(lambda x : x.split()[0])
         test["hour"] = test.datetime.apply(lambda x : x.split()[1].split(":")[0])
         test["month"] = test.date.apply(lambda dateString : datetime.strptime(dateString,"%Y-%m-%d").month)
```

Now our data is ready. Before making the model, we will create a validation set to validate our model. So, we will divide the train set into training and validation set. We will train the model on the training set and check its performance on the validation set. Since the data is time based, we will split it as per time. Let's take first 15 months for training and remaining 3 months in the validation set.

```
In [17]: training = train[train['datetime']<='2012-03-30 0:00:00']
         validation = train[train['datetime']>'2012-03-30 0:00:00']
```

- We will drop the datetime, date variable as we have already extracted features from these variables.
- We will also drop the atemp variable as we saw that it is highly correlated with the temp variable.

```
In [18]: train = train.drop(['datetime','date', 'atemp'],axis=1)
         test = test.drop(['datetime','date', 'atemp'], axis=1)
         training = training.drop(['datetime','date', 'atemp'],axis=1)
         validation = validation.drop(['datetime','date', 'atemp'],axis=1)
```



Model Building

Linear Regression Model

```
In [19]: from sklearn.linear_model import LinearRegression
```

```
In [20]: # initialize the linear regression model
         lModel = LinearRegression()
```

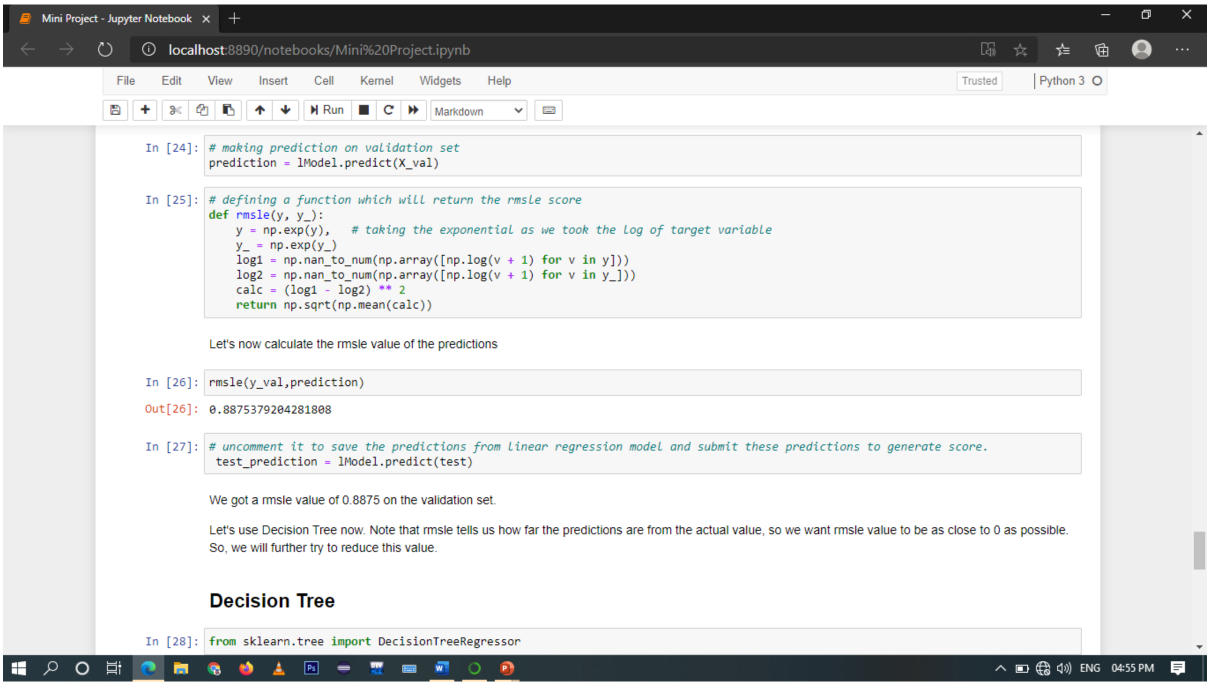
We will remove the target variable from both the training and validation set and keep it in a separate variable. We saw in the visualization part that the target variable is right skewed, so we will take its log as well before feeding it to the model.

