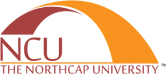


|  |  |  |
| --- | --- | --- |
| **Introduction to AI & ML** | | |
| Lab Manual | | |
| **Department of Computer Science and Engineering**  **The NorthCap University, Gurugram** | | |
|  | |
|  | |
|  | |
|  | |
|  |

**Introduction to AI &ML Lab Manual**

**CSL 236p**



Department of Computer Science and Engineering

NorthCap University, Gurugram- 122001, India

Session 2021-22

*Published by:*

**School of Engineering and Technology**

**Department of Computer Science & Engineering**

**The NorthCap University Gurugram**

**• Laboratory Manual is for Internal Circulation only**

**© Copyright Reserved**

*No part of this Practical Record Book may be*

*reproduced, used, stored without prior permission of The NorthCap University*



Copying or facilitating copying of lab work comes under cheating and is considered as use of unfair means. Students indulging in copying or facilitating copying shall be awarded zero marks for that particular experiment. Frequent cases of copying may lead to disciplinary action. Attendance in lab classes is mandatory.

Labs are open up to 7 PM upon request. Students are encouraged to make full use of labs beyond normal lab hours.

**PREFACE**

Machine Learning Lab Manual is designed to meet the course and program requirements of NCU curriculum for B.Tech III year students of CSE branch. The concept of the lab work is to give brief practical experience for basic lab skills to students. It provides the space and scope for self-study so that students can come up with new and creative ideas.

The Lab manual is written on the basis of “teach yourself pattern” and expected that students who come with proper preparation should be able to perform the experiments without any difficulty. Brief introduction to each experiment with information about self-study material is provided. The pre-requisite is having a basic working knowledge of Python. The laboratory exercises will include familiarization with data pre-processing techniques for ML like handling missing data, duplicate data, outliers feature scaling and encoding. Feature Selection and Dimensionality Reduction are included to enhance the performance and reduce the computational time. Various ML classification and regression techniques are taught. Students would learn the algorithms pertaining to these and implement the same using a high-level language, i.e. Python. Students are expected to come thoroughly prepared for the lab. General disciplines, safety guidelines and report writing are also discussed.

The lab manual is a part of curriculum for the The NorthCap University, Gurugram. Teacher’s copy of the experimental results and answer for the questions are available as sample guidelines.

We hope that lab manual would be useful to students of CSE, IT, ECE and BSc branches and author requests the readers to kindly forward their suggestions / constructive criticism for further improvement of the work book.

Author expresses deep gratitude to Members, Governing Body-NCU for encouragement and motivation.

**Authors**

**The NorthCap University**

**Gurugram, India**

|  |  |  |  |
| --- | --- | --- | --- |
| **CONTENTS** | | | |
| S.N. | **Details** | Page No. | |
|  | **Syllabus** | 6 | |
| 1 | **Introduction** | 9 | |
| 2 | **Lab Requirement** | 10 |
| 3 | **General Instructions** | 11 |
| 4 | **List of Experiments** | 13 |
| 5 | **List of Flip Assignment** | 14 |
| 6 | **List of Projects** | 15 |
| 7 | **Rubrics** | 16 |
| 8 | **Annexure 1 (Format of Lab Report)** | 18 |

**SYLLABUS**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1. **Department:** | | | Department of Computer Science and Engineering | | | | | | |
| 1. **Course Name: Machine Learning** | | | | | 1. **Course Code** | 1. **L-T-P** | | | 1. **Credits** |
| CSL236 | 3-0-2 | | | 4 |
| 1. **Type of Course (Check one):** | | | Programme Core  Programme Elective **✔** Open Elective | | | | | | |
| 1. **Pre-requisite(s), if any:** Introduction to AI and ML | | | | | | | | | |
| 1. **Frequency of offering (check one):** Odd  Even **✔** Either semester Every semester | | | | | | | | | |
| 1. **Brief Syllabus:**   Introduction to artificial intelligence, History of AI, Proposing and evaluating AI application, Preprocessing and Feature Engineering, Case study: Exploratory Analysis of Delhi Pollution, Simple Linear Regression, Multiple Regression, Polynomial Regression, Support Vector Regression SVR, Decision Tree Regression, Random Forest Regression, Logistic Regression, K Nearest Neighbors, Support Vector Machine, Kernel SVM, Naïve Bayes, Decision Trees Classification, Random Forest Classification, Basic Terminologies: Overfitting, Underfitting, Bias and Variance model, Bootstrapping, Cross-Validation and Resampling Methods, Performance Measures: Confusion matrix, ROC. Comparing two classification Algorithms:  McNamara’s Test, paired t-test. | | | | | | | | | |
| **Total lecture, Tutorial and Practical Hours for this course (Take 15 teaching weeks per semester): 75** hours  The class size is maximum 30 learners | | | | | | | | | |
| **Lectures: 40** hours | | | | **Practice** | | | | | |
| **Tutorials:** 0 hours | | | **Lab Work: 35** hours | | |
| 1. **Course Outcomes (COs)**   On successful completion of this course students will be able to: | | | | | | | | | |
| **CO 1** | Understand and implement the preprocessing of the data to be used for machine learning models. | | | | | | | | |
| **CO 2** | Understand the strengths and limitations of various ML algorithms. | | | | | | | | |
| **CO 3** | Understand why models degrade and how to maintain them. | | | | | | | | |
| **CO 4** | Implement and use model grading metrics. | | | | | | | | |
| **CO 5** | Apply ML techniques and technologies to solve real world business problems. | | | | | | | | |
| 1. **UNIT WISE DETAILS No. of Units: 5** | | | | | | | | | |
| **Unit Number: 1** | | **Title: Introduction to AI and ML** | | | | | | **No. of hours: 4** | |
| **Content Summary:**  Introduction to artificial intelligence, History of AI, Overview of machine learning, techniques in machine learning, deep learning, differences between deep learning, machine learning and AI, different applications of machine learning, different types of data. | | | | | | | | | |
| **Unit Number: 2** | | **Data preprocessing and engineering** | | | | | | **No. of hours:10** | |
| Content Summary:  Introduction to Data Preprocessing, different preprocessing techniques, data cleaning, data transformation: standardization and normalization, data smoothing, dimensionality reduction, different encoding schemes for categorical and numerical features. | | | | | | | | | |
| **Unit Number: 3** | | **Title: Regression Techniques** | | | | | | **No. of hours:12** | |
| **Content Summary:**  Simple Linear Regression, Multiple Regression, Polynomial Regression, Support Vector Regression SVR, Decision Tree Regression, Random Forest Regression | | | | | | | | | |
| **Unit Number: 4** | | **Title: Classification algorithm techniques** | | | | | | **No. of hours:12** | |
| Logistic Regression, K Nearest Neighbors, Support Vector Machine, Kernel SVM, Naïve Bayes, Decision Trees Classification, Random Forest Classification | | | | | | | | | |
| **Unit Number: 5** | | **Title: Analysis of various algorithms** | | | | | | **No. of hours:7** | |
| Basic Terminologies: Over fitting, Under fitting, Bias and Variance model, Bootstrapping, Cross-Validation and Resampling Methods, Performance Measures: Confusion matrix, ROC. | | | | | | | | | |
| 1. **Brief Description of Self-learning components by students (through books/resource material etc.):**   Data-preprocessing techniques | | | | | | | | | |
| **13. Advance Learning Components:**  Probability and Statistics and Linear Algebra | | | | | | | | | |
| **14. Books Recommended :**  **Text Books:**  1. Michael Bowles, “Machine Learning in Python ” Wiley, Third Edition, 2019 2. Jiawei Han, Micheline Kamber and Jian Pei. Data Mining: Concepts and Techniques, 3rd ed.  **Reference Books:**  1. Ian H. Witten & Eibe Frank., “Data Mining Practical Machine Learning Tools and Techniques”, Morgan Kauffmann Publishers, Second Edition, 2010  2. Ethem Alpaydin, “Introduction to Machine Learning”, MIT Press, Third Edition, 2015  3. Tom Mitchell. Machine Learning. Mc Graw Hill  **Reference Websites: (NPTEL, Swayam, Coursera, Edx, Udemy, LMS, official documentation weblink)**   * [**https://nculms.ncuindia.edu/**](https://nculms.ncuindia.edu/) * [**https://www.simplilearn.com/big-data-and-analytics/machine-learning-certification-training-course**](https://www.simplilearn.com/big-data-and-analytics/machine-learning-certification-training-course) * [**https://www.coursera.org/learn/machine-learning**](https://www.coursera.org/learn/machine-learning) | | | | | | | | | |

1. **INTRODUCTION**

That ‘learning is a continuous process’ cannot be over emphasized. The theoretical knowledge gained during lecture sessions need to be strengthened through practical experimentation. Thus, practical makes an integral part of a learning process.

**OBJECTIVE:**

The purpose of conducting experiments can be stated as follows:

* To familiarize the students with the basic concepts of Machine Learning like supervised, unsupervised and reinforcement learning.
* The lab sessions will be based on exploring the concepts discussed in class.
* Learning and understanding Data Preprocessing techniques.
* Learning and understanding regression and classification problems and algorithms.
* Learning and understanding Feature selection and Dimensionality Reduction.
* Learning and understanding performance metrics.
* Hands on experience

1. **LAB REQUIREMENTS**

|  |  |  |
| --- | --- | --- |
| **S.No.** | **Requirements** | **Details** |
| **1** | **Software Requirements** | Python 3. |
| **2** | **Operating System** | Windows(64-bit), Linux |
| **3** | **Hardware Requirements** | 8 GB RAM (Recommended)  2.60 GHz (Recommended) |
| **4** | **Required Bandwidth** | NA |

1. **GENERAL INSTRUCTIONS** 
   1. **General discipline in the lab**
   * Students must turn up in time and contact concerned faculty for the experiment they are supposed to perform.
   * Students will not be allowed to enter late in the lab.
   * Students will not leave the class till the period is over.
   * Students should come prepared for their experiment.
   * Experimental results should be entered in the lab report format and certified/signed by concerned faculty/ lab Instructor.
   * Students must get the connection of the hardware setup verified before switching on the power supply.
   * Students should maintain silence while performing the experiments. If any necessity arises for discussion amongst them, they should discuss with a very low pitch without disturbing the adjacent groups.
   * Violating the above code of conduct may attract disciplinary action.
   * Damaging lab equipment or removing any component from the lab may invite penalties and strict disciplinary action.
   1. **Attendance**

* Attendance in the lab class is compulsory.
* Students should not attend a different lab group/section other than the one assigned at the beginning of the session.
* On account of illness or some family problems, if a student misses his/her lab classes, he/she may be assigned a different group to make up the losses in consultation with the concerned faculty / lab instructor. Or he/she may work in the lab during spare/extra hours to complete the experiment. No attendance will be granted for such case**.**
  1. **Preparation and Performance**
* Students should come to the lab thoroughly prepared on the experiments they are assigned to perform on that day. Brief introduction to each experiment with information about self study reference is provided on LMS.
* Students must bring the lab report during each practical class with written records of the last experiments performed complete in all respect.
* Each student is required to write a complete report of the experiment he has performed and bring to lab class for evaluation in the next working lab. Sufficient space in work book is provided for independent writing of theory, observation, calculation and conclusion.
* Students should follow the Zero tolerance policy for copying / plagiarism. Zero marks will be awarded if found copied. If caught further, it will lead to disciplinary action.
* Refer **Annexure 1** for Lab Report Format

1. **LIST OF EXPERIMENTS**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Sr. No.** | **Title of the Experiment** | **Software**  **used** | **Unit**  **Covered** | **CO**  **Covered** | **Time**  **Required** |
|  | To introduce various python libraries used for machine learning. | Python  (Jupyter) | 1 | CO1 | 4 hours |
|  | To apply various data pre-processing techniques used for effective machine learning on the given dataset. | Python  (Jupyter) | 1 | CO1 | 2 hours |
|  | To apply feature encoding schemes such as label encoder and onehotencoder. | Python  (Jupyter) | 1 | CO1 | 3 hours |
|  | To apply different feature selection techniques in machine learning. | Python  (Jupyter) | 1 | CO1 | 3 hours |
|  | To apply PCA as feature reduction technique on IRIS dataset. | Python  (Jupyter) | 2 | CO1 | 2 hours |
|  | To apply Simple Linear Regression on the given dataset. | Python  (Jupyter) | 2 | CO2, CO3, CO4 | 2 hours |
|  | To apply multiple linear regression on any regression dataset. | Python  (Jupyter) | 2 | CO2,CO3,CO4 | 3 hours |
|  | To apply Polynomial Linear Regression on the given dataset. | Python  (Jupyter) | 2 | CO2,CO3,CO4 | 3 hours |
|  | To solve classification problems using Logistic Regression. | Python  (Jupyter) | 3 | CO2,CO3,CO4 | 2 hours |
|  | To solve classification problems using KNN classification. | Python  (Jupyter) | 3 | CO2,CO3,CO4 | 2 hours |
|  | To solve classification problems using Naïve Bayes. | Python  (Jupyter) | 4 | CO2,CO3,CO4 | 2 hours |
|  | To apply Support Vector Machines (SVM) on classification problems. | Python  (Jupyter) | 4 | CO2,CO3,CO4 | 3 hours |
|  | To apply Decision Trees for classification problems. | Python  (Jupyter) | 4 | CO2,CO3,CO4 | 3 hours |
| **Value Added Experiments** | | | | | |
|  | Build a ML model from scratch using data-preprocessing and regression algorithms and calculating various performance metrics. | Python  (Jupyter) | 1,2,3,4 | CO1,CO2,  CO3,CO4,CO5 | 5 hours |
|  | Build a ML model from scratch using data-preprocessing and classification algorithms and calculating various performance metrics. | Python  (Jupyter) | 1,2,3,4 | CO1,CO2,  CO3,CO4,CO5 | 5 hours |

1. **LIST OF FLIP EXPERIMENTS**
2. Project – Dimensionality reduction using LDA.
3. Competition on Kaggle
4. **LIST OF PROJECTS**
   * + 1. Titanic Challenge: The sinking of the Titanic is one of the most infamous shipwrecks in history.On April 15, 1912, during her maiden voyage, the widely considered “unsinkable” RMS Titanic sank after colliding with an iceberg. Unfortunately, there weren’t enough lifeboats for everyone onboard, resulting in the death of 1502 out of 2224 passengers and crew.While there was some element of luck involved in surviving, it seems some groups of people were more likely to survive than others.In this project, the students need to build a predictive model that answers the question: “what sorts of people were more likely to survive?” using passenger data (ie name, age, gender, socio-economic class, etc)*.*
       2. House Price Prediction Using Advanced Regression Techniques: 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 Kaggle’s advanced house price prediction dataset proves that much more influences price negotiations than the number of bedrooms or a white-picket fence. With 79 explanatory variables describing (almost) every aspect of residential homes in Ames, Iowa, this dataset can be used to predict the final price of each home.
       3. Mechanism of Action (MoA) Prediction: Mechanism of action means the biochemical interactions through which a drug generates its pharmacological effect.  If we know a disease affects some particular receptor or downstream set of cell activity, we can develop drugs faster if we can predict how cells and genes affect various receptor sites.  Using a dataset that combines gene expression and cell viability data in addition to the MoA annotations of more than 5,000 drugs.  In this, each drug was tested under two dose (cp\_dose) and three times (cp\_time). So, six samples basically correspond to one drug.  We need to train a model that classifies drugs based on their biological activity.  This problem is a multi-label classification, which means we have multiple targets (not multiple classes).  In this project, perform explanatory data analysis and then train a model using deep neural networks with Keras.
5. **RUBRICS**

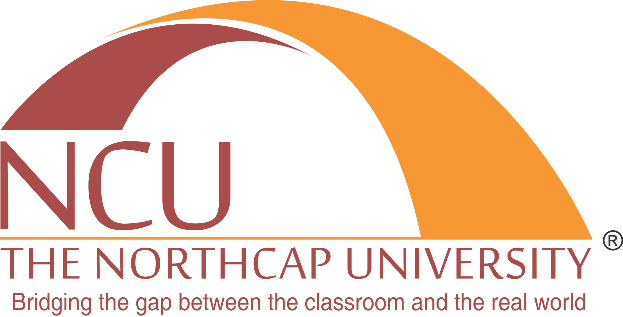
|  |  |
| --- | --- |
| **Marks Distribution** | |
| **Continuous Evaluation (20 Marks)** | **Project Evaluations (30 Marks)** |
| Each experiment shall be evaluated for 10 marks and at the end of the semester proportional marks shall be awarded out of total 20. | Both the projects shall be evaluated for 30 marks each and at the end of the semester viva will be conducted related to the projects as well as concepts learned in labs and this component carries 20 marks. |
| Following is the breakup of 10 marks for each  **4 Marks**: Observation & conduct of experiment. Teacher may ask questions about experiment.  **3 Marks:** For report writing  **3 Marks:** For the 15 minutes quiz to be conducted in every lab. |

**Annexure 1**

**Introduction to AI and ML**

**(CSL 236)**

**Lab Practical Report**



**Faculty name: Nidhi Malik Student name: Sunil Kharsu**

**Roll No.: 20csu371**

**Semester: 5th**

**Group: FSB**

**Department of Computer Science and Engineering**

**NorthCap University, Gurugram- 122001, India**

**Session 2021-22**

**INDEX**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **S.No** | **Experiment** | **Page No.** | **Date of Experiment** | **Date of Submission** | **Marks** | **CO Covered** | **Sign** |
| **1.** | To introduce various libraries of python used for machine learning. |  |  |  |  |  |  |
| **2.** | Write a program to perform data pre-processing techniques for effective machine learning. |  |  |  |  |  |  |
| **3.** | To apply different feature encoding schemes on the given dataset. |  |  |  |  |  |  |
| **4.** | Write a program to apply filter feature selection techniques. |  |  |  |  |  |  |
| **5.** | Reduce dimensionality of Iris dataset using Principal Component Analysis. |  |  |  |  |  |  |
| **6.** | To apply Simple Linear Regression on the given dataset. |  |  |  |  |  |  |
| **7.** | To apply multiple linear regression on any regression dataset. |  |  |  |  |  |  |
| **8.** | To apply Polynomial Linear Regression on the given dataset. |  |  |  |  |  |  |
| **9.** | To solve classification problems using Logistic Regression. |  |  |  |  |  |  |
| **10.** | To solve classification problems using KNN classification. |  |  |  |  |  |  |
| **11.** | To solve classification problems using Naïve Bayes. |  |  |  |  |  |  |
| **13.** | Apply Decision Tree classifier for solving classification problems. |  |  |  |  |  |  |

**EXPERIMENT NO. 1**

|  |
| --- |
| **Student Name and Roll Number: SUNIL KHARSU and 20csu371** |
| **Semester /Section: 5th, FSB** |
| **Link to Code:** |
| **Date: 12/08/21** |
| **Faculty Signature:** |
| **Grade:** |

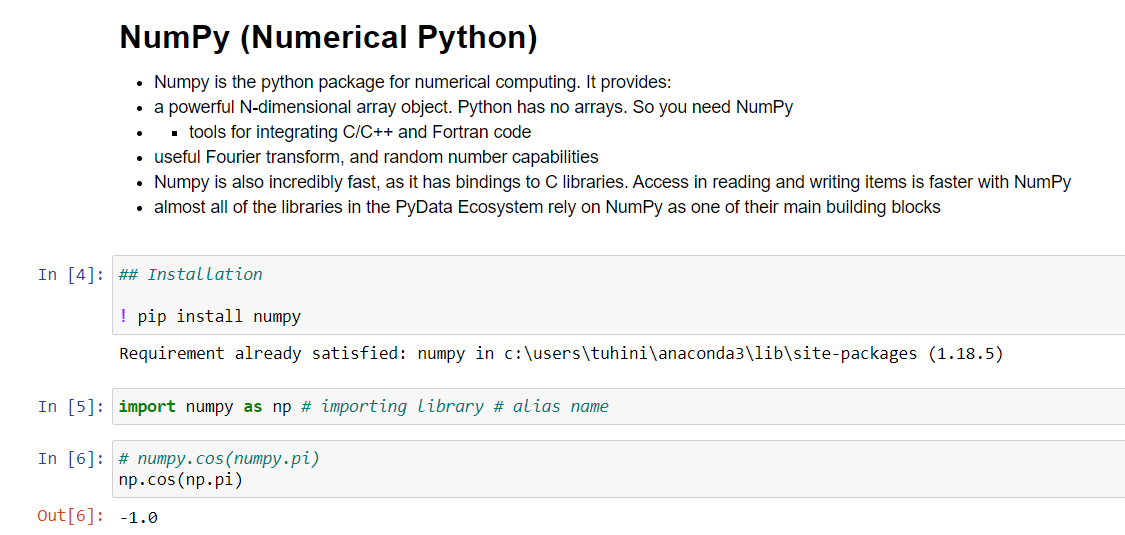
|  |
| --- |
| **Objective(s):**   * To understand the basic libraries of python. * To differentiate between numpys and pandas. |
| **Outcome:**  Students will be familiarized with handling dataset using python basic libraries and applying various operations on dataset using these. |
| **Problem Statement:**  To introduce various libraries of python used for machine learning. |
| **Background Study:**  **Basic libraries** of python are necessary to import datasets and applying various data pre-processing and machine learning techniques on them. |
| **Question Bank:**  **1. How pandas can be used to read data from internet and from your system?**  **Ans 1:-** Pandas also has a comprehensive I/O API that you can use to input data from various sources and output data to various formats. ... Here's a look at how you can use the pandas **read\_html** and read\_clipboard to get tables from websites with just a couple lines of code  **2. How pandas dataframe can be converted to numpy arrays and vice versa?**  **Ans:-** We can do this by using **dataframe. to\_numpy() method** . This will convert the given Pandas Dataframe to Numpy Array.  **3. How to access different rows and columns using loc and iloc?**  **loc is label-based**, which means that you have to specify rows and columns based on their row and column labels. iloc is integer position-based, so you have to specify rows and columns by their integer position values (0-based integer position).  **1. Differentiate between Feature Selection and Dimensionality Reduction.**  Ans:- While both methods are used for reducing the number of features in a dataset, there is an important difference. Feature selection is simply selecting and excluding given features without changing them. **Dimensionality reduction transforms features into a lower dimension**.  **2. What are the advantages of Wrapper methods over filter methods for feature selection?**  Ans:- Filter methods might fail to find the best subset of features in many occasions but wrapper methods can always provide the best subset of features. Using the subset of features from the wrapper **methods make the model more prone to overfitting** as compared to using subset of features from the filter methods.  **3. Explain Regularization methods for Feature Selection.**  **Ans:-** Regularisation consists in **adding a penalty to the different parameters of the machine learning model to reduce the freedom of the model** and in other words to avoid overfitting. ... From the different types of regularisation, Lasso or L1 has the property that is able to shrink some of the coefficients to zero  **4. What are Embedded feature selection methods.**  **Ans:-** In Embedded Methods, the feature selection algorithm is integrated as part of the learning algorithm. Embedded methods **combine the qualities of filter and wrapper methods**. ... Tree algorithms select a feature in each recursive step of the tree growth process and divide the sample set into smaller subsets. |

**Student Work Area**

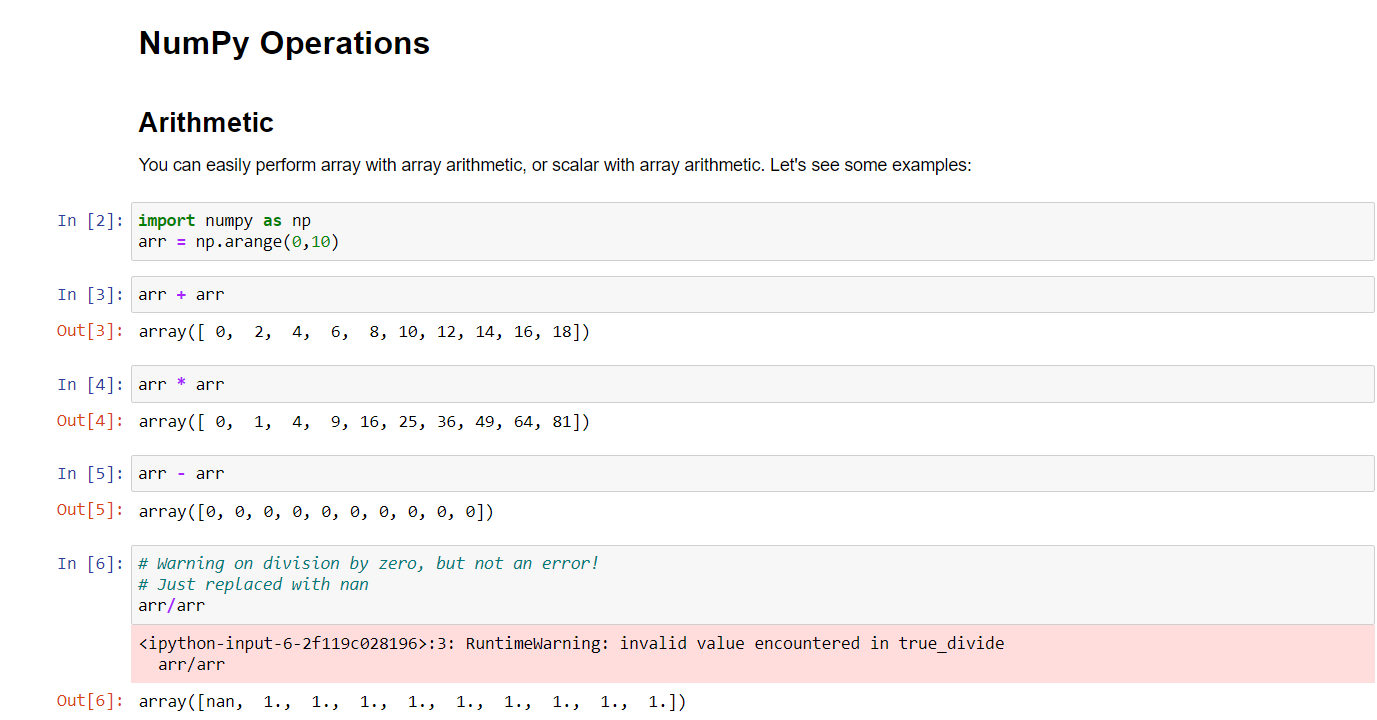
**Algorithm/Flowchart/Code/Sample Outputs**

**Numpy**

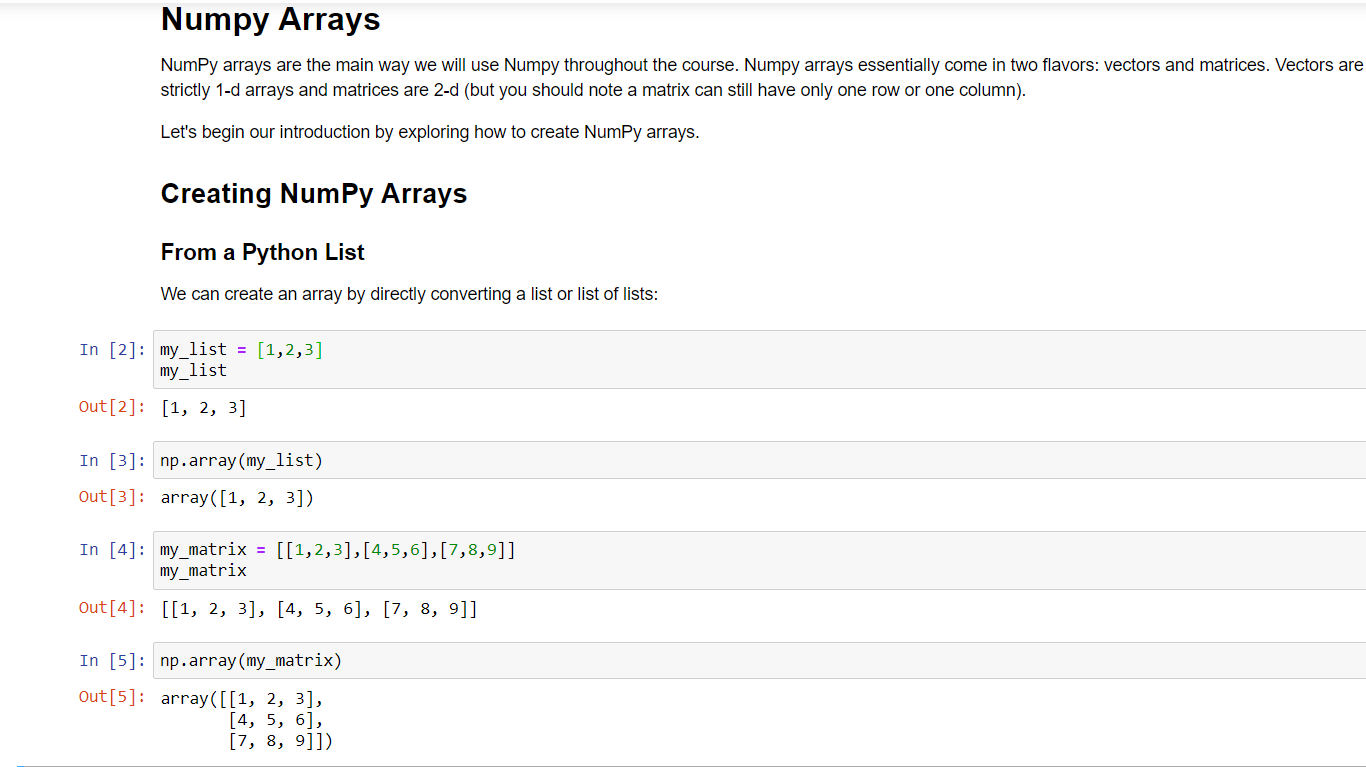
1. **Numpy**

****

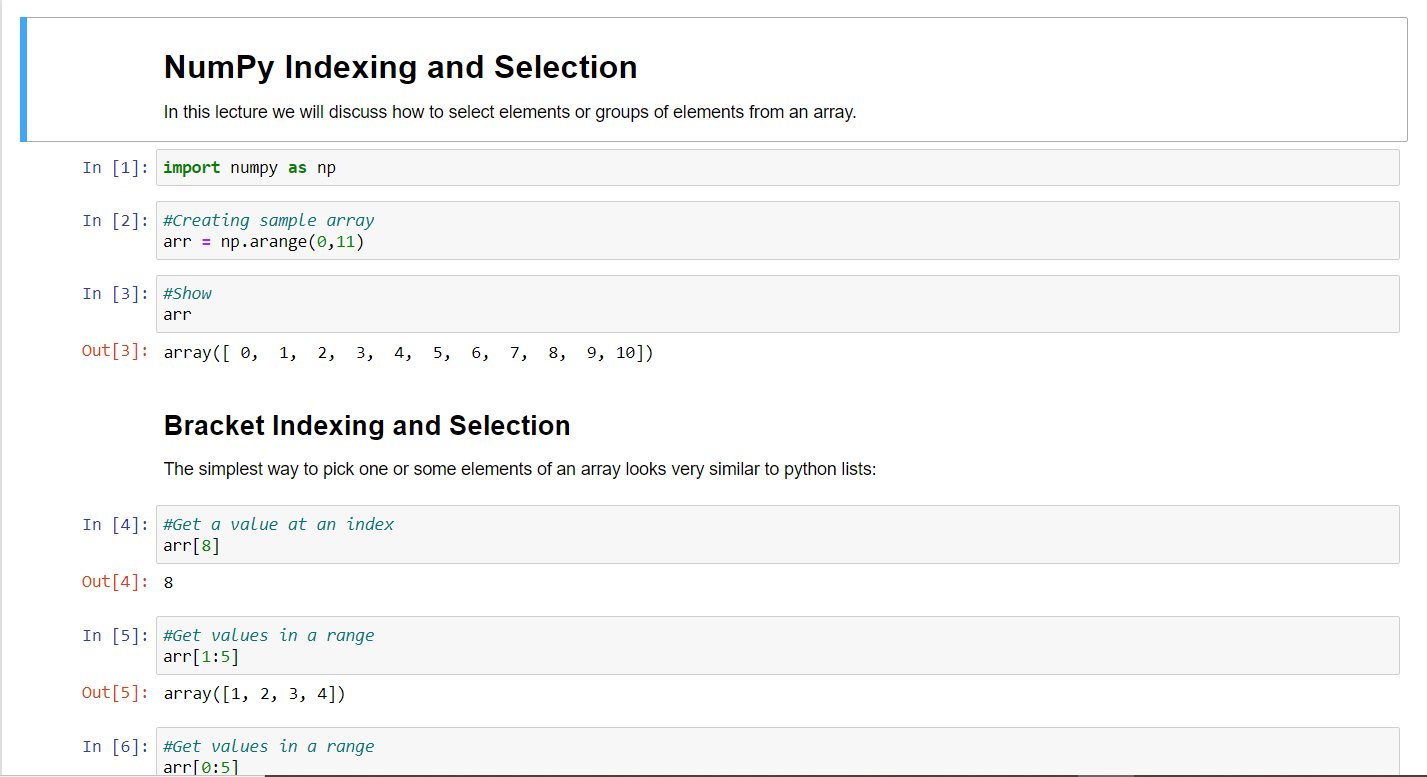
1. **Numpy Operations**

****

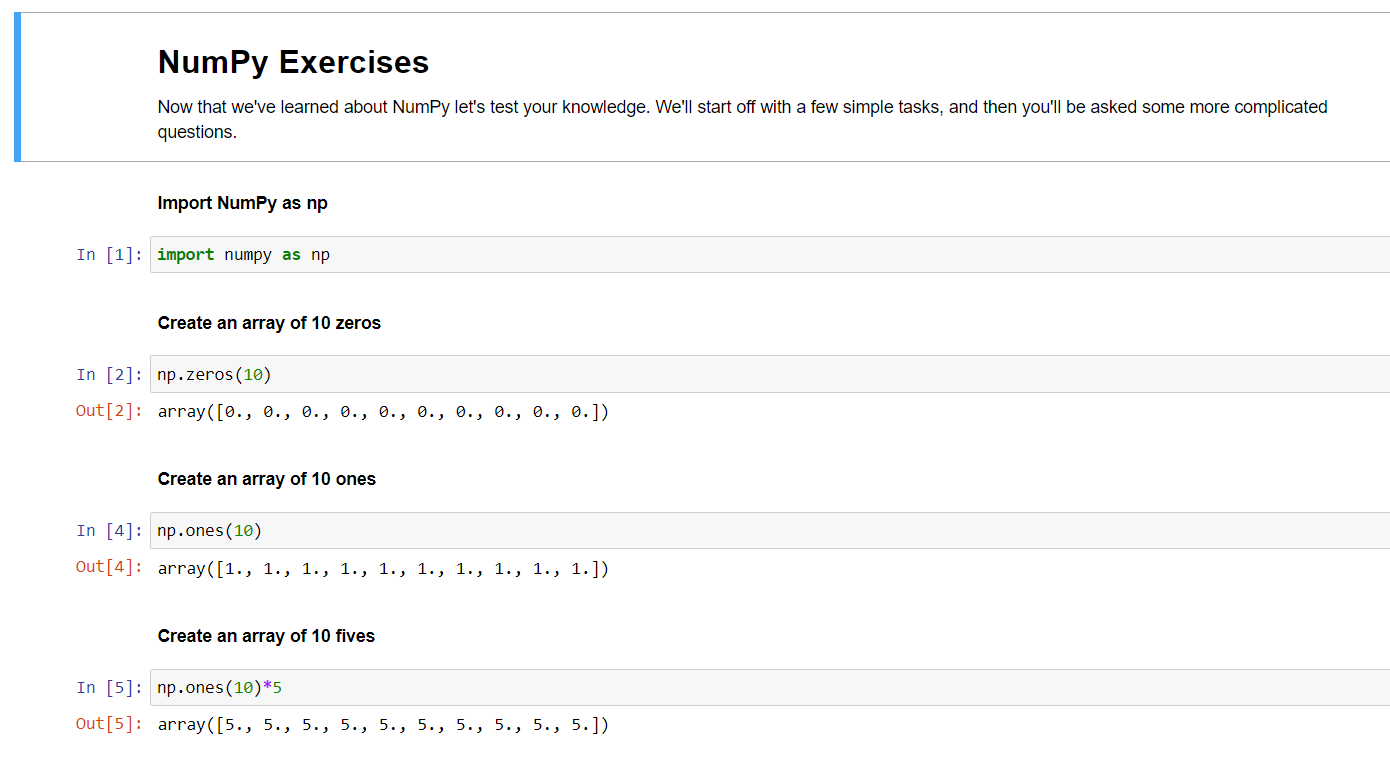
1. **Numpy Arrays**

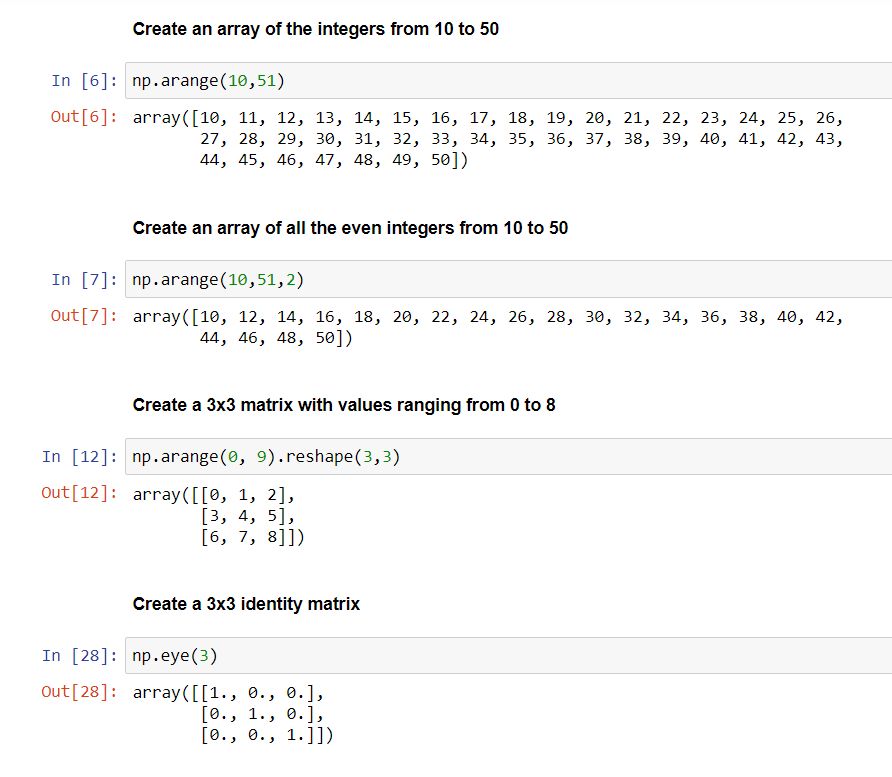
****

1. **Numpy Indexing and Selection**

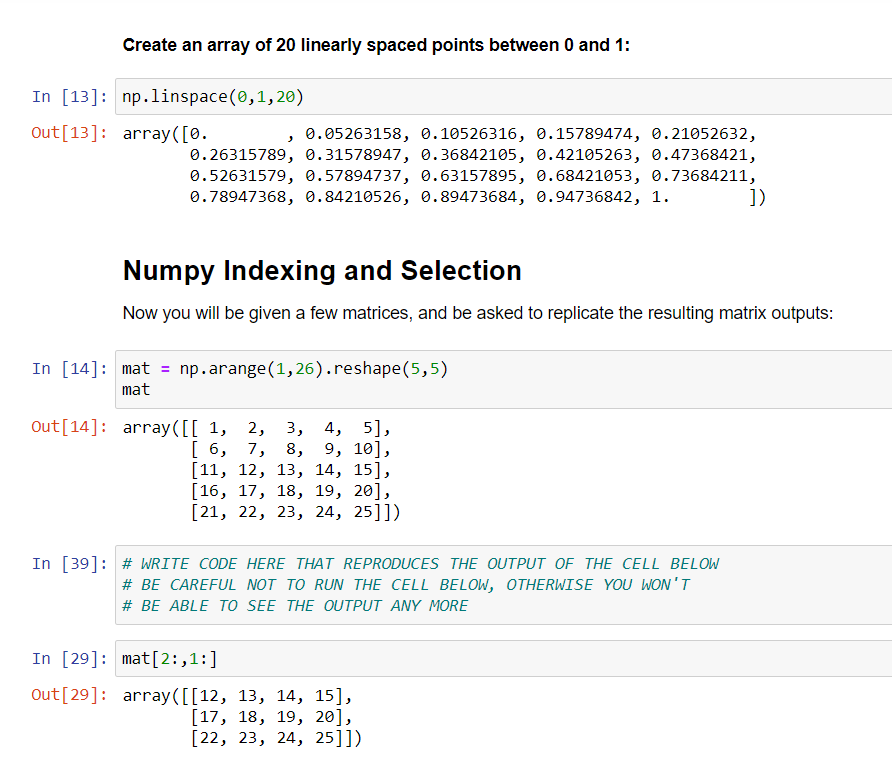
****

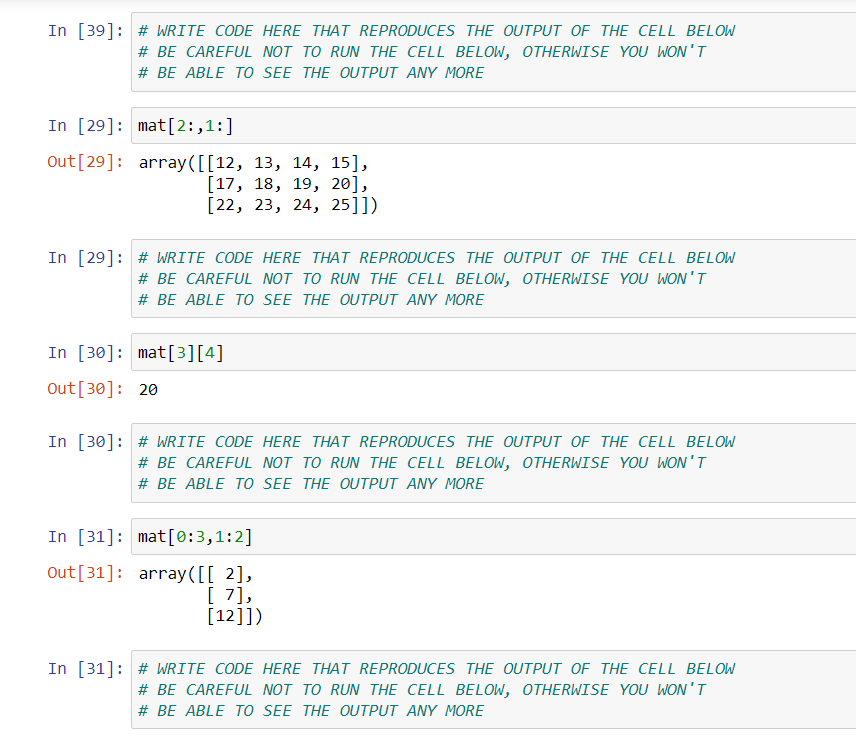
1. **Numpy Drill**

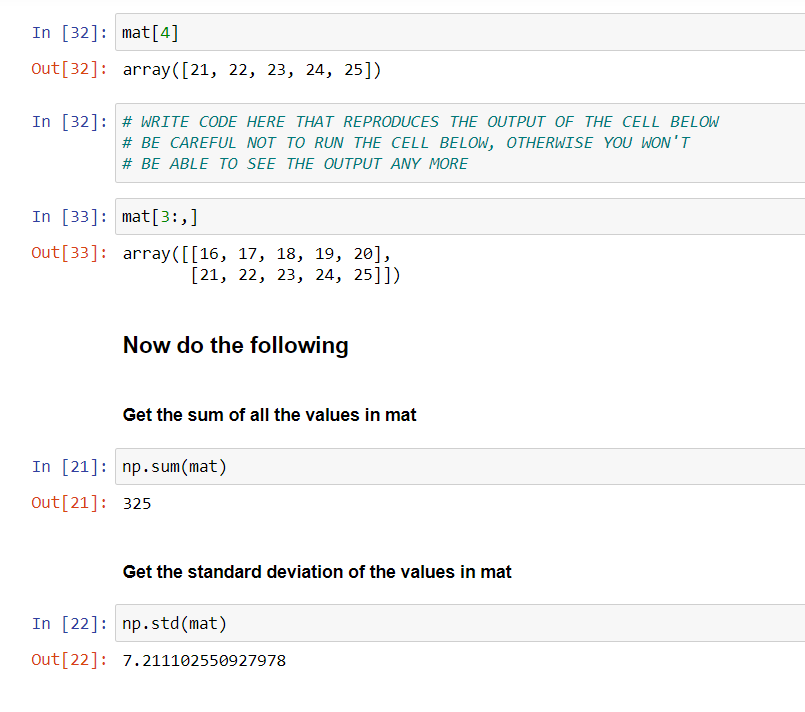
****

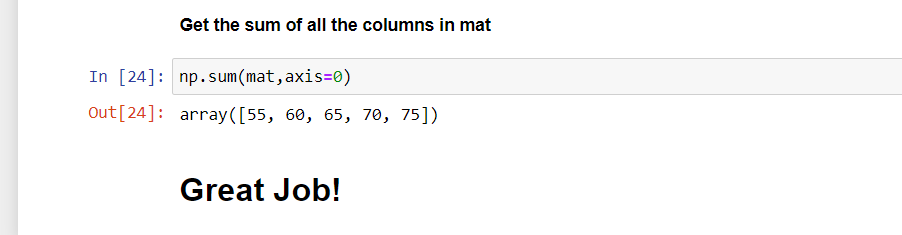
****

****

****

****

****

****

**EXPERIMENT NO. 2**

|  |
| --- |
| **Student Name and Roll Number: SUNIL KHARSU and 20csu371** |
| **Semester /Section: 5th , FSB** |
| **Link to Code:** |
| **Date: 19/08/21** |
| **Faculty Signature:** |
| **Grade:** |

|  |
| --- |
| **Objective(s):**   * To understand the importance of data pre-processing techniques. * To handle missing values, duplicate values, feature scaling etc. |
| **Outcome:**  Students will be familiarized with the understanding and importance of applying various data pre-processing techniques. |
| **Problem Statement:**  Write a program to perform data pre-processing techniques for effective machine learning. |
| **Background Study:** Data preprocessing in Machine Learning is a crucial step thathelps enhance the quality of data to promote the extraction of meaningful insights from the data. Data preprocessing in Machine Learning refers to the technique of preparing (cleaning and organizing) the raw data to make it suitable for a building and training Machine Learning models. |
| **Question Bank:**  1. What are different ways to handle missing values both for numerical as well as categorical data?  **There is various ways to handle missing values ​​of categorical ways.**   1. Ignore observations of missing values ​​if we are dealing with large data sets and less number of records has missing values. 2. Ignore variable, if it is not significant. 3. Develop model to predict missing values. 4. Treat missing data as just another category.   **2. What is the function in python used for finding duplicate rows in data?**  **duplicated()** In Python's Pandas library, Dataframe class provides a member function to find duplicate rows based on all columns or some specific columns i.e. It returns a Boolean Series with True value for each duplicated row. be checked for finding duplicate rows.  **3. Differentiate between two scaling methods used for feature scaling?**  The two most popular techniques for scaling numerical data prior to modeling are **normalization and standardization**. Normalization scales each input variable separately to the range 0-1, which is the range for floating-point values where we have the most precision. |