```
In [21]: X_train = training.drop('count', 1)
         y_train = np.log(training['count'])
         X_val = validation.drop('count', 1)
         y_val = np.log(validation['count'])
```

```
In [22]: # checking the shape of X_train, y_train, X_val and y_val
         X_train.shape, y_train.shape, X_val.shape, y_val.shape

Out[22]: ((10774, 11), (10774,), (2206, 11), (2206,))
```

```
In [23]: # fitting the model on X_train and y_train
         lModel.fit(X_train,y_train)

Out[23]: LinearRegression(copy_X=True, fit_intercept=True, n_jobs=None, normalize=False)
```

Now we have a trained linear regression model with us. We will now make prediction on the X_val set and check the performance of our model. Since the evaluation metric for this problem is RMSLE, we will define a model which will return the RMSLE score.


Mini Project - Jupyter Notebook
localhost:8890/notebooks/Mini%20Project.ipynb
File
Edit
View
Insert
Cell
Kernel
Widgets
Help
Trusted
Python 3
Run
Markdown
ENG
04:55 PM

```
In [24]: # making prediction on validation set
         prediction = lModel.predict(X_val)
```

```
In [25]: # defining a function which will return the rmsle score
         def rmsle(y, y_):
             y = np.exp(y),   # taking the exponential as we took the log of target variable
             y_ = np.exp(y_)
             log1 = np.nan_to_num(np.array([np.log(v + 1) for v in y]))
             log2 = np.nan_to_num(np.array([np.log(v + 1) for v in y_]))
             calc = (log1 - log2) ** 2
             return np.sqrt(np.mean(calc))
```

Let's now calculate the rmsle value of the predictions

```
In [26]: rmsle(y_val,prediction)

Out[26]: 0.8875379204281808
```

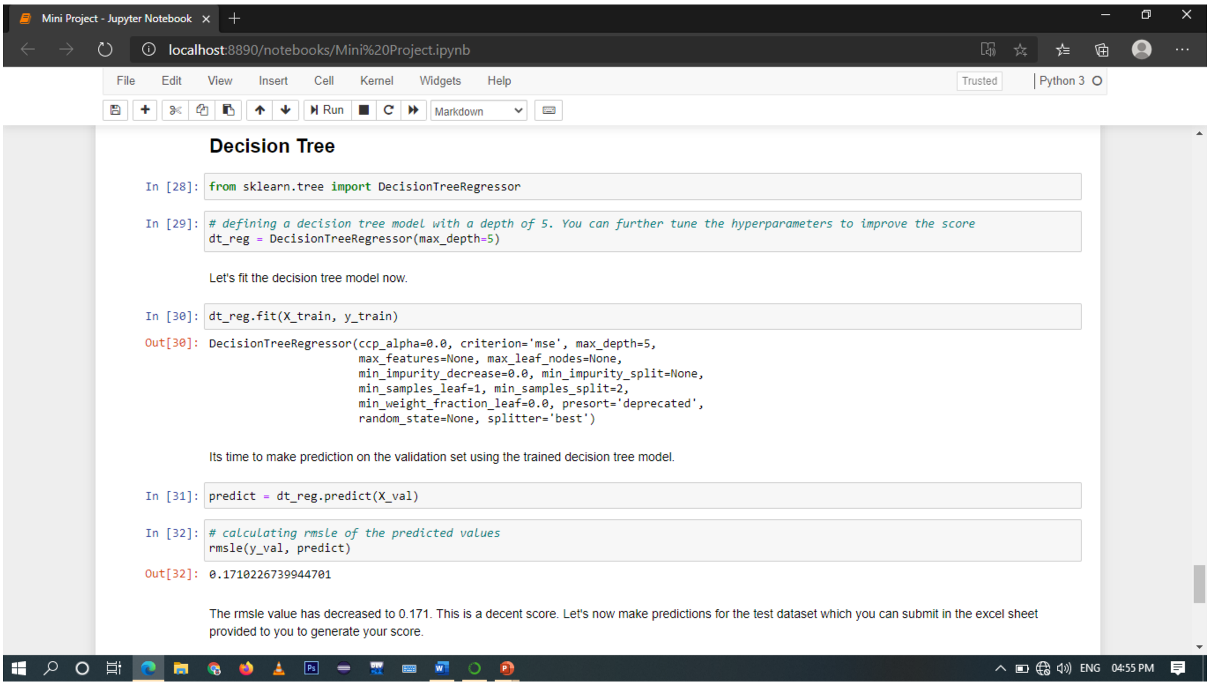
```
In [27]: # uncomment it to save the predictions from linear regression model and submit these predictions to generate score.
          test_prediction = lModel.predict(test)
```

We got a rmsle value of 0.8875 on the validation set.

Let's use Decision Tree now. Note that rmsle tells us how far the predictions are from the actual value, so we want rmsle value to be as close to 0 as possible. So, we will further try to reduce this value.

Decision Tree