**Student Work Area**

**Algorithm/Flowchart/Code/Sample Outputs**

**CODE**

import numpy as np

import matplotlib.pyplot as plt

import pandas as pd

data = pd.read\_csv('Data.csv')

x= data.iloc[:,:-3]

y= data.iloc[:,-1]

x= data.iloc[1:5,1:3]

print(data['Salary'].isnull())

dataset = pd.read\_csv('Sample\_real\_estate\_data.csv')

print(dataset['ST\_NUM'].isnull())

print(dataset['ST\_NAME'].isnull())

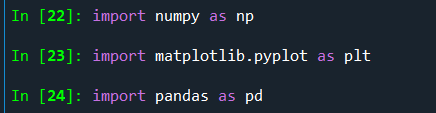
print(dataset['NUM\_BEDROOMS'].isnull())

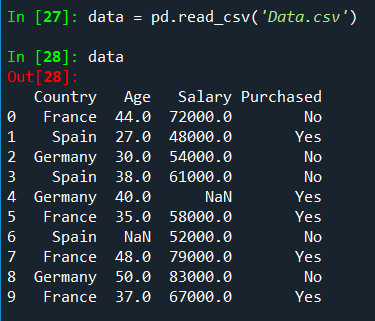
print(dataset['OWN\_OCCUPIED'].isnull())

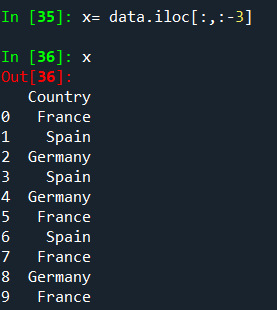
print(dataset['NUM\_BATH'].isnull())

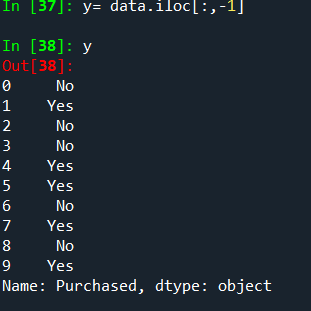
print(dataset['SQ\_FT'].isnull())

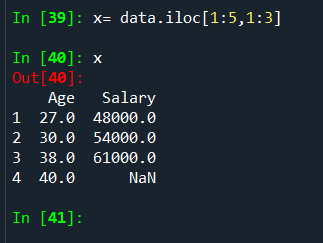
**OUTPUT**

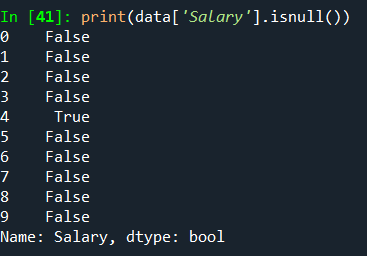


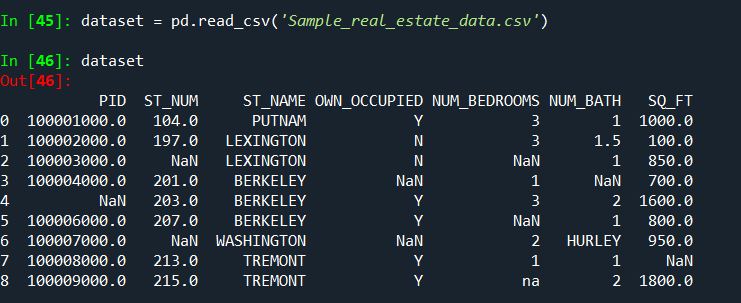




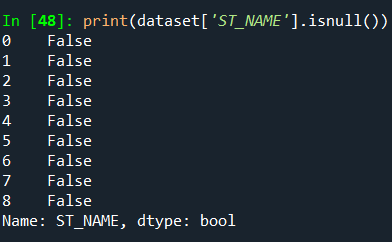


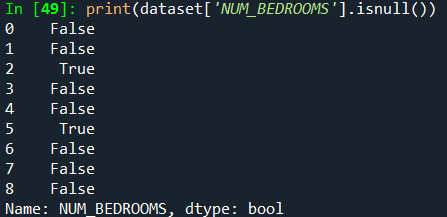


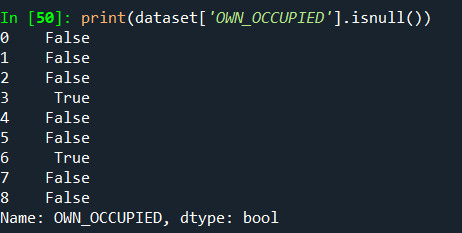


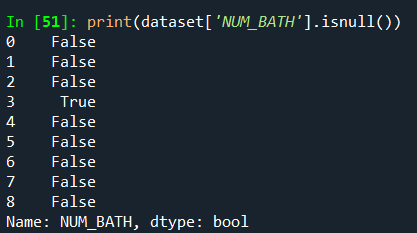


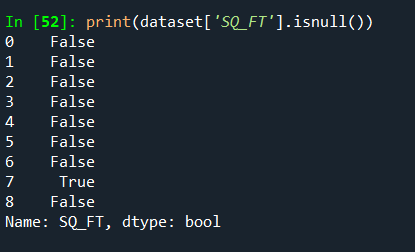












**Imputer**

**CODE :**

import numpy as np

import matplotlib.pyplot as plt

import pandas as pd

#Importing dataset

dataset = pd.read\_csv('Data.csv')

x= dataset.iloc[:,:-1]

y = dataset.iloc[:,:-1]

#Taking care of missing values

from sklearn.impute import SimpleImputer

imputer = SimpleImputer(missing\_values=np.nan, strategy='mean')

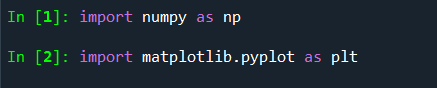
imputer = SimpleImputer(missing\_values=np.nan, strategy='most\_frequent')

imputer = SimpleImputer(missing\_values=np.NaN, strategy='median')

imputer = imputer.fit(x.iloc[:,:-1])

x.iloc[:,1:] = imputer.transform(x.iloc[:,1:])

**Output:**



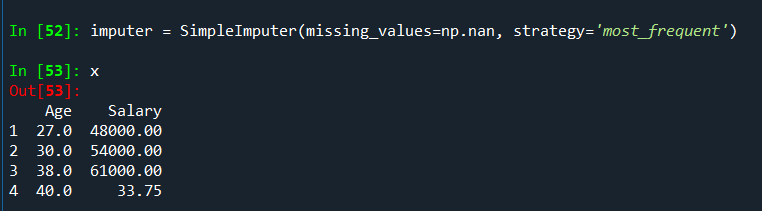




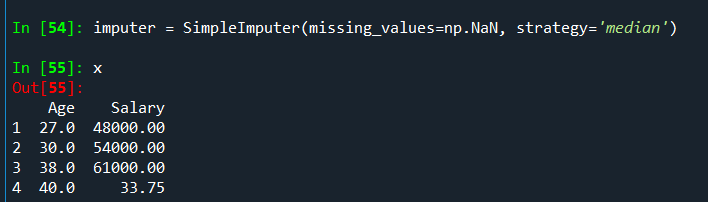
**Importing SimpleImputer from sklearn module and using mean strategy**

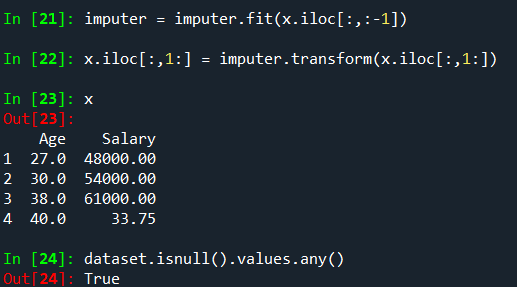


**using most frequent strategy**



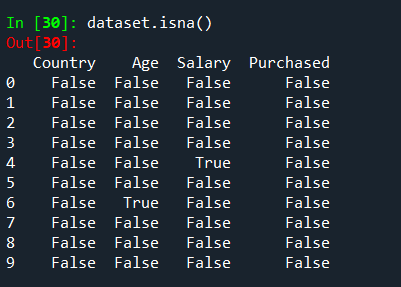
**using median strategy**



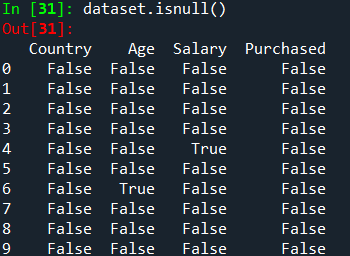




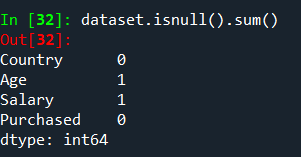
**To find na values**



**To find null values**

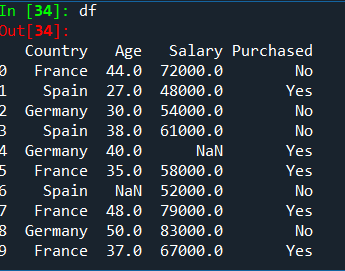


**To find total no. of null values**

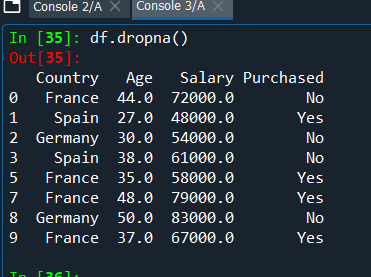


**To copy Dataset**

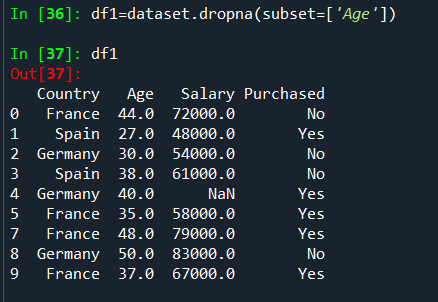




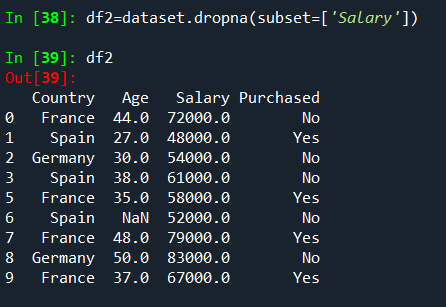
**To drop null values**



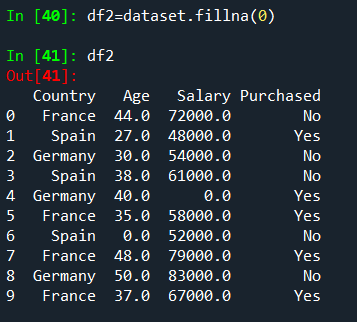
**To drop null values in age column**



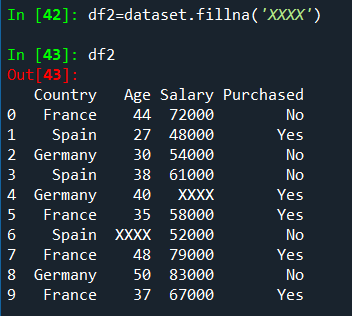
**To drop null values in salary column**



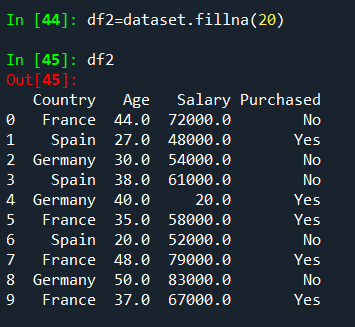
**To fill 0 in place of null values**



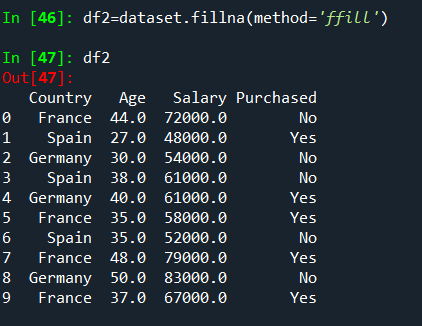
**To fill XXXX in place of null values**



**To fill 20 in place of null values**



**To fill the above values in place of null ones**



**To fill the below values in place of null ones**



**EXPERIMENT NO. 3**

|  |
| --- |
| **Student Name and Roll Number: SUNIL KHARSU and 20csu371** |
| **Semester /Section: 5th, FSB** |
| **Link to Code:** |
| **Date: 26/08/21** |
| **Faculty Signature:** |
| **Grade:** |

|  |
| --- |
| **Objective(s):**   * To perform different encoding schemes. * To prepare dataset by converting categorical data to numeric form for machine learning. * To understand and implement label encoding and one hot encoding. |
| **Outcome:**  Students will be able to understand different encoding schemes to prepare data for machine learning. |
| **Problem Statement:**  To apply different feature encoding schemes on the given dataset. |
| **Background Study:** Machine learning models require all input and output variables to be numeric. This means that if your data contains categorical data, you must encode it to numbers **before you can fit and evaluate a model**. The two most popular techniques are Label Encoding and One-Hot Encoding. |
| **Question Bank:**  1. Can ML algorithms handle categorical data directly?  machine learning algorithms cannot work directly with categorical data and you do need to do some amount of engineering and transformations on this data before you can start modeling on your data.  2. What are the different schemes for encoding categorical data?  One hot encoding and label encoding  3. Differentiate between Label Encoding and One Hot Encoding? |

**Student Work Area**

**Algorithm/Flowchart/Code/Sample Outputs**

**CODE :**

**Car Data Set**

* **Removing duplicates**
* **Missing value treatment**
* **Outlier Treatment**
* **Scaling**
* **Encoding Categorical variables**
* **Reading Data**

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns

%matplotlib inline

df=pd.read\_excel('EDA Cars-1.xlsx')

**head function will tell you the top records in the data set. By default python shows you only top 5 records**

df.head()

**Shape attribute tells us number of observations and variables we have in the data set. It is used to check the dimension of data. The cars data set has 303 observations and 13 variables in the data set.**

df.shape

**info() is used to check the Information about the data and the datatypes of each respective attributes.**

df.info()

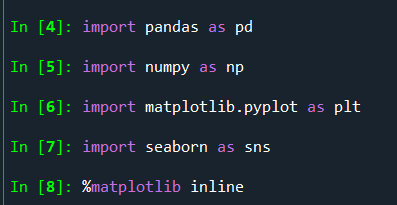
df["POSTAL CODE"]= pd.Categorical(df['POSTAL CODE']) # Converting Postal Code into Category

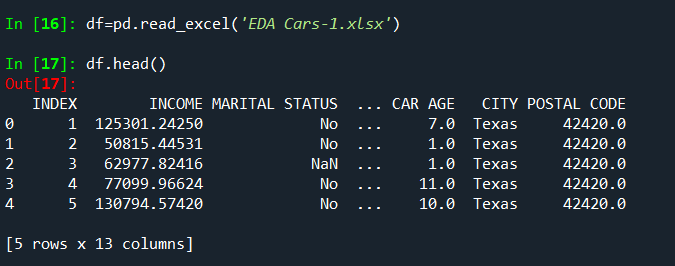
df.info()

df.head()

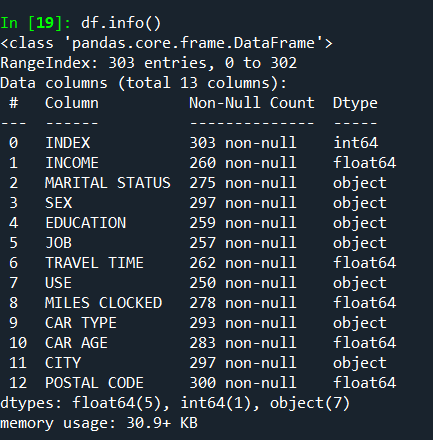
**The describe method will help to see how data has been spread for the numerical values. We can clearly see the minimum value, mean values, different percentile values and maximum values.**

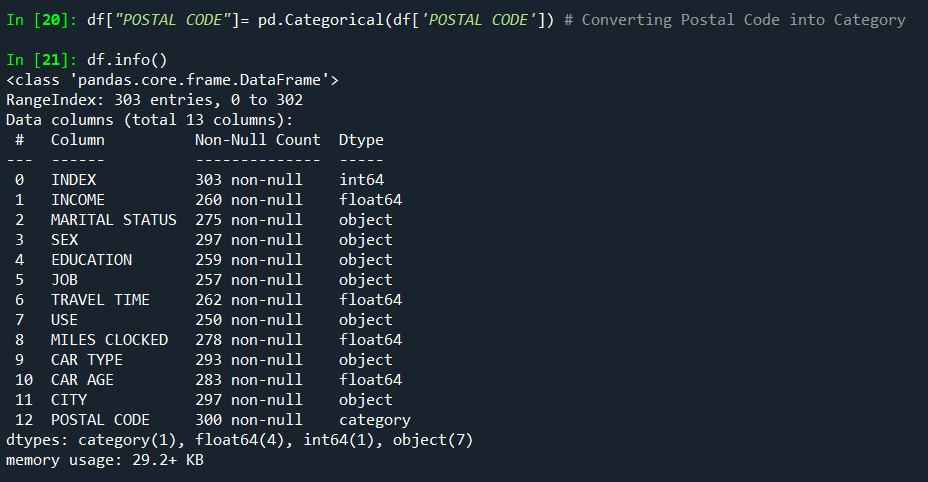
df.describe()

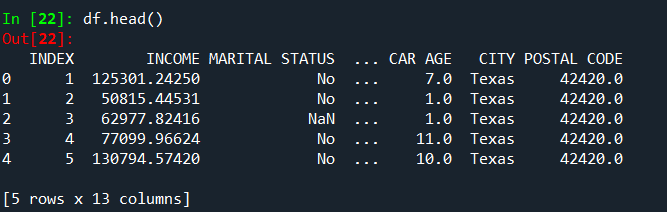


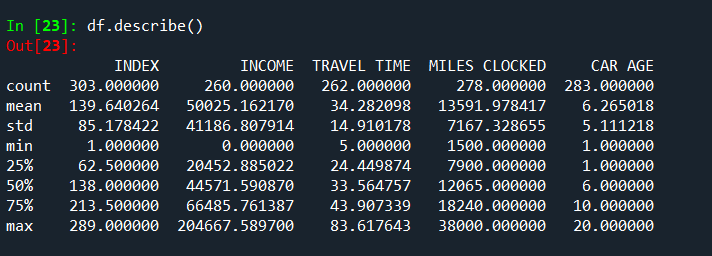












* **Check for duplicate records**

# CHECKING DUPLICATE RECORDS

dups = df.duplicated()

print('Number of duplicate rows = %d' % (dups.sum()))

df[dups]

**Since we have 14 duplicate records in the data, we will remove this from the data set so that we get only distinct records.**

df.drop\_duplicates(inplace=True)

**Post removing the duplicate, we will check whether the duplicates has been removed from the data set or not.**

# Check for duplicate data

dups = df.duplicated()