```
In [28]: from sklearn.tree import DecisionTreeRegressor
```


Mini Project - Jupyter Notebook
localhost:8890/notebooks/Mini%20Project.ipynb
File
Edit
View
Insert
Cell
Kernel
Widgets
Help
Trusted
Python 3
Run
Markdown
ENG
04:55 PM

Decision Tree

```
In [28]: from sklearn.tree import DecisionTreeRegressor
```

```
In [29]: # defining a decision tree model with a depth of 5. You can further tune the hyperparameters to improve the score
         dt_reg = DecisionTreeRegressor(max_depth=5)
```

Let's fit the decision tree model now.

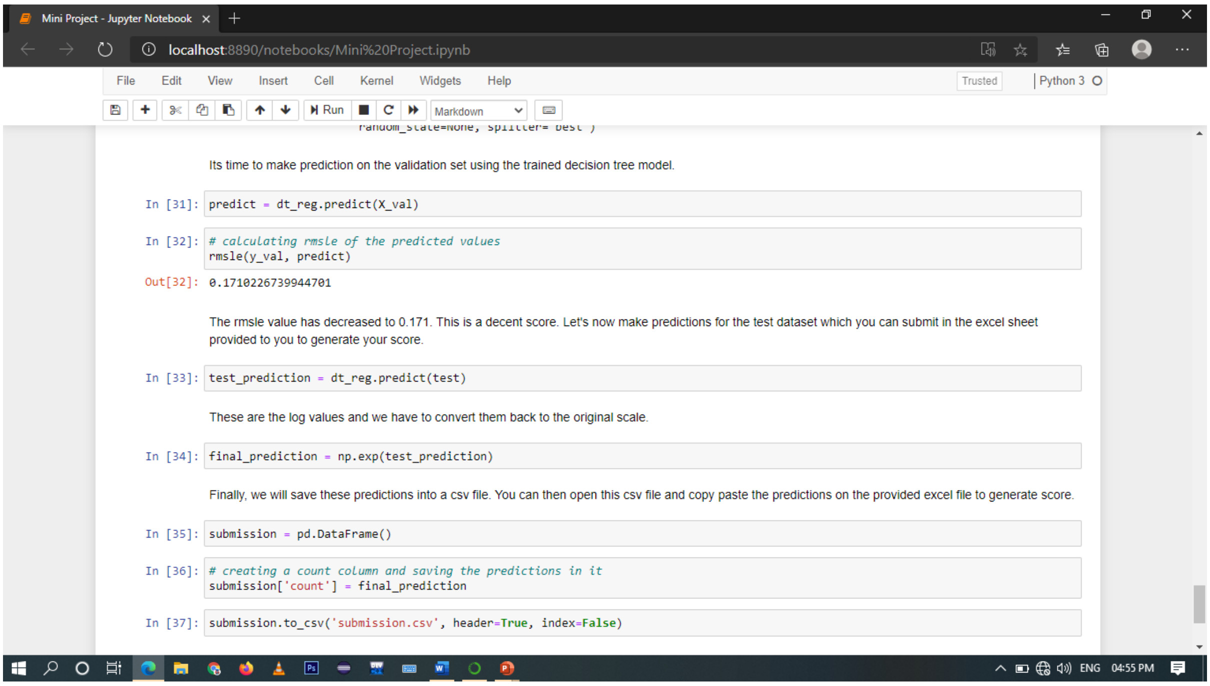
```
In [30]: dt_reg.fit(X_train, y_train)

Out[30]: DecisionTreeRegressor(ccp_alpha=0.0, criterion='mse', max_depth=5,
                               max_features=None, max_leaf_nodes=None,
                               min_impurity_decrease=0.0, min_impurity_split=None,
                               min_samples_leaf=1, min_samples_split=2,
                               min_weight_fraction_leaf=0.0, presort='deprecated',
                               random_state=None, splitter='best')
```

Its time to make prediction on the validation set using the trained decision tree model.

```
In [31]: predict = dt_reg.predict(X_val)
```

```
In [32]: # calculating rmsle of the predicted values
         rmsle(y_val, predict)

Out[32]: 0.1710226739944701
```

The rmsle value has decreased to 0.171. This is a decent score. Let's now make predictions for the test dataset which you can submit in the excel sheet provided to you to generate your score.


Mini Project - Jupyter Notebook
localhost:8890/notebooks/Mini%20Project.ipynb
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3
Run
Markdown

```
random_state=None, splitter='best')
```

Its time to make prediction on the validation set using the trained decision tree model.

```
In [31]: predict = dt_reg.predict(X_val)

In [32]: # calculating rmsle of the predicted values
         rmsle(y_val, predict)

Out[32]: 0.1710226739944701
```

The rmsle value has decreased to 0.171. This is a decent score. Let's now make predictions for the test dataset which you can submit in the excel sheet provided to you to generate your score.

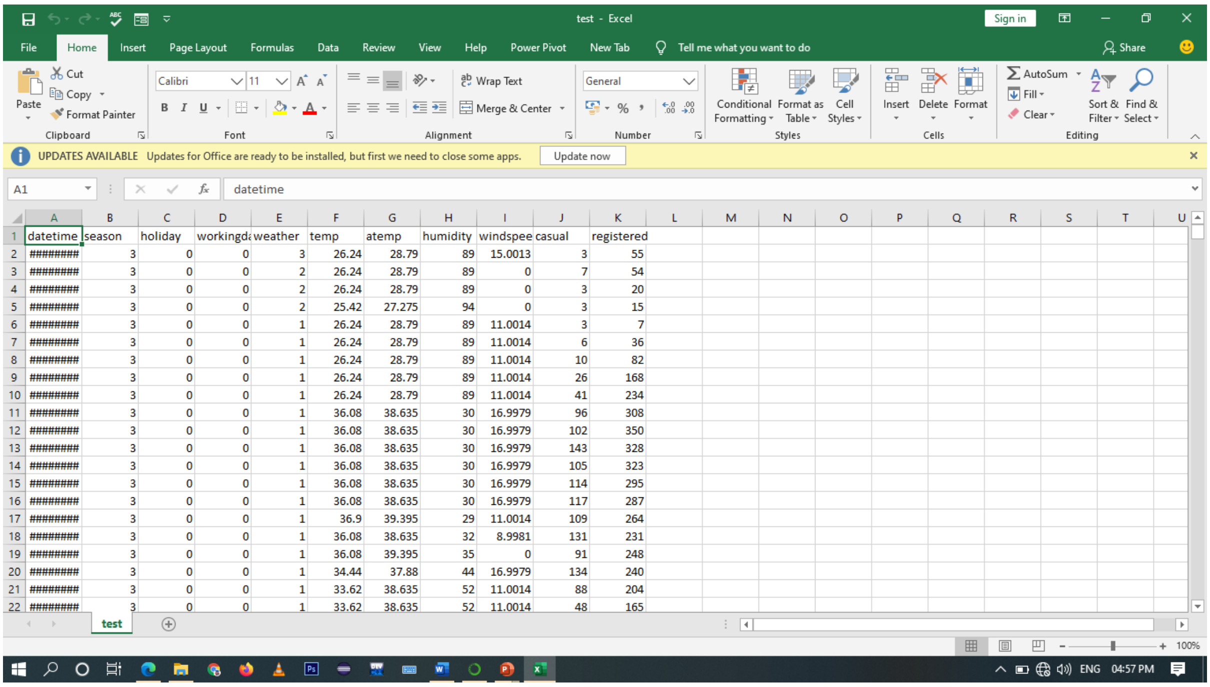
```
In [33]: test_prediction = dt_reg.predict(test)
```

These are the log values and we have to convert them back to the original scale.