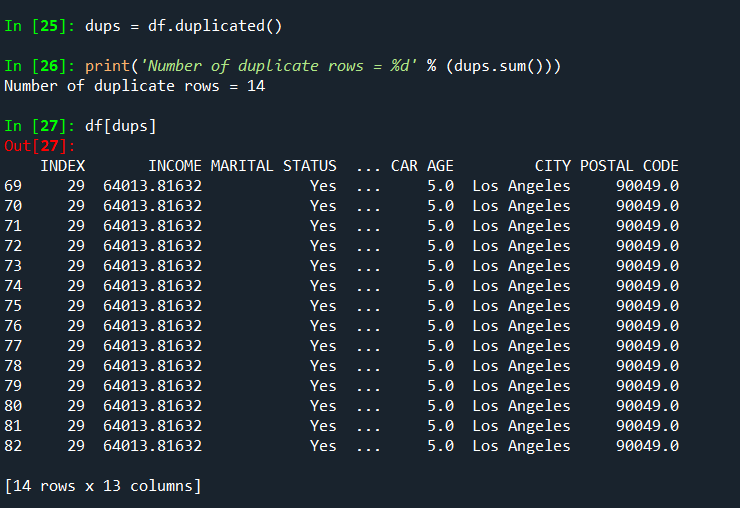
print('Number of duplicate rows = %d' % (dups.sum()))

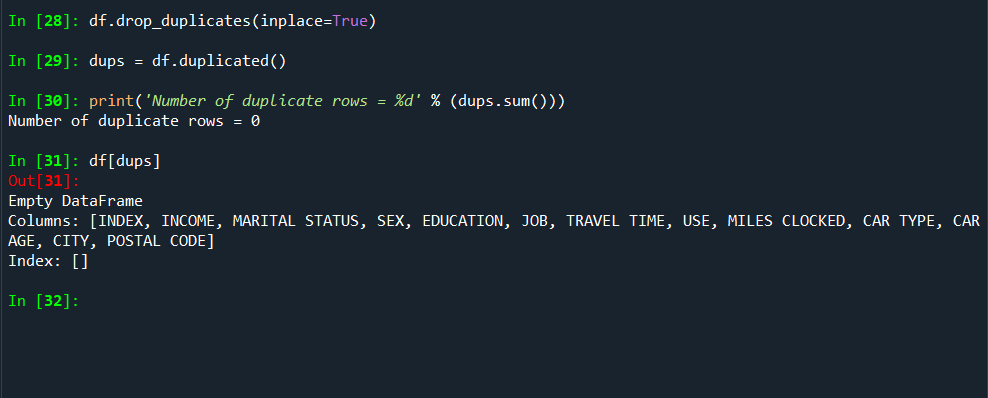
df[dups]

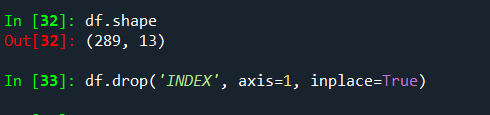
**Now, we can clearly see that there are no duplicate records in the data set. We can also quickly confirm the number of records by using the shape attribute as those 14 records should be removed from the original data. Initially it had 303 records now it should have 289**

df.shape

df.drop('INDEX', axis=1, inplace=True)







* **Outliner Treatment**

# Outlier Treatment

**To check for outliers, we will be plotting the box plots.**

df.boxplot(column=['INCOME', 'TRAVEL TIME'])

plt.show()

df.boxplot(column=['TRAVEL TIME'])

plt.show()

df.boxplot(column=['CAR AGE'])

plt.show()

df.boxplot(column=['MILES CLOCKED'])

plt.show()

**Looking at the box plot, it seems that the three variables INCOME, MILES CLOCKED and TRAVEL TIME have outlier present in the variables.**

**These outliers value needs to be treated and there are several ways of treating them:**

**Drop the outlier value**

**Replace the outlier value using the IQR**

**Created a user defined function for finding the lower and upper range for a variable so that outlier can be treated.**

def remove\_outlier(col):

sorted(col)

Q1,Q3=col.quantile([0.25,0.75])

IQR=Q3-Q1

lower\_range= Q1-(1.5 \* IQR)

upper\_range= Q3+(1.5 \* IQR)

return lower\_range, upper\_range

lrincome,urincome=remove\_outlier(df['INCOME'])

df['INCOME']=np.where(df['INCOME']>urincome,urincome,df['INCOME'])

df['INCOME']=np.where(df['INCOME']<lrincome,lrincome,df['INCOME'])

lrtravel,urtravel=remove\_outlier(df['TRAVEL TIME'])

df['TRAVEL TIME']=np.where(df['TRAVEL TIME']>urtravel,urtravel,df['TRAVEL TIME'])

df['TRAVEL TIME']=np.where(df['TRAVEL TIME']<lrtravel,lrtravel,df['TRAVEL TIME'])

lrmiles,urmiles=remove\_outlier(df['MILES CLOCKED'])

df['MILES CLOCKED']=np.where(df['MILES CLOCKED']>urmiles,urmiles,df['MILES CLOCKED'])

df['MILES CLOCKED']=np.where(df['MILES CLOCKED']<lrmiles,lrmiles,df['MILES CLOCKED'])

df.shape

# Make Boxplots after Outlier Treatment

df.boxplot(column=['INCOME'])

plt.show()

df.boxplot(column=['TRAVEL TIME'])

plt.show()

df.boxplot(column=['MILES CLOCKED'])

plt.show()

**If you look at the box plots above,post treating the outlier there are no outliers in all these columns.**

**Check for missing value**

df.isnull().sum()

**Replacing NULL values in Numerical Columns using Median**

median1=df["INCOME"].median()

median2=df["TRAVEL TIME"].median()

median3=df["MILES CLOCKED"].median()

median4=df["CAR AGE"].median()

df["INCOME"].replace(np.nan,median1,inplace=True)

df["TRAVEL TIME"].replace(np.nan,median2,inplace=True)

df["MILES CLOCKED"].replace(np.nan,median3,inplace=True)

df["CAR AGE"].replace(np.nan,median4,inplace=True)

**Replacing NULL values in Categorical Columns using Mode**

mode1=df["SEX"].mode().values[0]

mode2=df["MARITAL STATUS"].mode().values[0]

mode3=df["EDUCATION"].mode().values[0]

mode4=df["JOB"].mode().values[0]

mode5=df["USE"].mode().values[0]

mode6=df['CITY'].mode().values[0]

mode7=df["CAR TYPE"].mode().values[0]

mode8=df["POSTAL CODE"].mode().values[0]

df["SEX"]=df["SEX"].replace(np.nan,mode1)

df["MARITAL STATUS"]= df["MARITAL STATUS"].replace(np.nan,mode2)

df["EDUCATION"]=df["EDUCATION"].replace(np.nan,mode3)

df["JOB"]=df["JOB"].replace(np.nan,mode4)

df["USE"]=df["USE"].replace(np.nan,mode5)

df["CAR TYPE"]=df["CAR TYPE"].replace(np.nan,mode7)

df['CITY']=df['CITY'].replace(np.nan,mode6)

df['POSTAL CODE']=df['POSTAL CODE'].replace(np.nan,mode8)

pip install -U seaborn

sns.histplot(df.INCOME,bins=10);

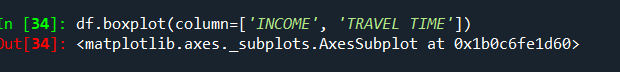
sns.countplot(x=df["EDUCATION"]);

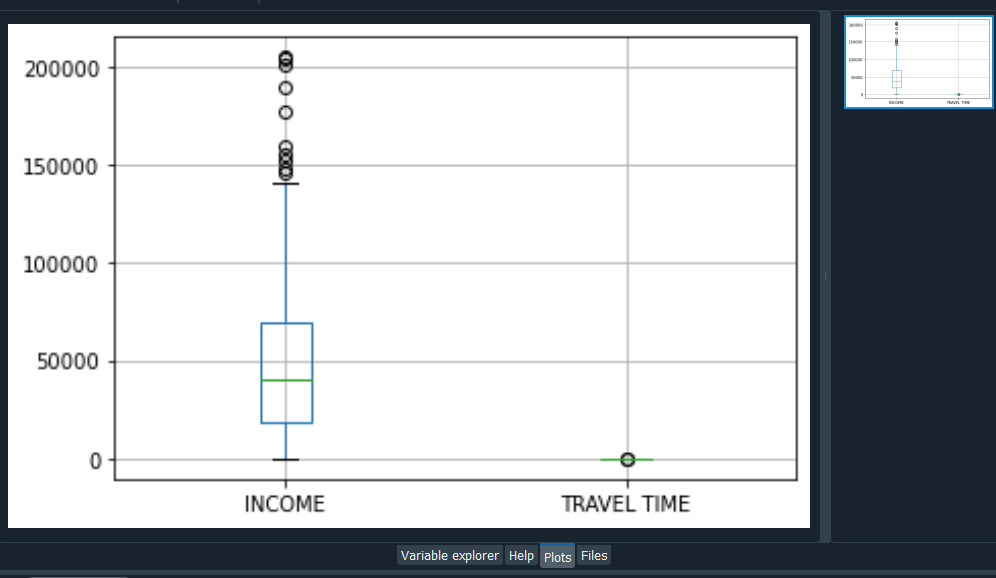
df.corr()

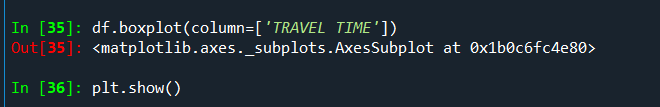
**Correlation Heatmap**

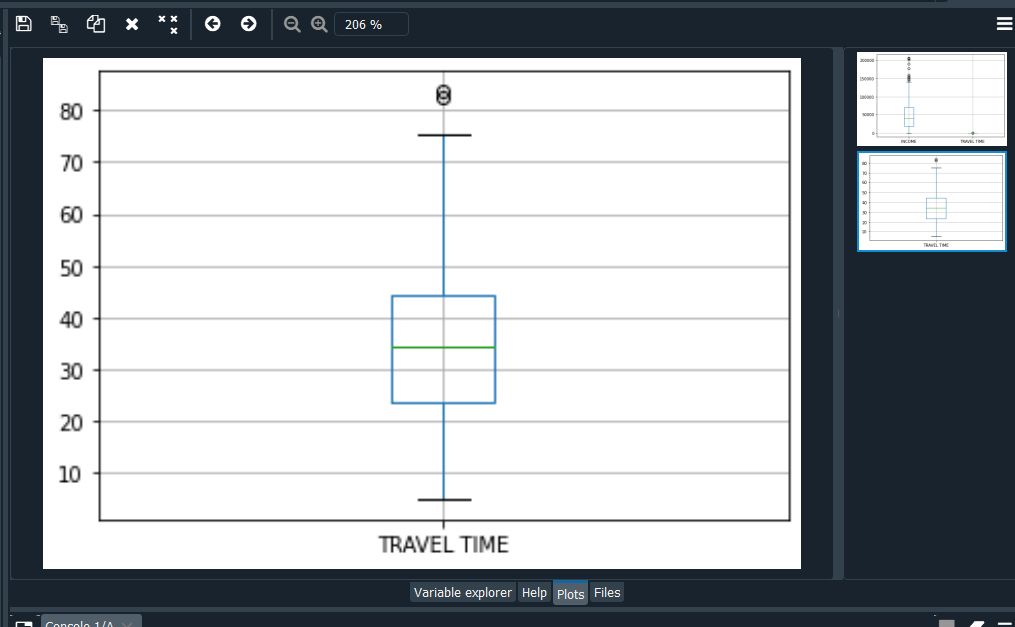
sns.heatmap(df.corr(), annot=True, fmt='.2f')

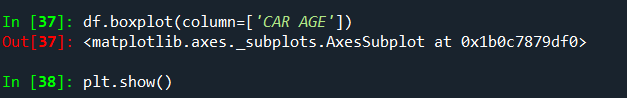
plt.show()