```
In [34]: final_prediction = np.exp(test_prediction)
```

Finally, we will save these predictions into a csv file. You can then open this csv file and copy paste the predictions on the provided excel file to generate score.

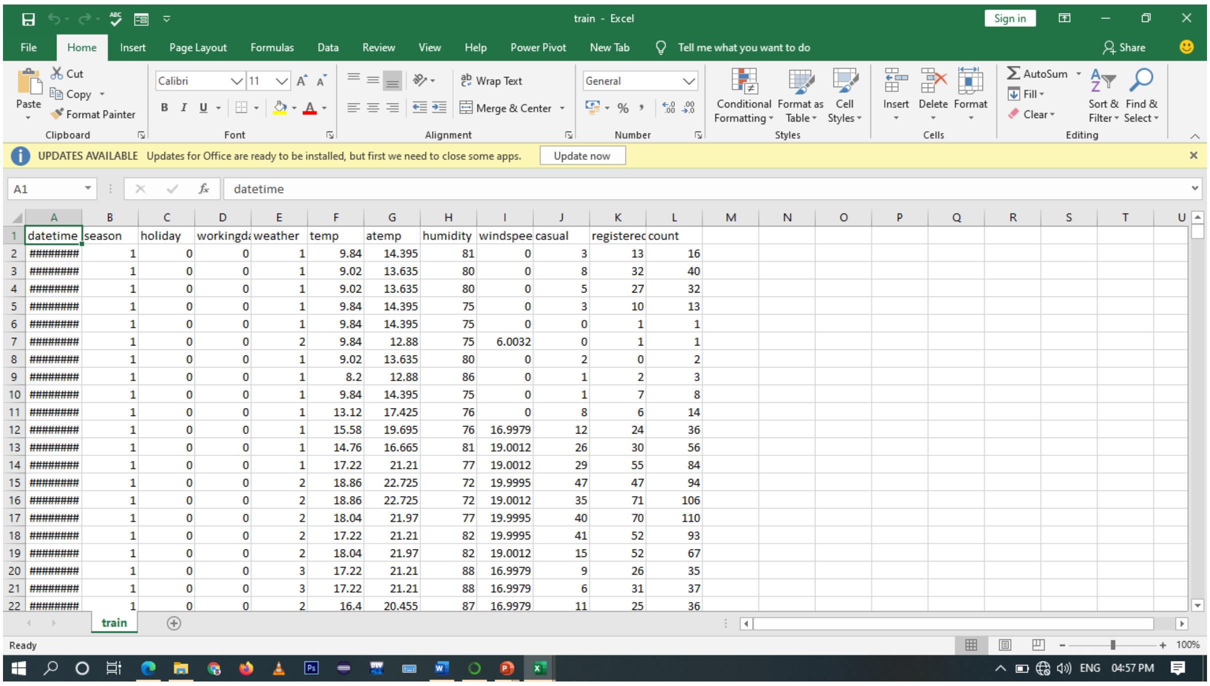
```
In [35]: submission = pd.DataFrame()

In [36]: # creating a count column and saving the predictions in it
         submission['count'] = final_prediction

In [37]: submission.to_csv('submission.csv', header=True, index=False)
```



UPDATES AVAILABLE Updates for Office are ready to be installed, but first we need to close some apps. Update now

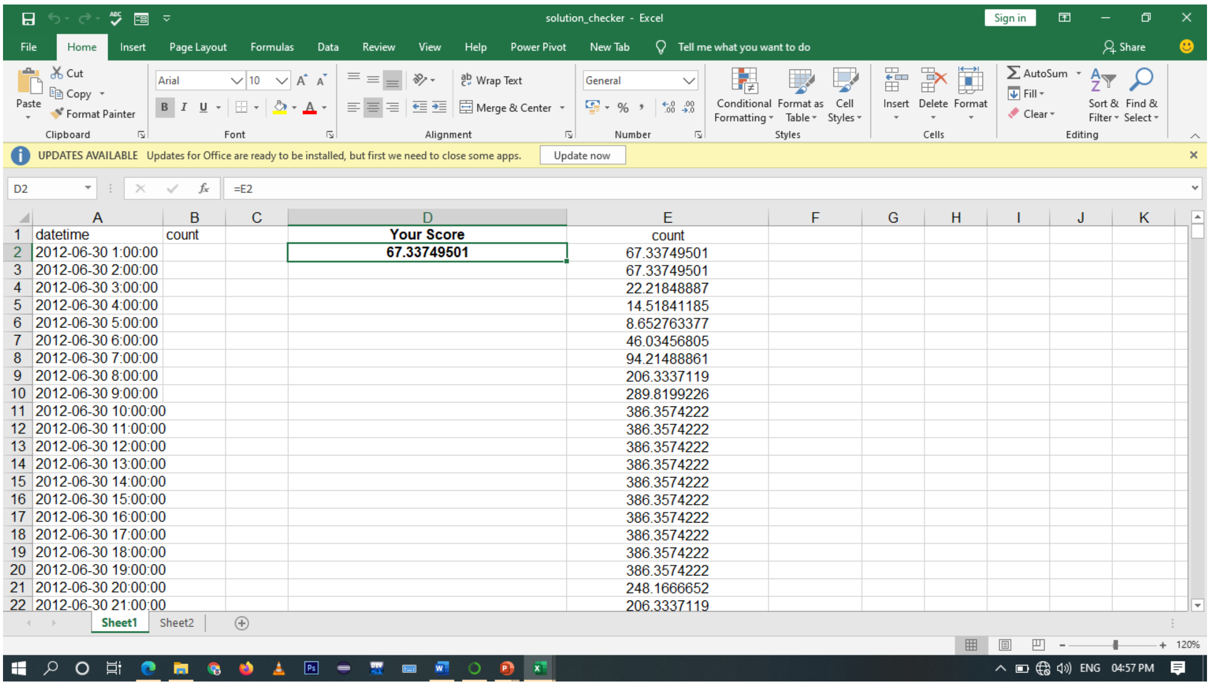
A1 datetime

	A	B	C	D	E	F	G	H	I	J	K
1	datetime	season	holiday	workingda	weather	temp	atemp	humidity	windspee	casual	registered
2	#########	3	0	0	3	26.24	28.79	89	15.0013	3	55
3	#########	3	0	0	2	26.24	28.79	89	0	7	54
4	#########	3	0	0	2	26.24	28.79	89	0	3	20
5	#########	3	0	0	2	25.42	27.275	94	0	3	15
6	#########	3	0	0	1	26.24	28.79	89	11.0014	3	7
7	#########	3	0	0	1	26.24	28.79	89	11.0014	6	36
8	#########	3	0	0	1	26.24	28.79	89	11.0014	10	82
9	#########	3	0	0	1	26.24	28.79	89	11.0014	26	168
10	#########	3	0	0	1	26.24	28.79	89	11.0014	41	234
11	#########	3	0	0	1	36.08	38.635	30	16.9979	96	308
12	#########	3	0	0	1	36.08	38.635	30	16.9979	102	350
13	#########	3	0	0	1	36.08	38.635	30	16.9979	143	328
14	#########	3	0	0	1	36.08	38.635	30	16.9979	105	323
15	#########	3	0	0	1	36.08	38.635	30	16.9979	114	295
16	#########	3	0	0	1	36.08	38.635	30	16.9979	117	287
17	#########	3	0	0	1	36.9	39.395	29	11.0014	109	264
18	#########	3	0	0	1	36.08	38.635	32	8.9981	131	231
19	#########	3	0	0	1	36.08	39.395	35	0	91	248
20	#########	3	0	0	1	34.44	37.88	44	16.9979	134	240
21	#########	3	0	0	1	33.62	38.635	52	11.0014	88	204
22	#########	3	0	0	1	33.62	38.635	52	11.0014	48	165



A1 datetime

	A	B	C	D	E	F	G	H	I	J	K	L
1	datetime	season	holiday	workingda	weather	temp	atemp	humidity	windspee	casual	registerec	count
2	########	1	0	0	1	9.84	14.395	81	0	3	13	16
3	########	1	0	0	1	9.02	13.635	80	0	8	32	40
4	########	1	0	0	1	9.02	13.635	80	0	5	27	32
5	########	1	0	0	1	9.84	14.395	75	0	3	10	13
6	########	1	0	0	1	9.84	14.395	75	0	0	1	1
7	########	1	0	0	2	9.84	12.88	75	6.0032	0	1	1
8	########	1	0	0	1	9.02	13.635	80	0	2	0	2
9	########	1	0	0	1	8.2	12.88	86	0	1	2	3
10	########	1	0	0	1	9.84	14.395	75	0	1	7	8
11	########	1	0	0	1	13.12	17.425	76	0	8	6	14
12	########	1	0	0	1	15.58	19.695	76	16.9979	12	24	36
13	########	1	0	0	1	14.76	16.665	81	19.0012	26	30	56
14	########	1	0	0	1	17.22	21.21	77	19.0012	29	55	84
15	########	1	0	0	2	18.86	22.725	72	19.9995	47	47	94