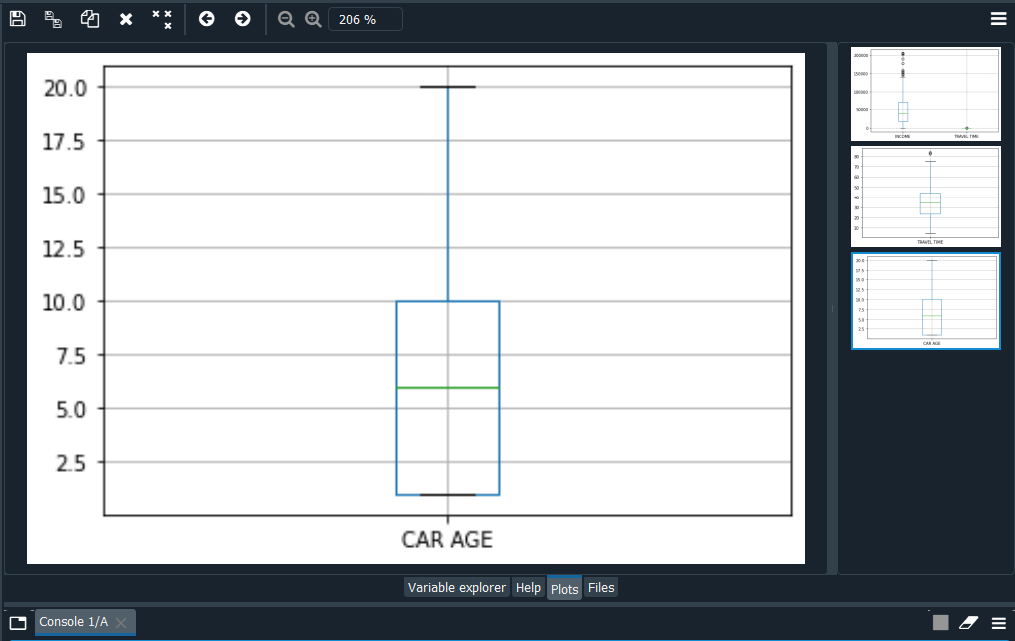
****

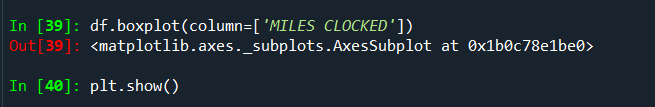
****

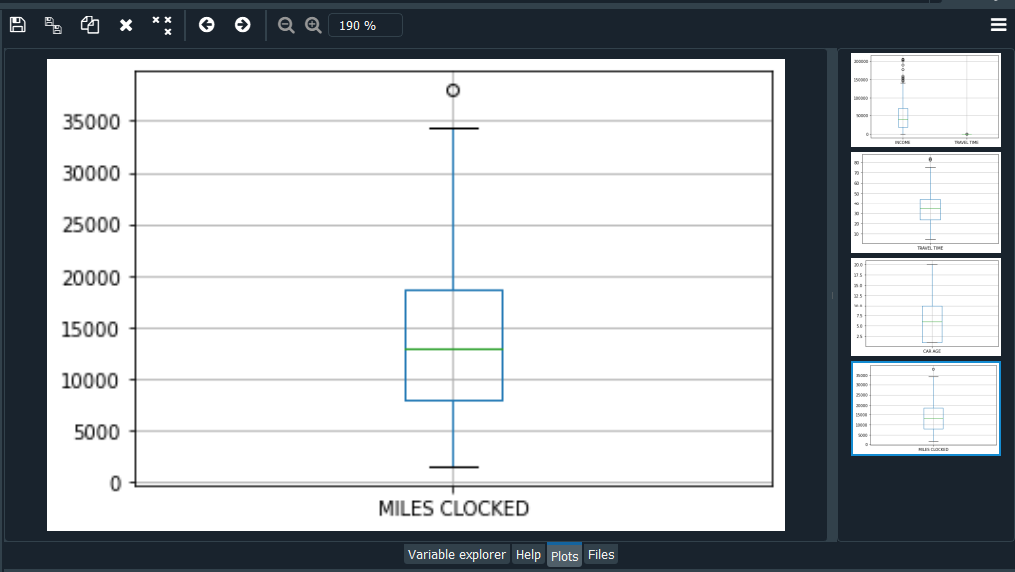
****

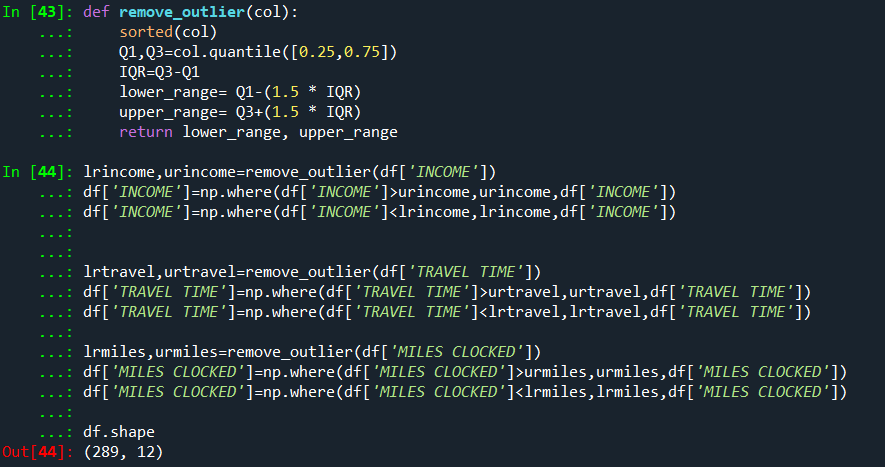
****

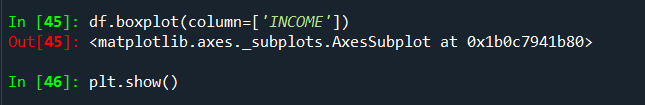
****

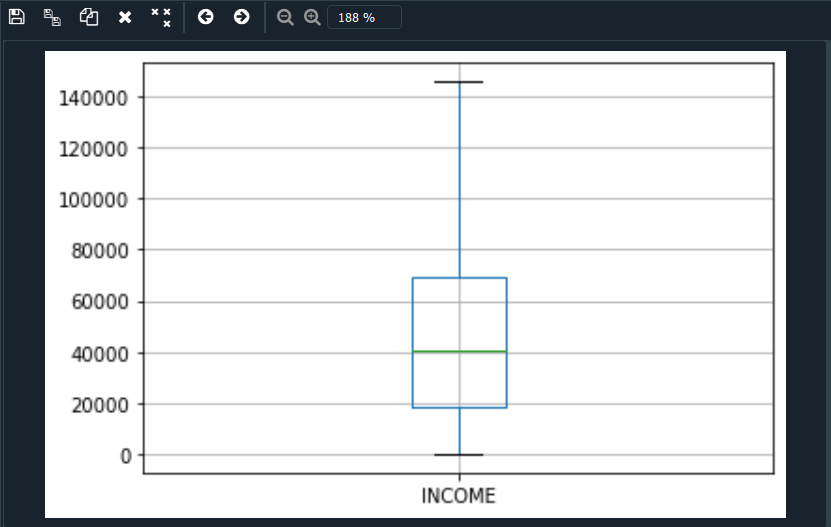
****

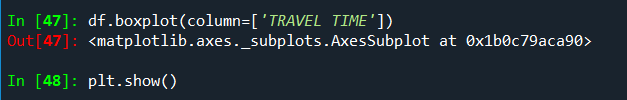
****

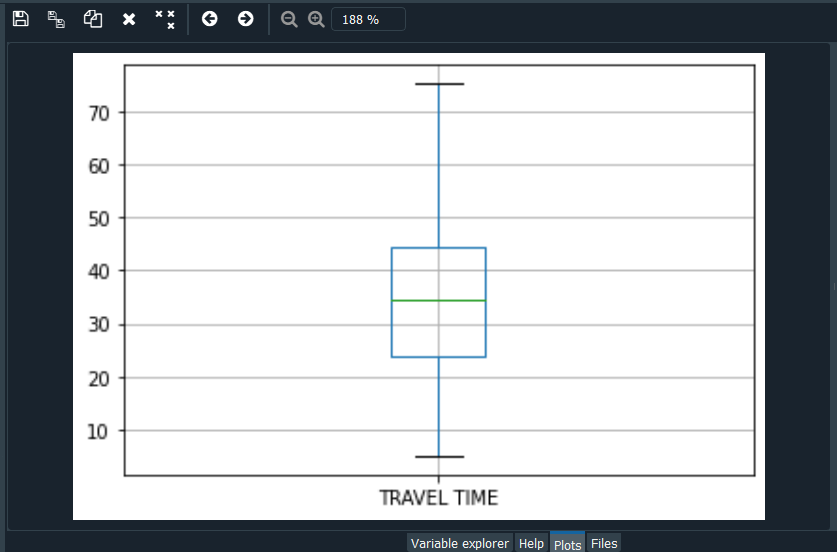
****

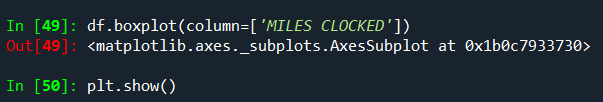
****

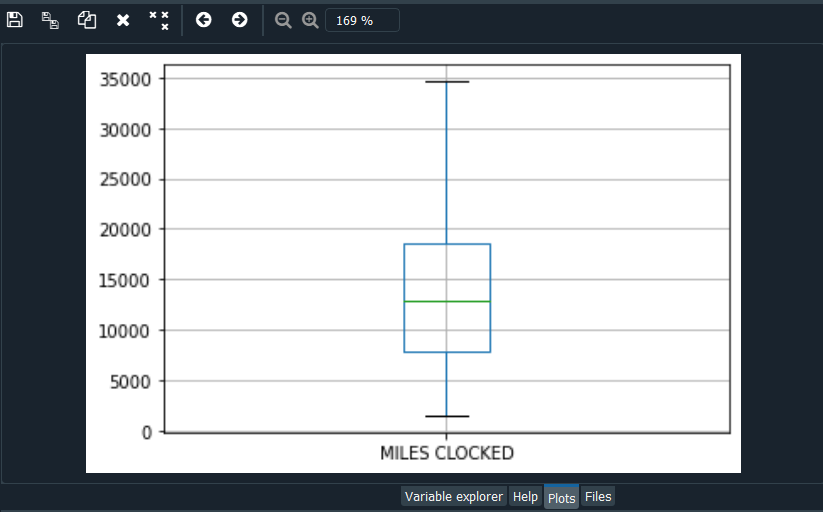
****

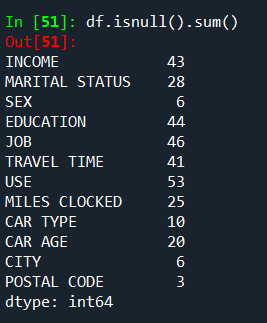
****

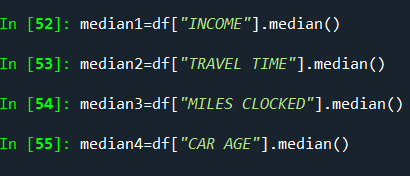
****

****

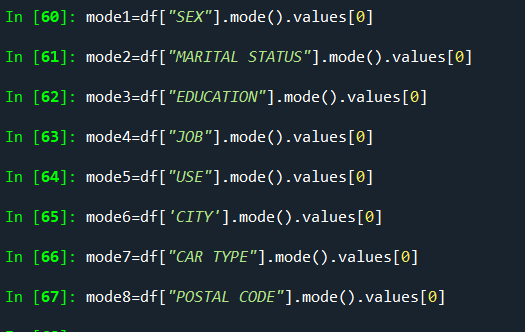
****

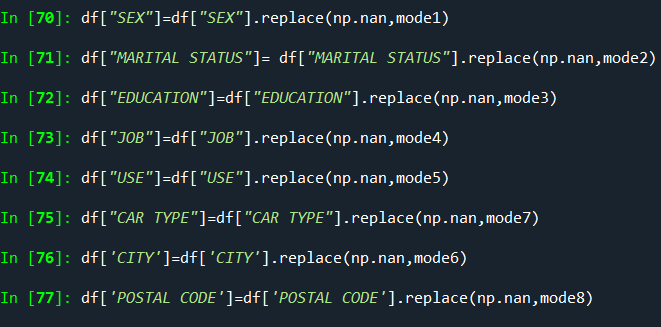
****

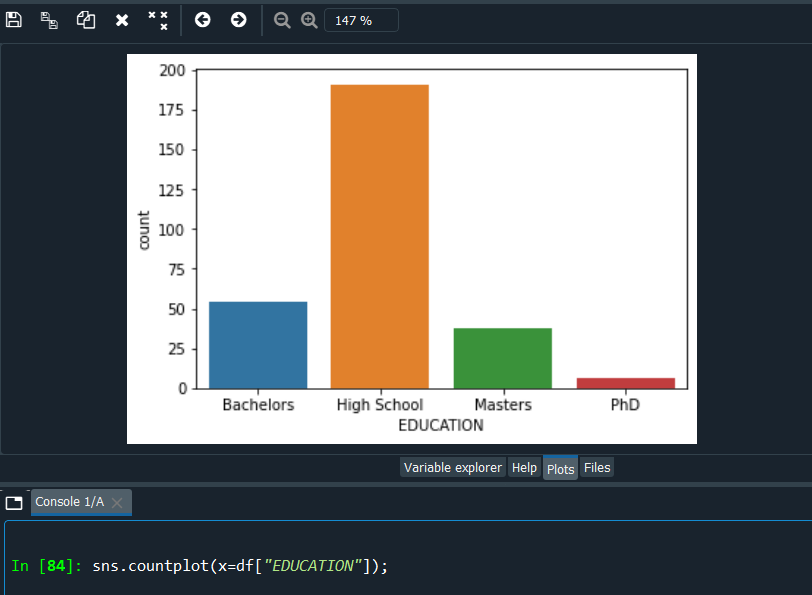
****

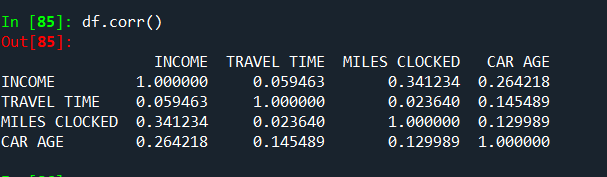
****

****

****

****

****

****

****

* **Scaling**

**Scales the data. Essentially returns the z-scores of every attribute**

from sklearn.preprocessing import StandardScaler

std\_scale = StandardScaler()

std\_scale

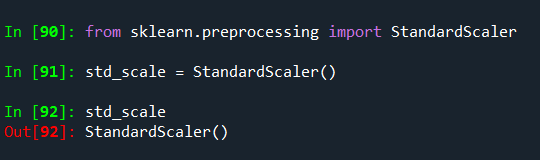
df['INCOME'] = std\_scale.fit\_transform(df[['INCOME']])

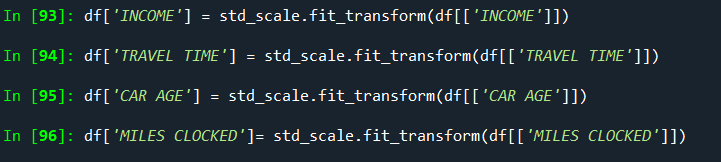
df['TRAVEL TIME'] = std\_scale.fit\_transform(df[['TRAVEL TIME']])

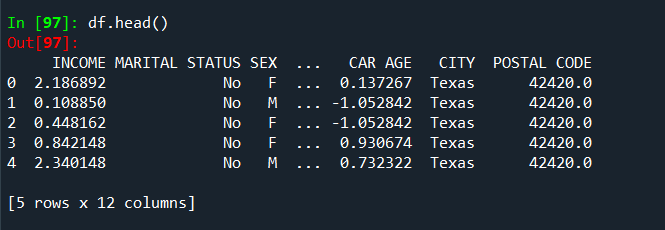
df['CAR AGE'] = std\_scale.fit\_transform(df[['CAR AGE']])

df['MILES CLOCKED']= std\_scale.fit\_transform(df[['MILES CLOCKED']])

df.head()

****





* **Encoding**

# ENCODING

**One-Hot-Encoding is used to create dummy variables to replace the categories in a categorical variable into features of each category and represent it using 1 or 0 based on the presence or absence of the categorical value in the record.**

**This is required to do since the machine learning algorithms only works on the numerical data. That is why there is a need to convert the categorical column into numerical one.**

**get\_dummies is the method which creates dummy variable for each categorical variable.**

dummies=pd.get\_dummies(df[["MARITAL STATUS", "SEX","EDUCATION","JOB","USE","CAR TYPE","CITY"]], columns=["MARITAL STATUS", "SEX","EDUCATION","JOB","USE","CAR TYPE","CITY"], prefix=["married", "sex","Education","Job","Use","cartype","city"],drop\_first=True).head()

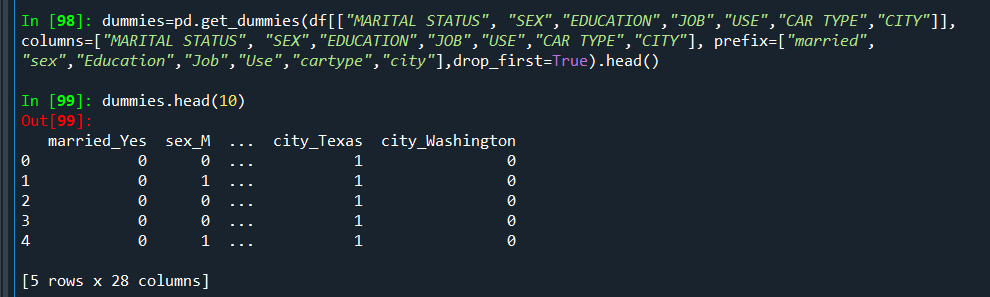
dummies.head(10)

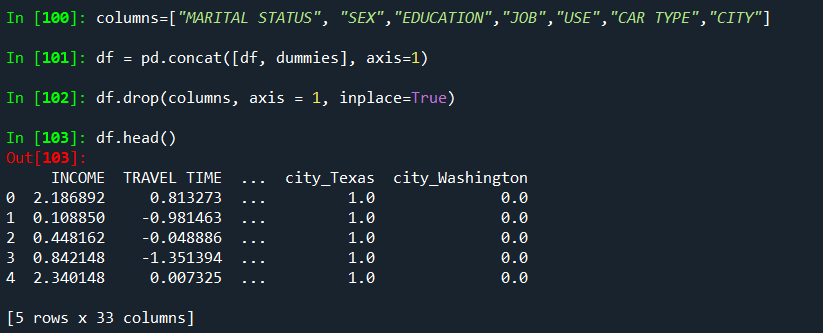
columns=["MARITAL STATUS", "SEX","EDUCATION","JOB","USE","CAR TYPE","CITY"]

df = pd.concat([df, dummies], axis=1)

df.drop(columns, axis = 1, inplace=True)

df.head()

****

****

**EXPERIMENT NO. 4**

|  |
| --- |
| **Student Name and Roll Number: SUNIL KHARSU and 20csu371** |
| **Semester /Section: 5th, FSB** |
| **Link to Code:** |
| **Date: 02/09/21** |
| **Faculty Signature:** |
| **Grade:** |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| |  | | --- | | **Objective(s):**   * To understand the importance of feature selection * To differentiate between different types of feature selection. * Build a model using feature selection techniques. | | **Outcome:**  Students will be familiarized with model building using feature selection techniques and  optimization. | | **Problem Statement:**  Write a program to apply filter feature selection techniques. | | **Background Study:** 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. | | **Question Bank:**  **1. What are different filter feature selection techniques?**   * **Pearson’s Correlation:** It is used as a measure for quantifying linear dependence between two continuous variables X and Y. Its value varies from -1 to +1. Pearson’s correlation is given as:   fs2   * **LDA:** Linear discriminant analysis is used to find a linear combination of features that characterizes or separates two or more classes (or levels) of a categorical variable. * **ANOVA:** ANOVA stands for Analysis of variance. It is similar to LDA except for the fact that it is operated using one or more categorical independent features and one continuous dependent feature. It provides a statistical test of whether the means of several groups are equal or not. * **Chi-Square:** It is a is a statistical test applied to the groups of categorical features to evaluate the likelihood of correlation or association between them using their frequency distribution.   **2. How feature selection techniques depend on the data type of input features and output variable?**  **Recursive Feature Elimination (RFE)** takes as input the instance of a Machine Learning model and the final desired number of features to use. It then recursively reduces the number of features to use by ranking them using the Machine Learning model accuracy as metrics.  **3. What is the mathematics behind Pearson’s Correlation to rank features?**  Pearson's correlation coefficient is the test statistics that **measures the statistical relationship, or association** , between two continuous variables. ... It gives information about the magnitude of the association, or correlation, as well as the direction of the relationship.  The assumptions for Pearson correlation coefficient are as follows: **level of measurement, related pairs, absence of outliers, normality of variables, linearity, and homoscedasticity** . Level of measurement refers to each variable. For a Pearson correlation, each variable should be continuous  The Pearson correlation coefficient is used to **measure the strength of a linear association between two variables** , where the value r = 1 means a perfect positive correlation and the value r = -1 means a perfect negative correlation. | |
|  |
|  |

**Student Work Area**

**Algorithm/Flowchart/Code/Sample Outputs**

**# Pearson’s Correlation with f\_regression function**

**# Creating regression dataset with make\_regression function**

**Feature Scaling**

**CODE :**

# Data Preprocessing Template

# Importing the libraries

import numpy as np

import matplotlib.pyplot as plt

import pandas as pd

# Importing the dataset

dataset = pd.read\_csv('Data.csv')

X = dataset.iloc[:, :-1].values

y = dataset.iloc[:, 3].values

# Taking care of missing data scilearn (sklearn) library for scientific computation

from sklearn.impute import SimpleImputer

imputer = SimpleImputer(missing\_values = np.nan, strategy = 'mean')

imputer = imputer.fit(X[:, 1:3])

X[:, 1:3] = imputer.transform(X[:, 1:3])

# Encoding categorical data

# Encoding the Independent Variable

from sklearn.preprocessing import LabelEncoder, OneHotEncoder

from sklearn.compose import ColumnTransformer

labelencoder\_X = LabelEncoder()

X[:, 0] = labelencoder\_X.fit\_transform(X[:, 0])

ct = ColumnTransformer([("Country", OneHotEncoder(), [0])], remainder = 'passthrough')

X = ct.fit\_transform(X)

#below two lines deprecated

#onehotencoder = OneHotEncoder( ColumnTransformer = [0])

#X = onehotencoder.fit\_transform(X).toarray()

# Encoding the Dependent Variable

labelencoder\_y = LabelEncoder()

y = labelencoder\_y.fit\_transform(y)

# Splitting the dataset into the Training set and Test set

# use model\_selection instead of cross\_validation

from sklearn.model\_selection import train\_test\_split

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size = 0.2, random\_state = 0)

# Feature Scaling - Normalisation

from sklearn.preprocessing import MinMaxScaler

sc\_X = MinMaxScaler()

X\_train = sc\_X.fit\_transform(X\_train)

X\_test = sc\_X.transform(X\_test)

"""sc\_y = MinMaxScaler()

y\_train = sc\_y.fit\_transform(y\_train)"""

# Feature Scaling - Standardisation

from sklearn.preprocessing import StandardScaler

sc\_X = StandardScaler()

X\_train = sc\_X.fit\_transform(X\_train)

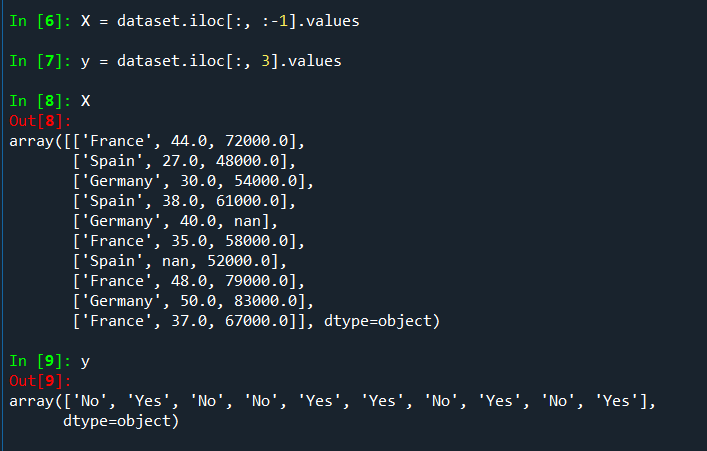
X\_test = sc\_X.transform(X\_test)

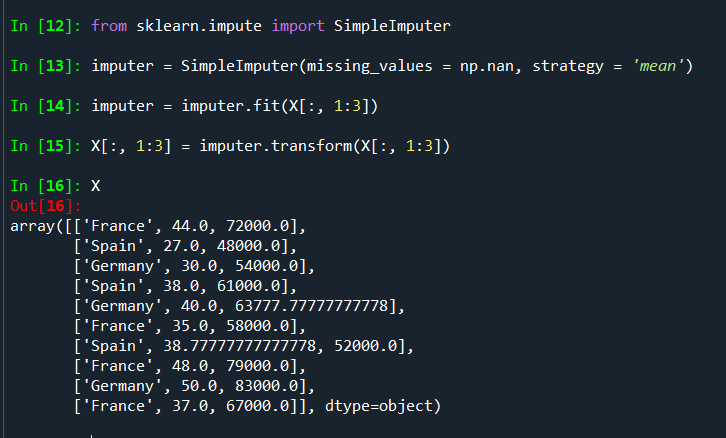
"""sc\_y = StandardScaler()

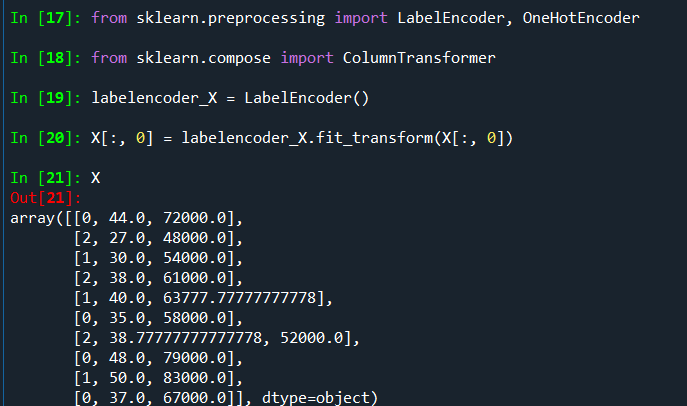
y\_train = sc\_y.fit\_transform(y\_train)"""

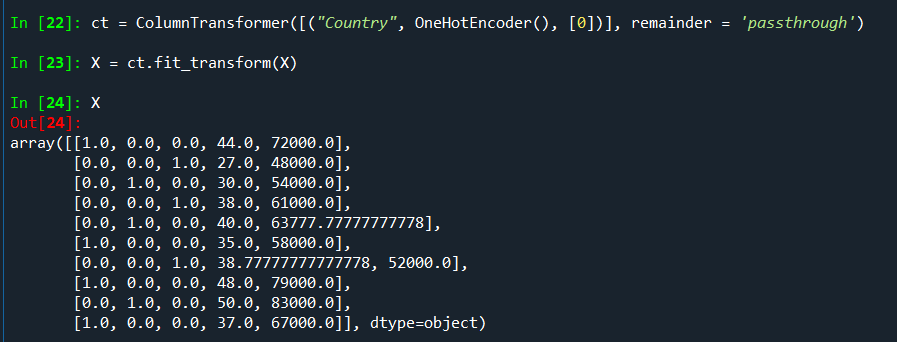
**OUTPUT :**

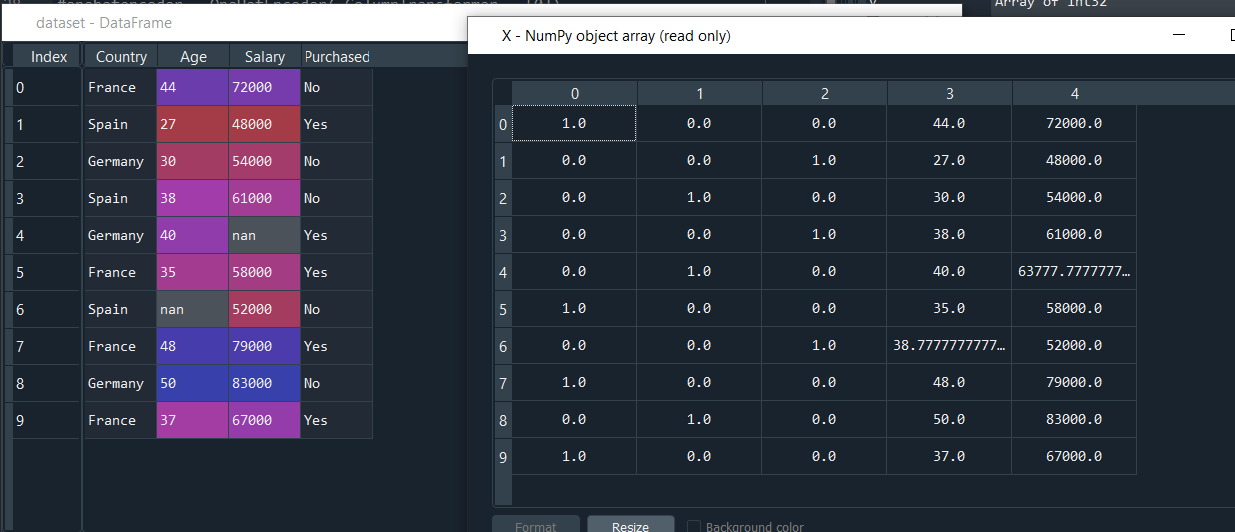


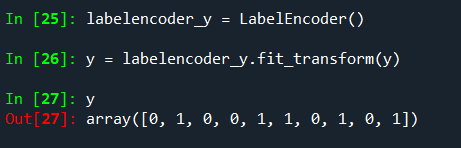


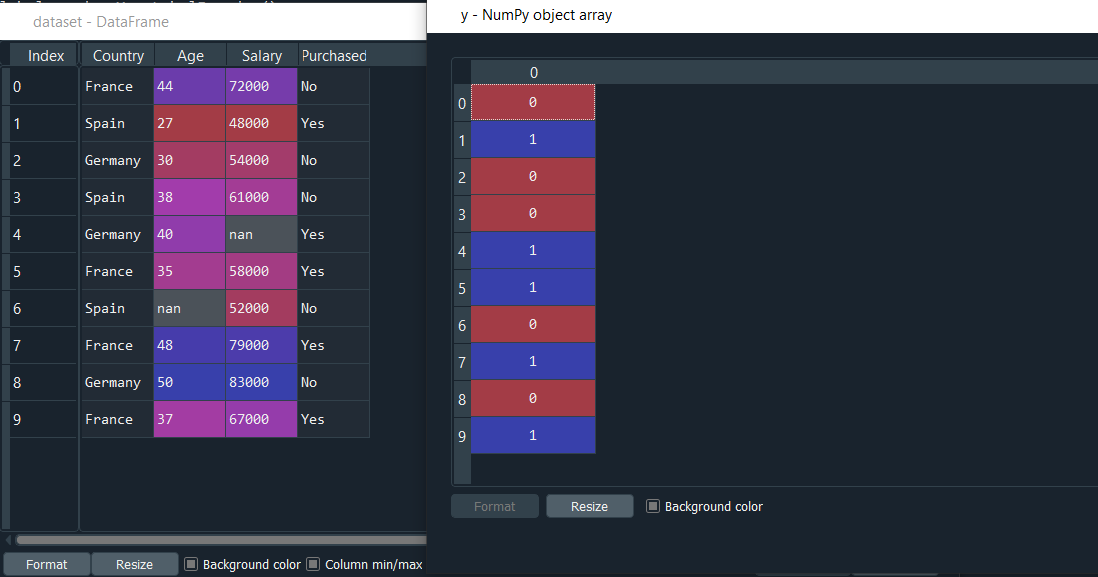


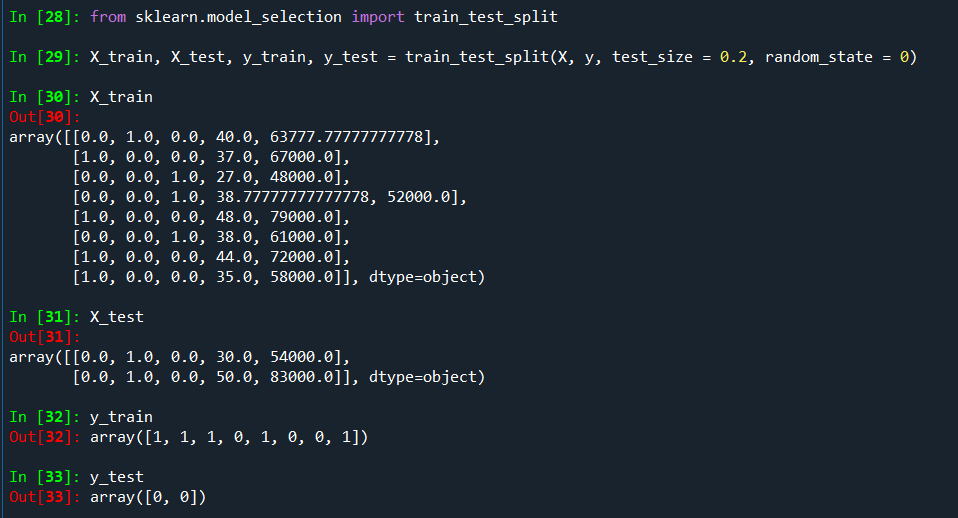


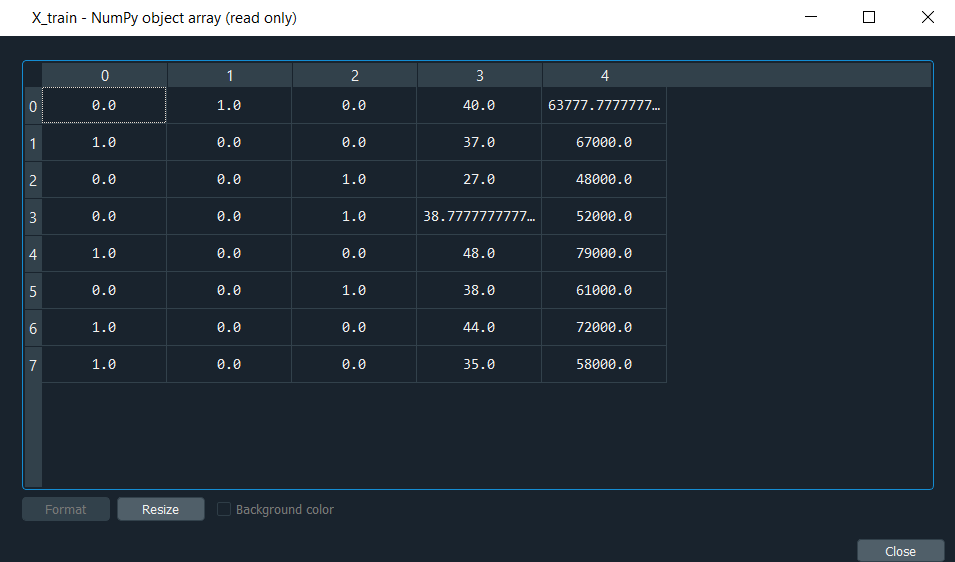




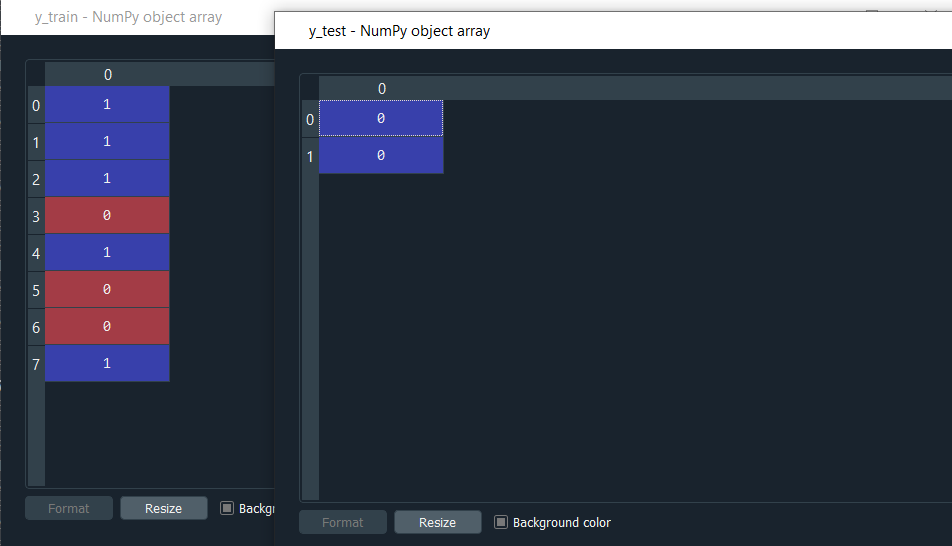


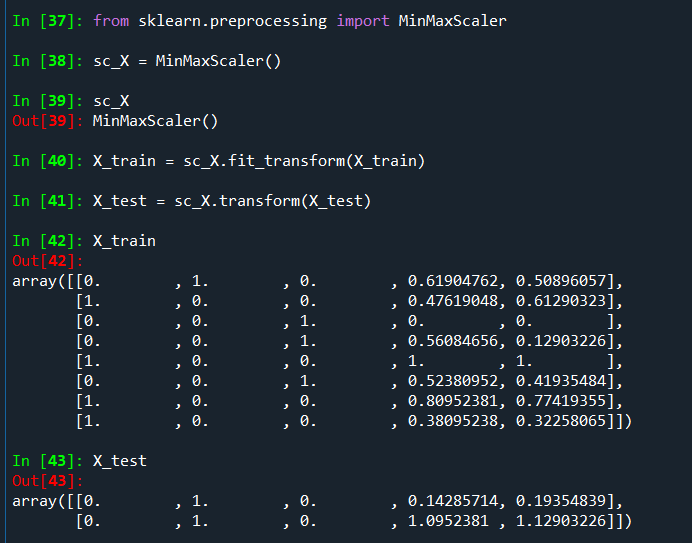


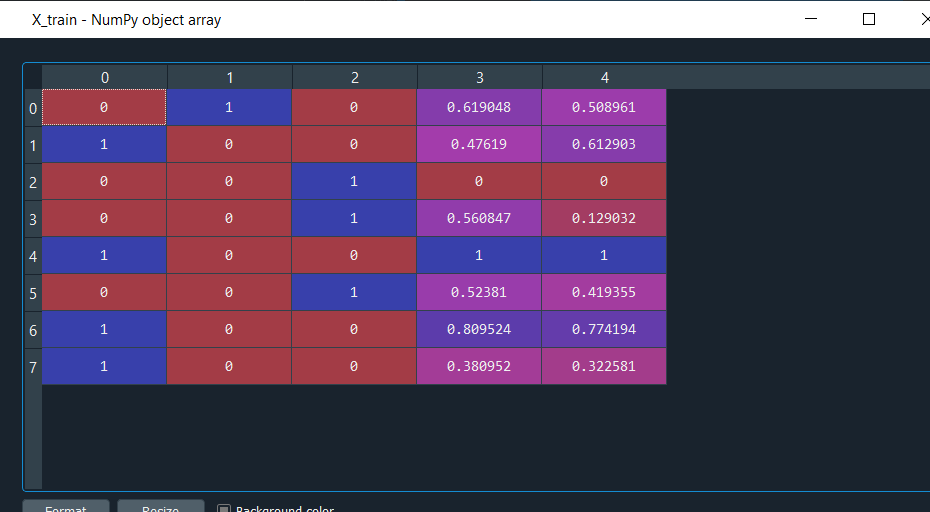




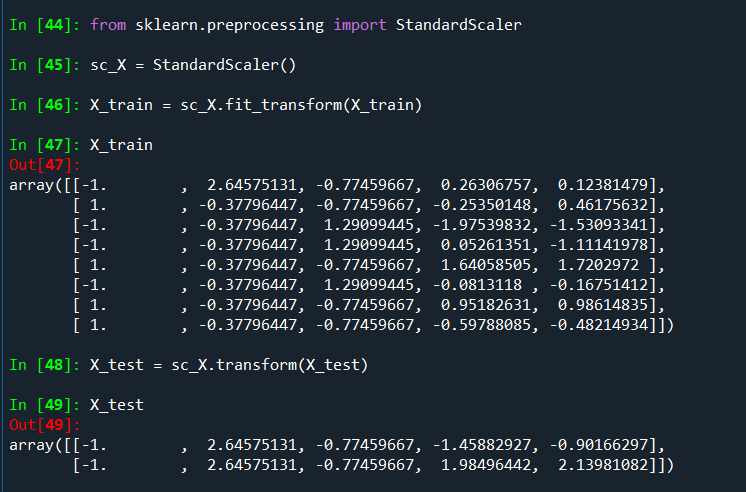




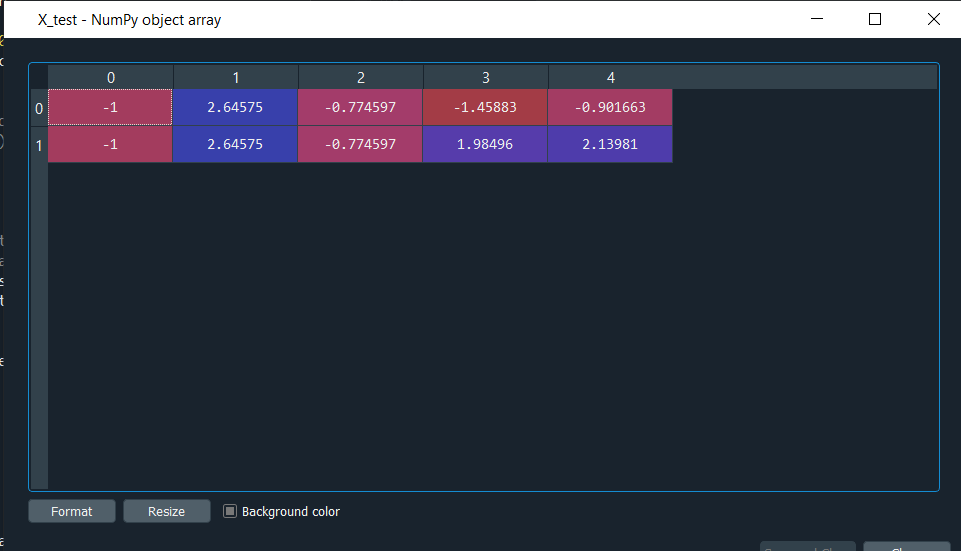












**DATA PREPROCESSING ON DIABETIES DATASET**

**CODE:**

import numpy as np

import pandas as pd

df=pd.read\_csv("C:\\Users\\tuhini\\Downloads\\pima-indians-diabetes.data.csv")

colm\_names=["preg","glucose","BP","SkinThickness","test","BMI","pedi","age","class"]

dataset=pd.read\_csv("C:\\Users\\tuhini\\Downloads\\pima-indians-diabetes.data.csv",names=colm\_names)

X = dataset.iloc[:, :-1].values

y = dataset.iloc[:, 3].values

from sklearn.impute import SimpleImputer

imputer = SimpleImputer(missing\_values = np.nan, strategy = 'mean')

imputer = imputer.fit(X[:, 1:3])

X[:, 1:3] = imputer.transform(X[:, 1:3])

from sklearn import preprocessing

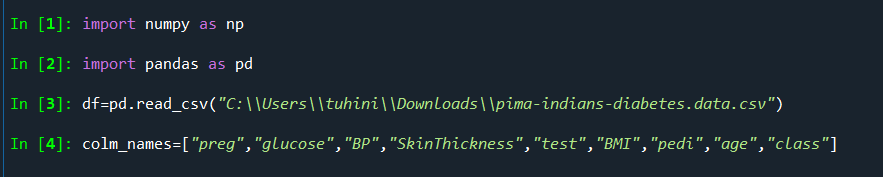
minscale=preprocessing.MinMaxScaler(feature\_range=(0,1))

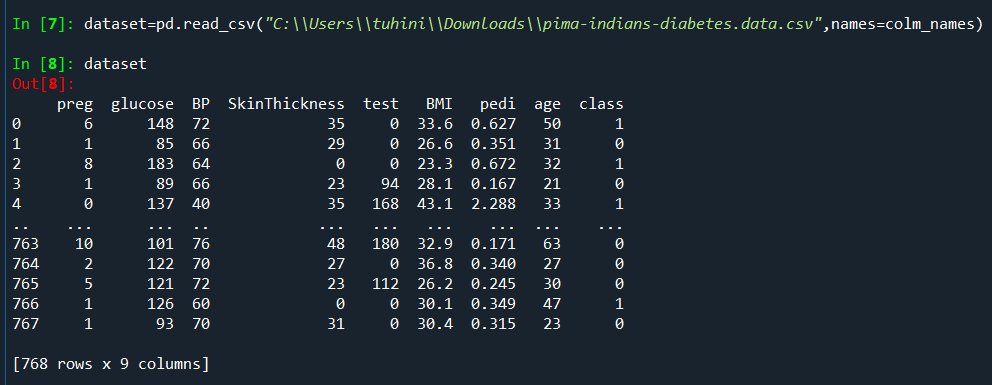
result1=minscale.fit\_transform(dataset)

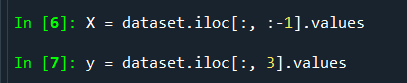
standardscale=preprocessing.StandardScaler()

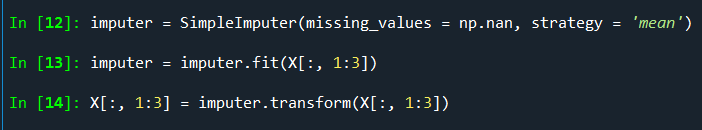
result2=standardscale.fit\_transform(dataset)