16	########	1	0	0	2	18.86	22.725	72	19.0012	35	71	106
17	########	1	0	0	2	18.04	21.97	77	19.9995	40	70	110
18	########	1	0	0	2	17.22	21.21	82	19.9995	41	52	93
19	########	1	0	0	2	18.04	21.97	82	19.0012	15	52	67
20	########	1	0	0	3	17.22	21.21	88	16.9979	9	26	35
21	########	1	0	0	3	17.22	21.21	88	16.9979	6	31	37
22	########	1	0	0	2	16.4	20.455	87	16.9979	11	25	36



D2 =E2

	A	B	C	D	E
1	datetime	count		**Your Score**	count
2	2012-06-30 1:00:00			**67.33749501**	67.33749501
3	2012-06-30 2:00:00				67.33749501
4	2012-06-30 3:00:00				22.21848887
5	2012-06-30 4:00:00				14.51841185
6	2012-06-30 5:00:00				8.652763377
7	2012-06-30 6:00:00				46.03456805
8	2012-06-30 7:00:00				94.21488861
9	2012-06-30 8:00:00				206.3337119
10	2012-06-30 9:00:00				289.8199226
11	2012-06-30 10:00:00				386.3574222
12	2012-06-30 11:00:00				386.3574222
13	2012-06-30 12:00:00				386.3574222
14	2012-06-30 13:00:00				386.3574222
15	2012-06-30 14:00:00				386.3574222
16	2012-06-30 15:00:00				386.3574222
17	2012-06-30 16:00:00				386.3574222
18	2012-06-30 17:00:00				386.3574222
19	2012-06-30 18:00:00				386.3574222
20	2012-06-30 19:00:00				386.3574222
21	2012-06-30 20:00:00				248.1666652
22	2012-06-30 21:00:00				206.3337119

CONCLUSION

I here by conclude my project on the topic “Hourly bike rental demand prediction”. I had provided with the sound knowledge of data science and now the required output and result is within satisfactory range with a quite good number of improvements on the project that are possible in this project like improving the user interface adding a few more modules also a few more features like a better database connectivity and connectivity to other systems through the same system program and other.

BIBLIOGRAPHY

In order to complete this project most of the help was provided by the supporting professors from internshala training and internet including many relevant websites and tutorial classes.

I also expand my regards to Prof. Dhiraj Kumar Mishra sir (Asst. Prof. at RVS College of Engineering & Technology) for the guidance and support regarding internships and trainings.

REFERENCES

- https://www.internshala.com/
- www.udemy.com