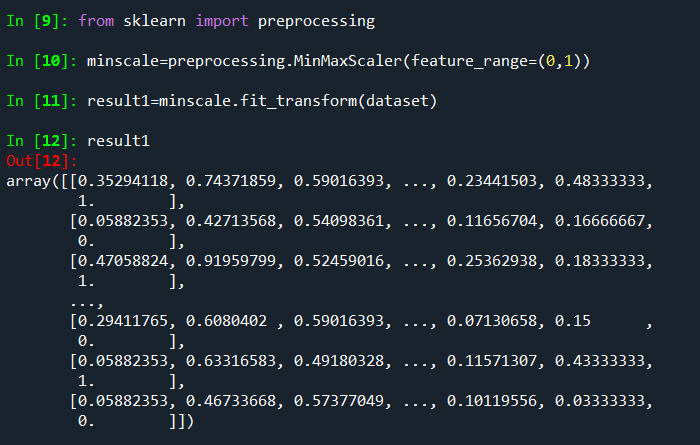
**OUTPUT:**

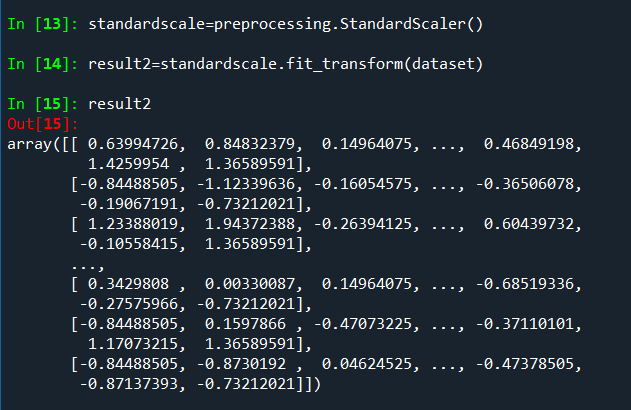


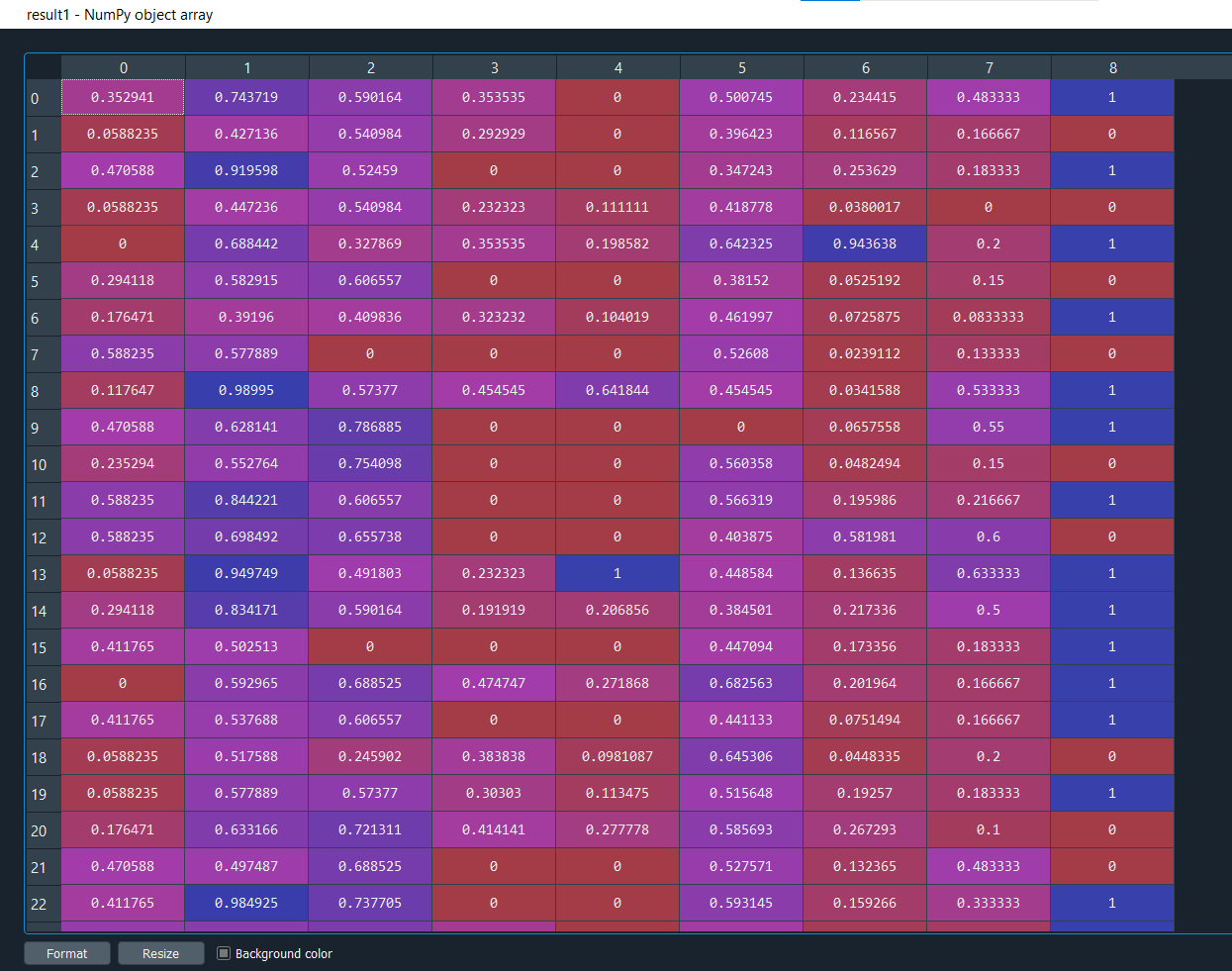


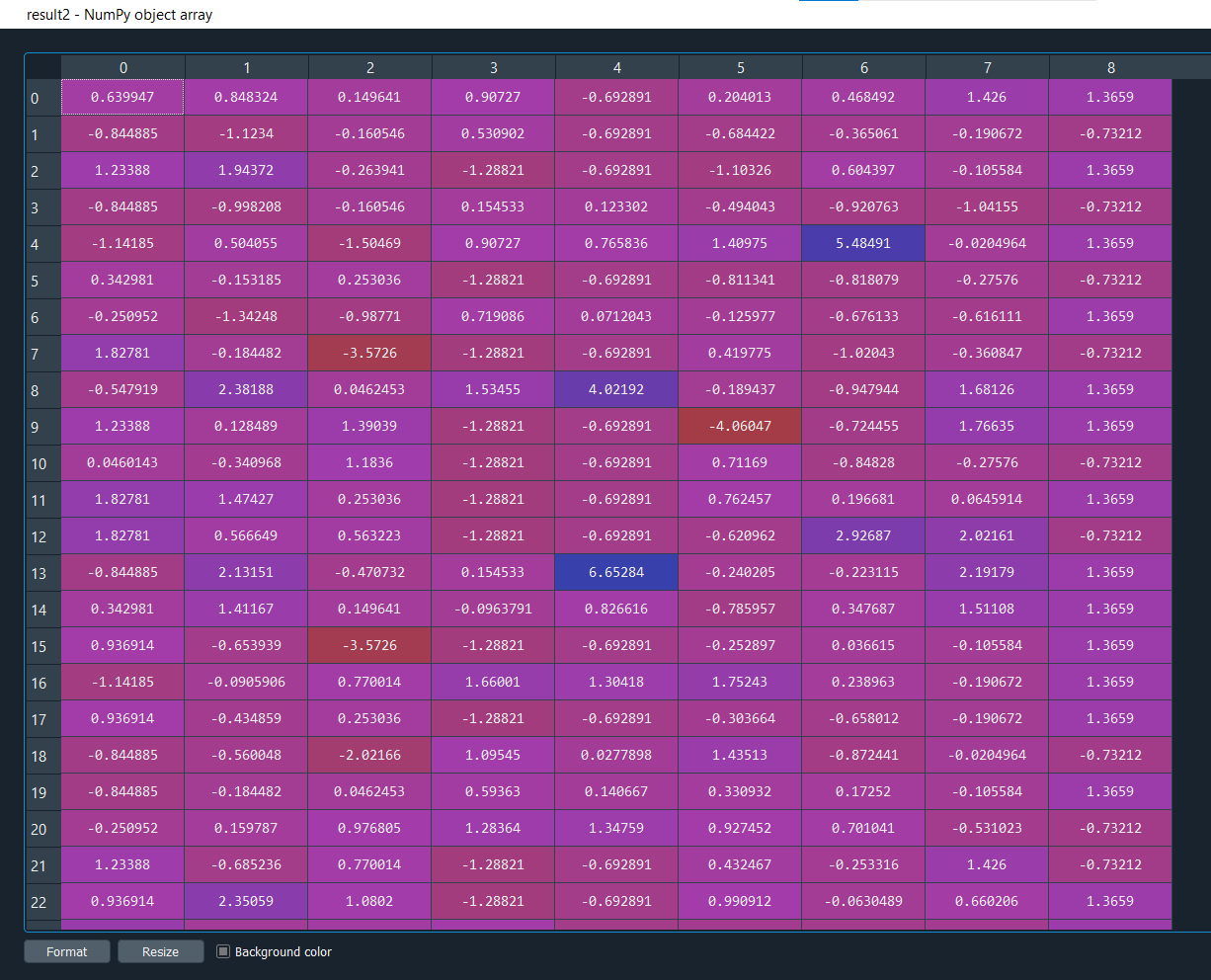




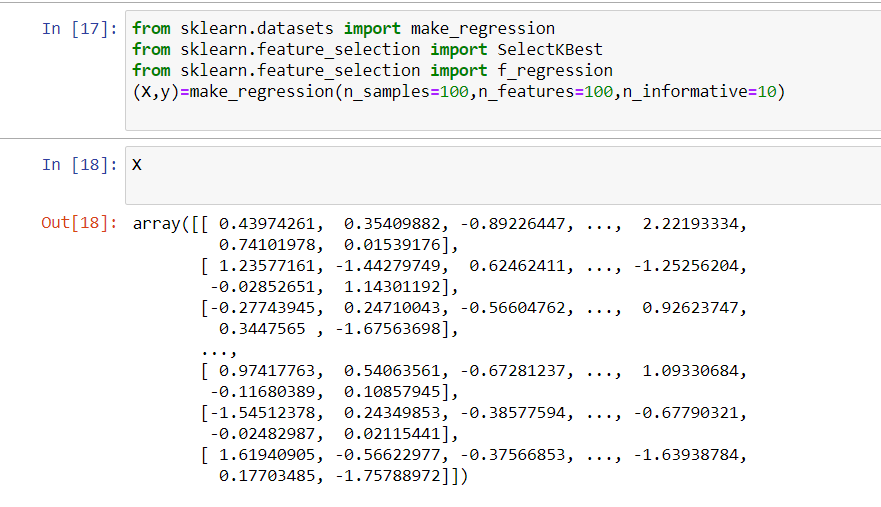


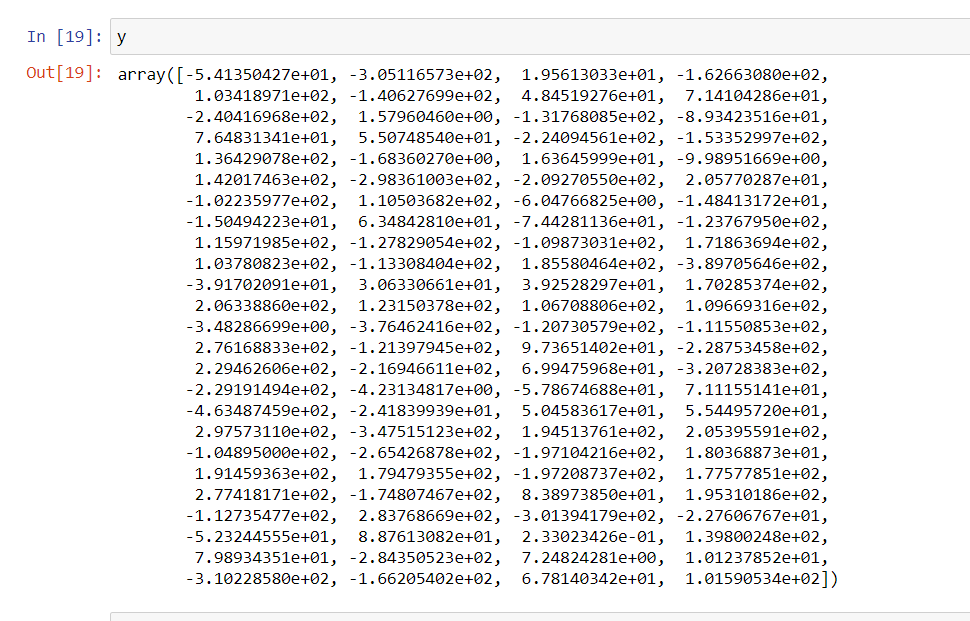


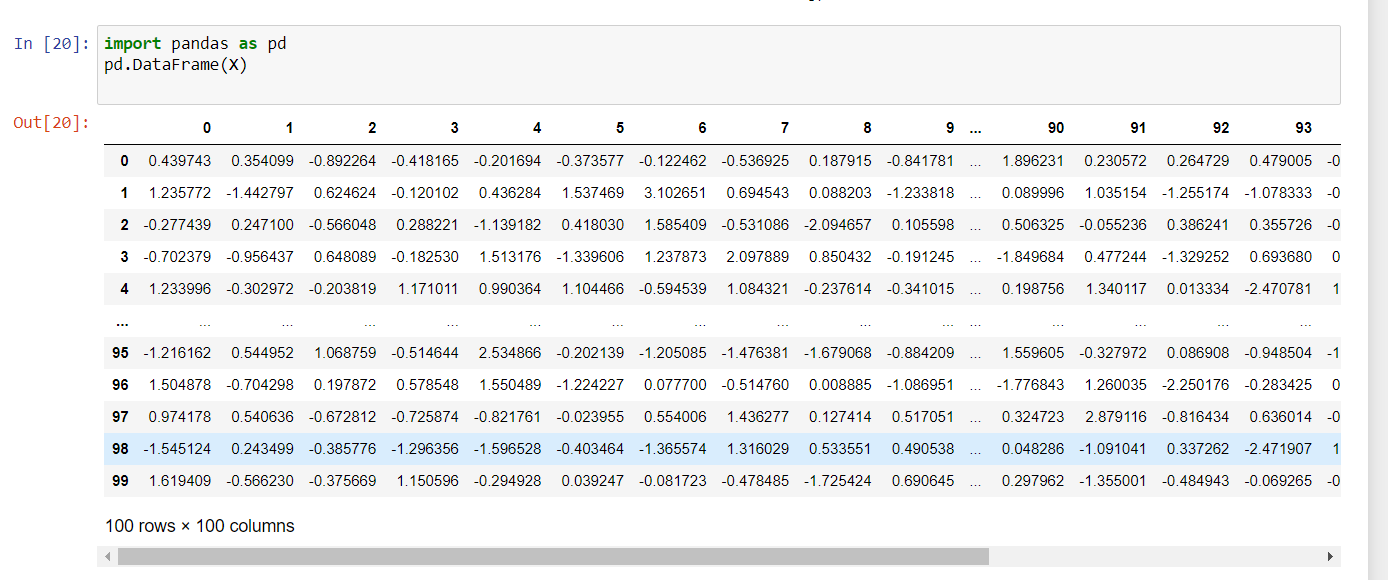


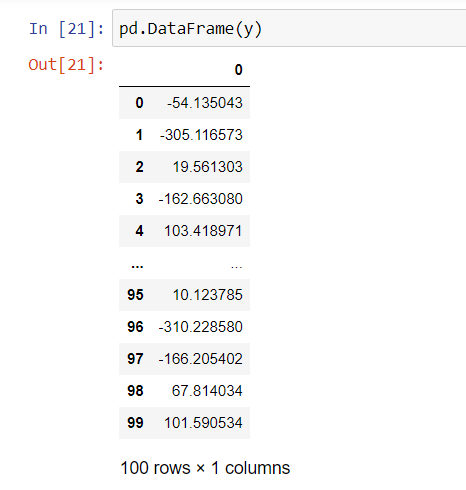


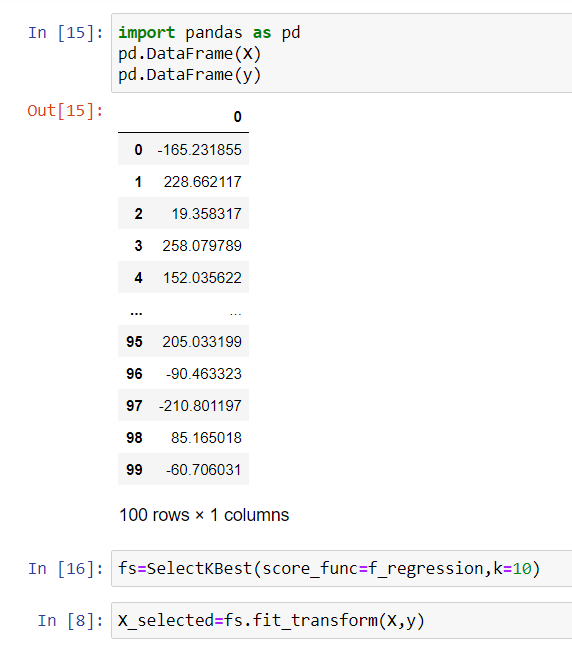
1. **Make Regression**

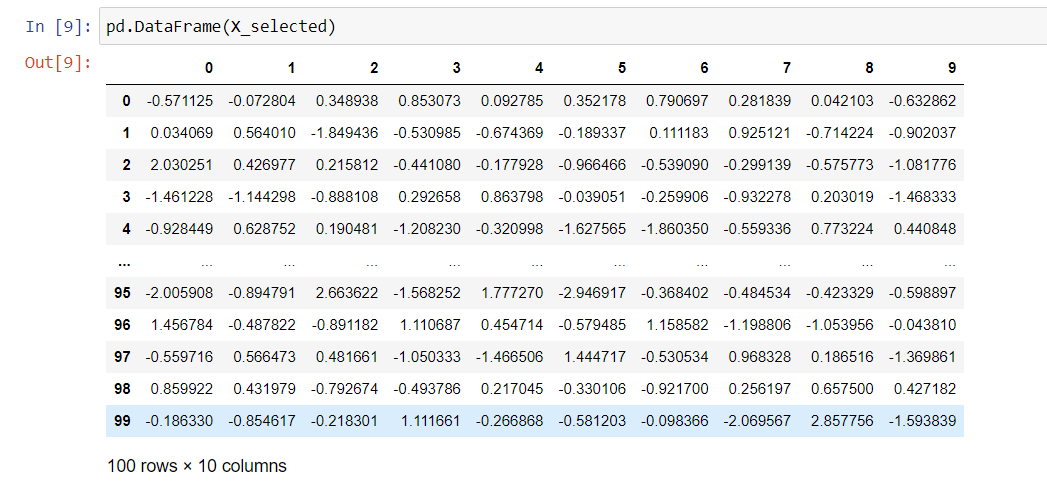












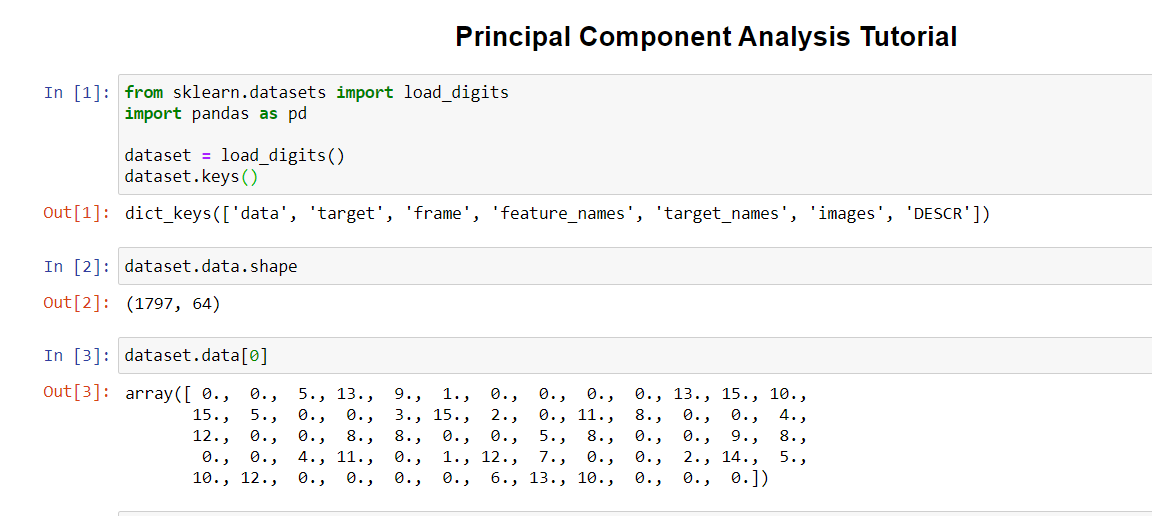
**EXPERIMENT NO. 5**

|  |
| --- |
| **Student Name and Roll Number: SUNIL KHARSU and 20csu371** |
| **Semester /Section: 5th , FSB** |
| **Link to Code:** |
| **Date: 09/09/21** |
| **Faculty Signature:** |
| **Grade:** |

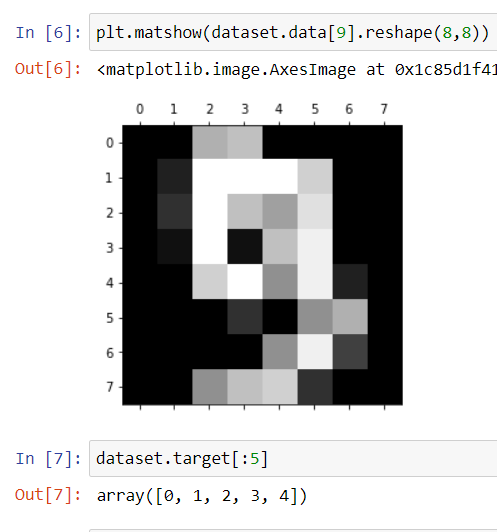
|  |
| --- |
| **Objective(s):**   * Study Dimensionality Reduction. * Understand the basic principle behind Principal Component Analysis. |
| **Outcome:**  Students will be familiarized with Dimensionality Reduction especially Principal Component  Analysis (PCA). |
| **Problem Statement:**  Reduce dimensionality of Iris dataset using Principal Component Analysis. |
| **Background Study:** Principal component analysis is a statistical technique that is used to analyze the interrelationships among a large number of variables and to explain these variables in terms of a smaller number of variables, called principal components, with a minimum loss of information. |
| **Question Bank:**  **1. What is dimensionality reduction?**  Dimensionality reduction, or dimension reduction, is **the transformation of data from a high-dimensional space into a low-dimensional space so that the low-dimensional representation retains some meaningful properties of the original data** , ideally close to its intrinsic dimension.  **2. Differentiate between Feature Selection, Feature Engineering and Dimensionality Reduction**.  Feature engineering enables **you to build more complex models than you could with only raw data**. It also allows you to build interpretable models from any amount of data.  Feature selection is simply selecting and excluding given features without changing them. **Dimensionality reduction transforms features into a lower dimension**.  **3. What are principal components?**  Principal components are **new variables that are constructed as linear combinations or mixtures of the initial variables** . ... Geometrically speaking, the principal components represent the directions of the data that explain a maximal amount of variance, that is to say, the lines that capture most information of the data. |

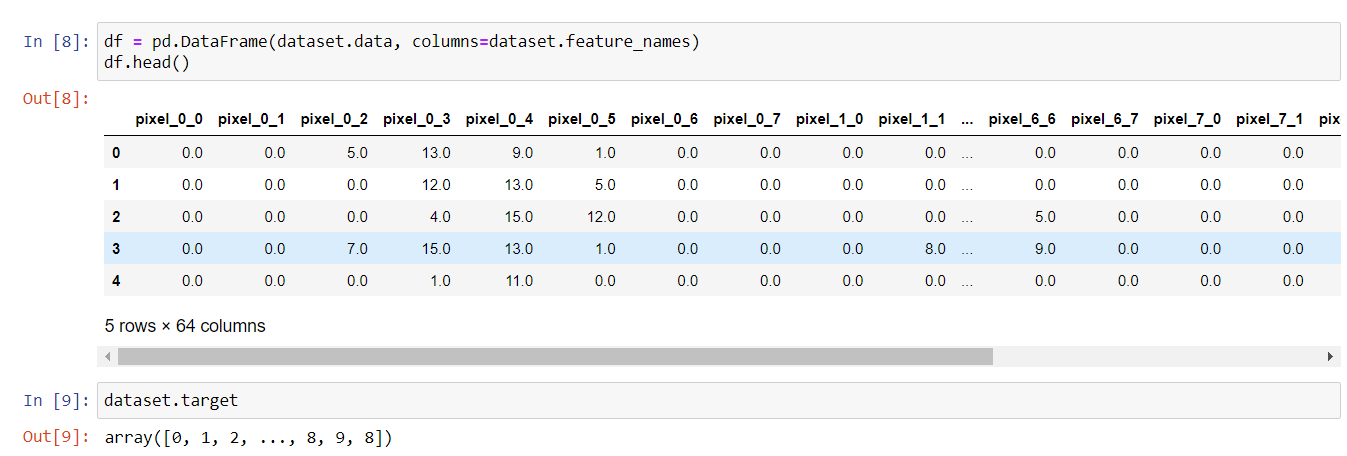
**Student Work Area**

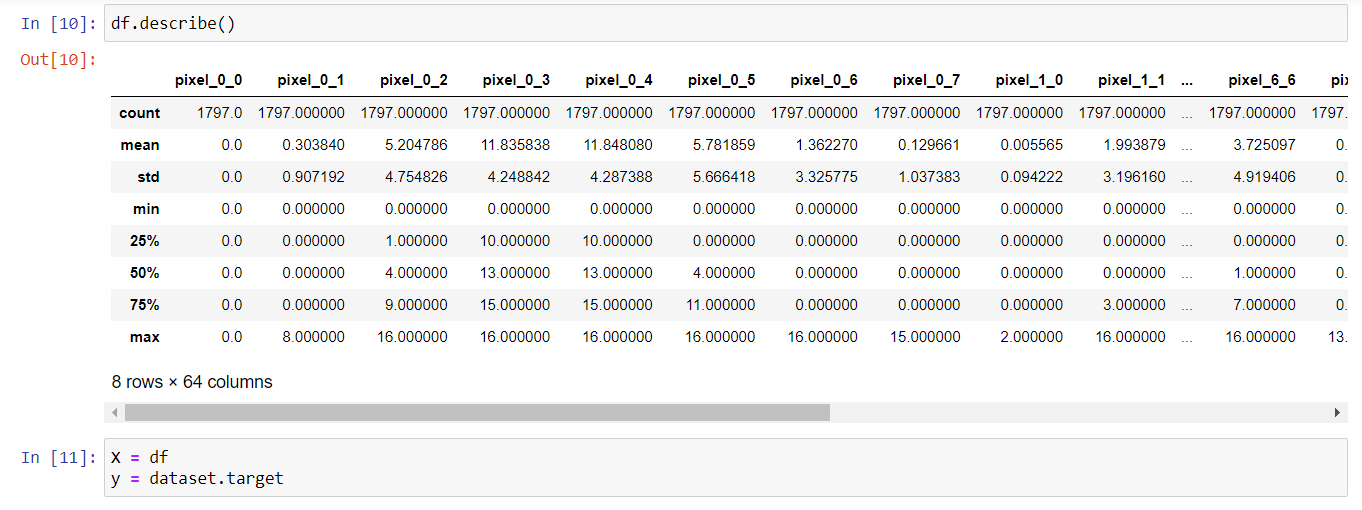
**Algorithm/Flowchart/Code/Sample Outputs**

****

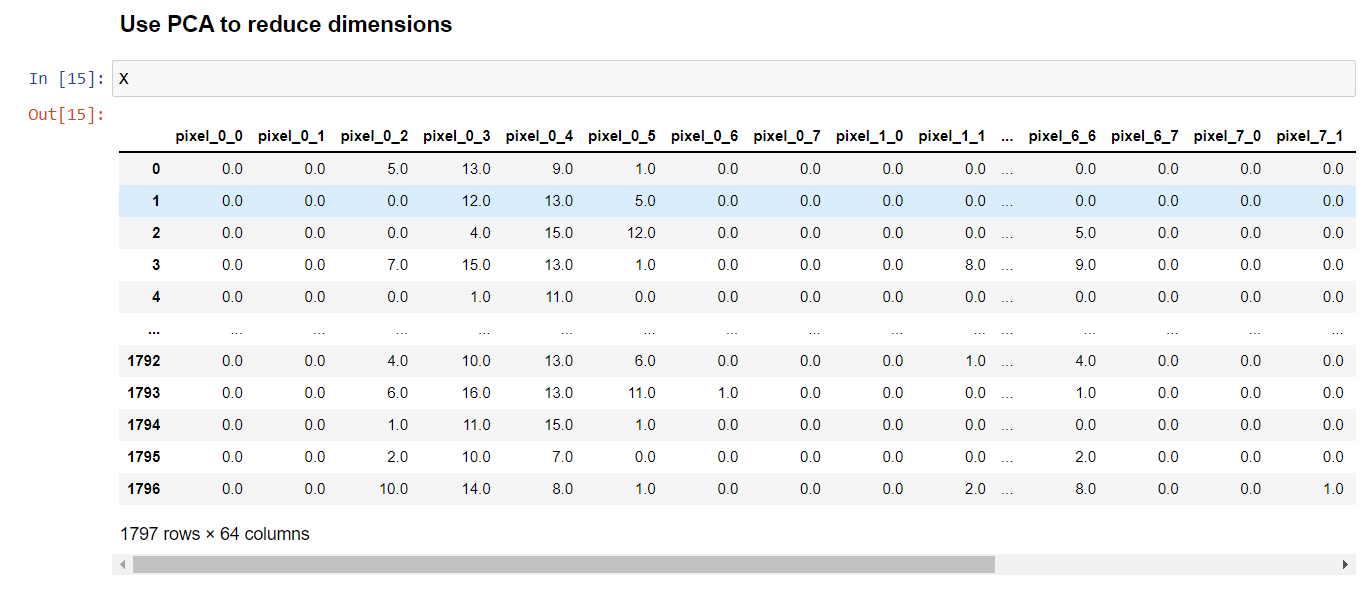
****

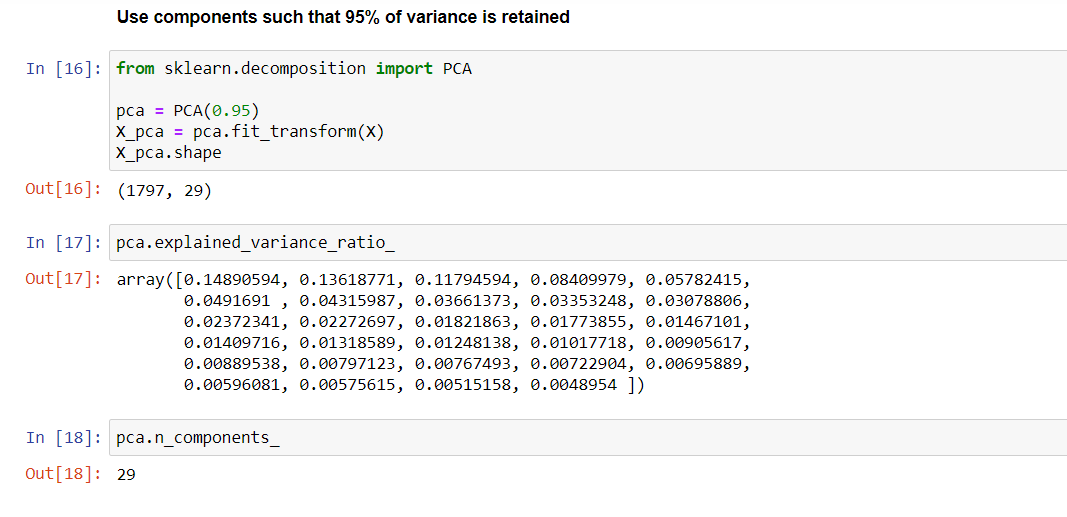
****

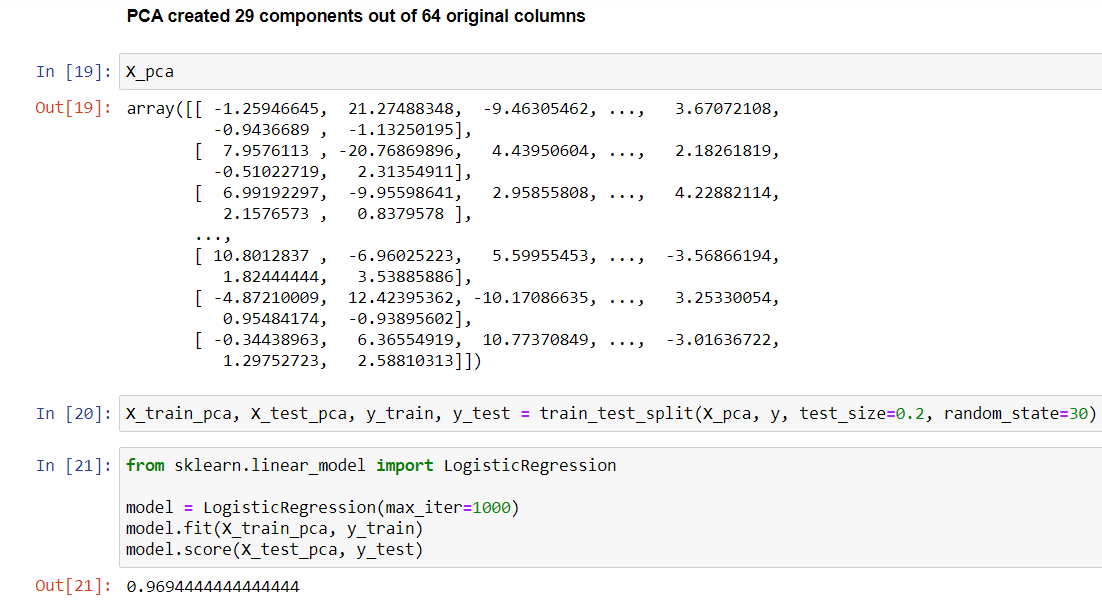
****

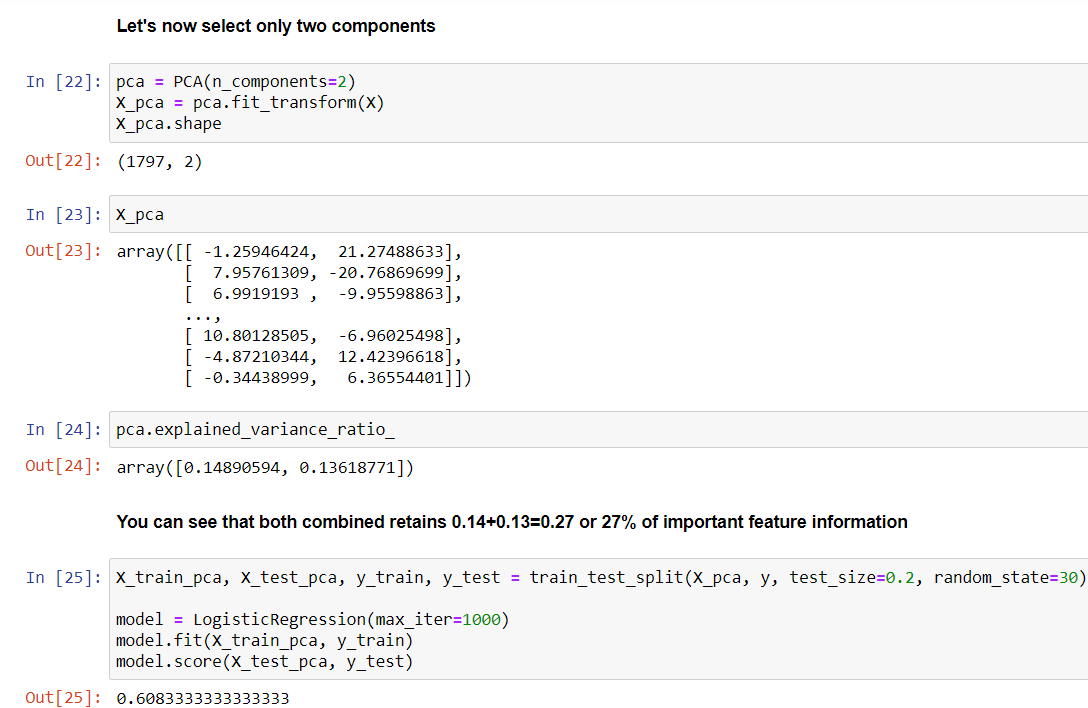
****

****

****

****

****

****

****

**EXPERIMENT NO. 6**

|  |
| --- |
| **Student Name and Roll Number: SUNIL KHARSU and 20csu371** |
| **Semester /Section: 5th, FSB** |
| **Link to Code:** |
| **Date: 30/09/21** |
| **Faculty Signature:** |
| **Grade:** |

|  |
| --- |
| **Objective(s):**   * Understand Simple Linear Regression (SLR). * Study about the different performance metrics of SLR. |
| **Outcome:**  Student will be familiarized with regression problems and SLR as a solution to single feature problem. |
| **Problem Statement:**  To apply Simple Linear Regression on the given dataset. |
| **Background Study:**  Simple linear regression isan approach for predicting a response using a single feature. It is assumed that the two variables are linearly related. Hence, we try to find a linear function that predicts the response value(y) as accurately as possible as a function of the feature or independent variable(x). |
| **Question Bank:**  **1. What is a regression problem?**  A regression problem is **when the output variable is a real or continuous value, such as "salary" or "weight"** . Many different models can be used, the simplest is the linear regression. It tries to fit data with the best hyper-plane which goes through the points.  **2. How Simple Linear Regression (SLR) helps in solving regression problems containing an input feature and an output variable?**  Input variables can also be termed as Independent/predictor variables, and the output variable is called the dependent variable. A simple linear regression model is a mathematical equation that allows us to predict a response for a given predictor value. ... The y-intercept is the predicted value for the response (y) when **x = 0** . The slope describes the change in y for each one unit change in x.  **3. What are the different performance metrics that can be used for evaluating SLR?**   * Mean Absolute Error(MAE) * Mean Squared Error(MSE) * Root Mean Squared Error(RMSE) * Root Mean Squared Log Error(RMSLE) * R Squared (R2) |

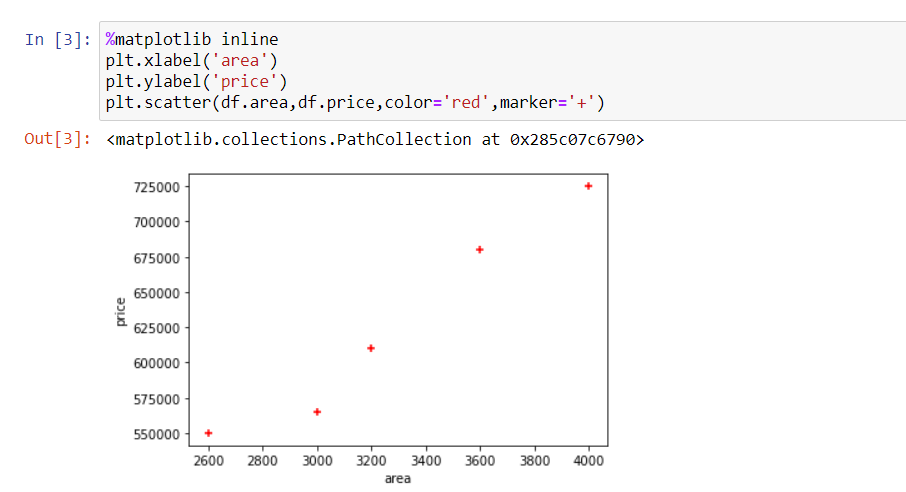
**Student Work Area**

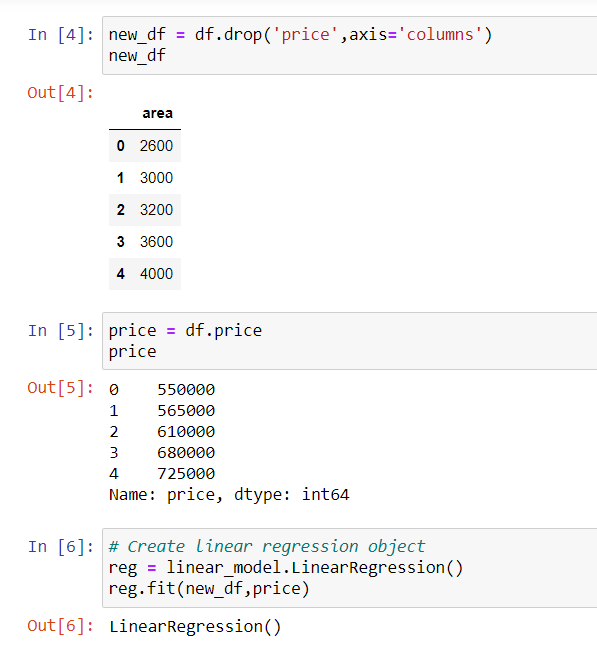
**Algorithm/Flowchart/Code/Sample Outputs**

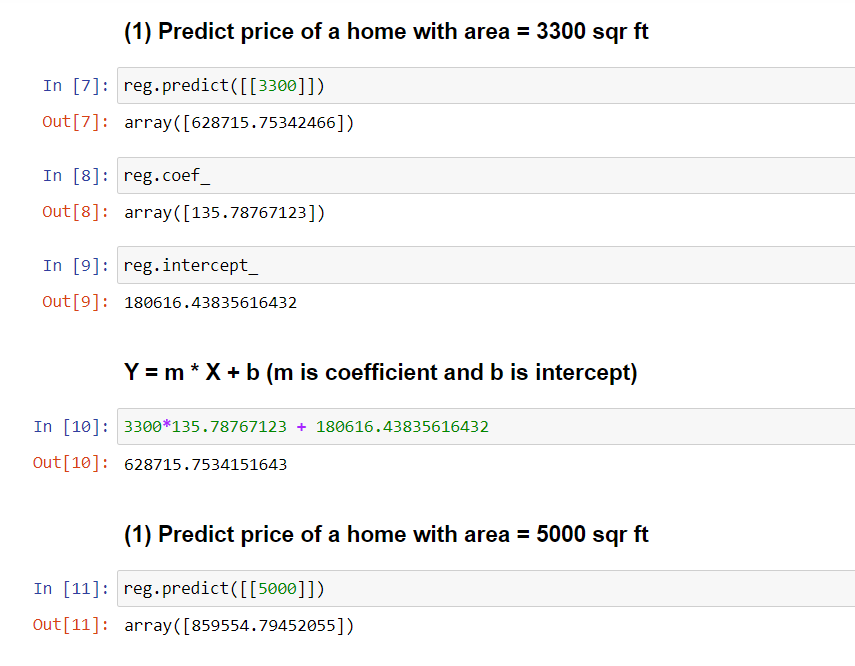
**Linear Regression**

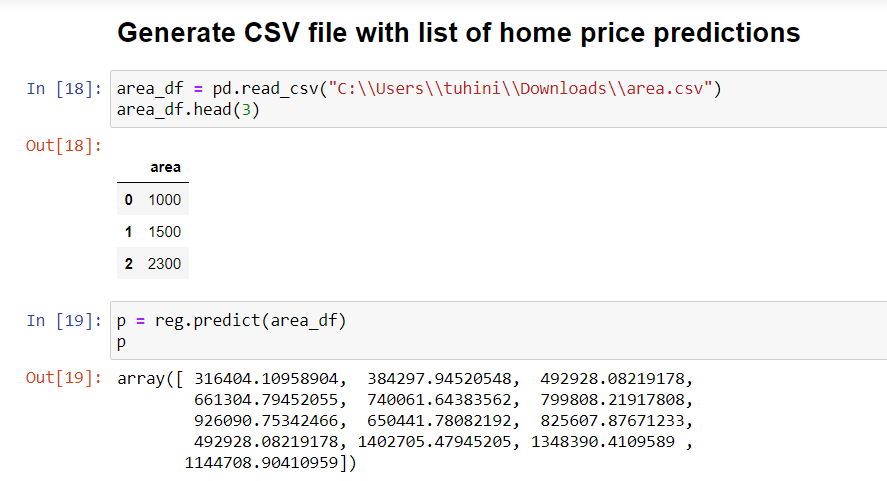
* 1. **House Price Prediction**

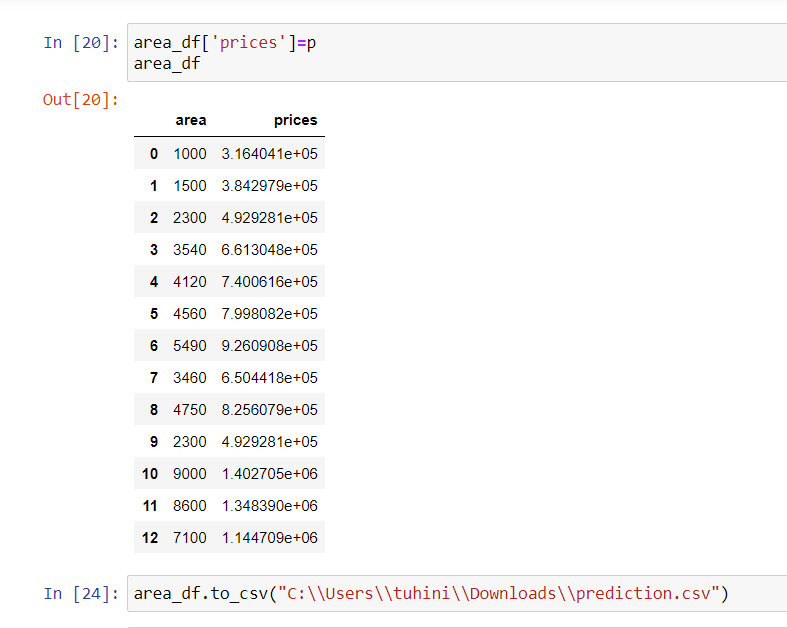
****

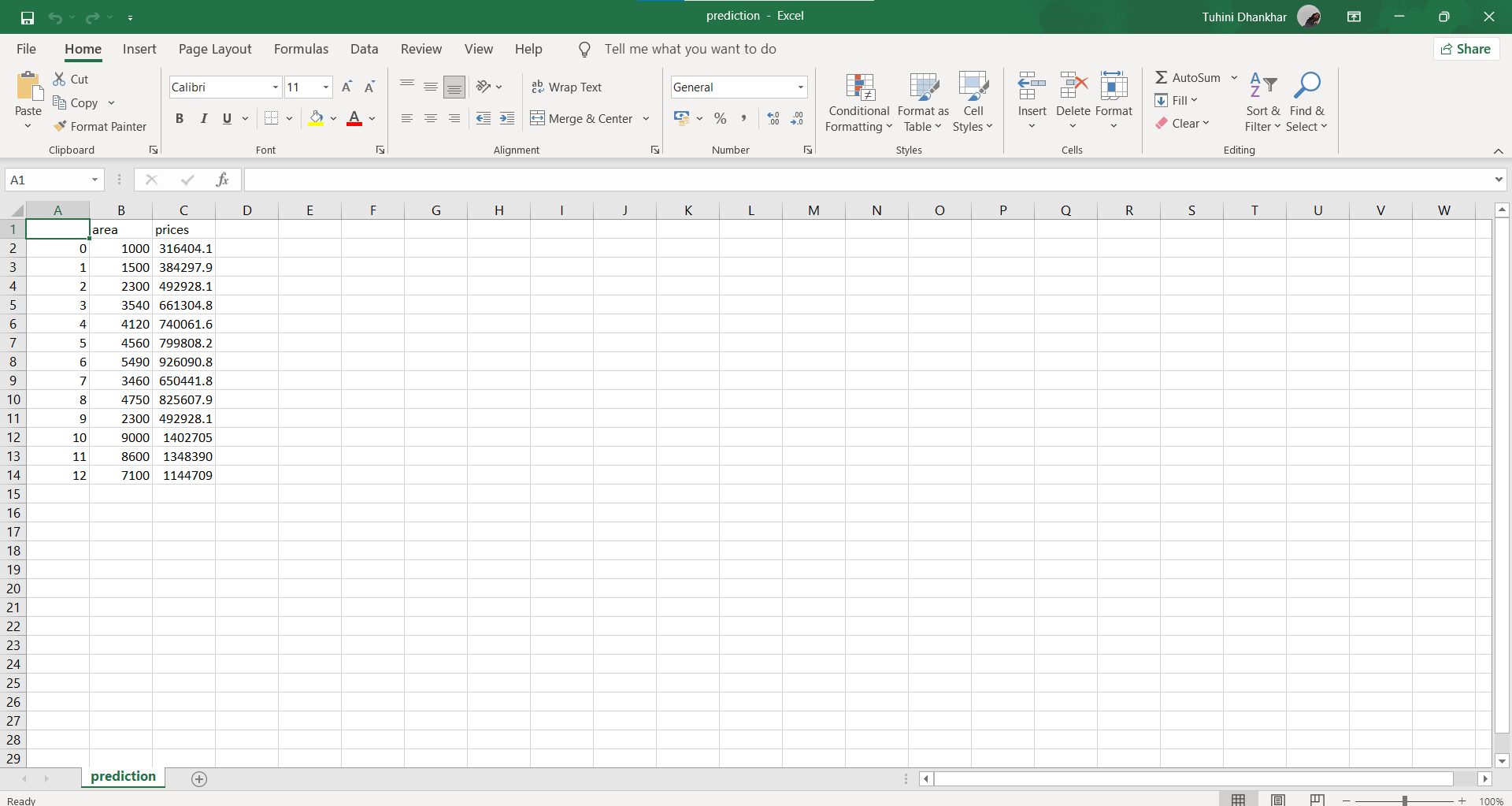
****

****

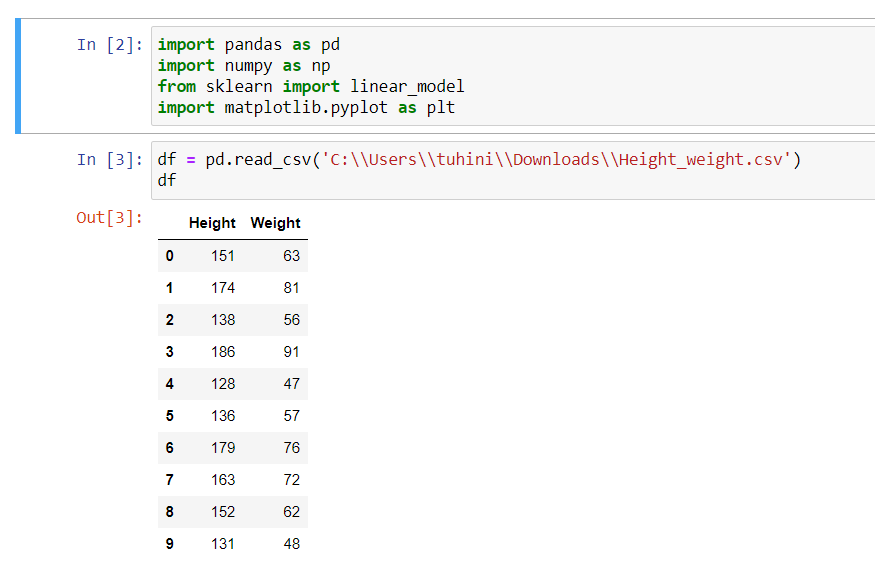
****

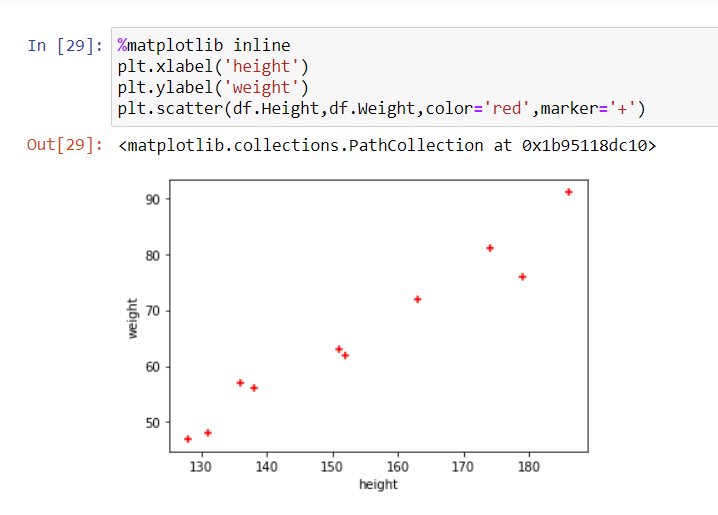
****

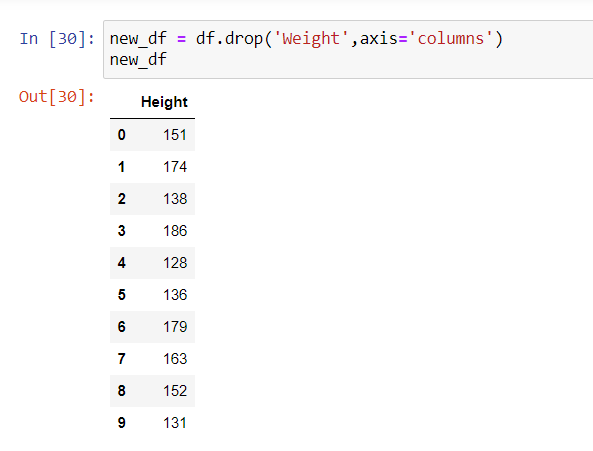
****

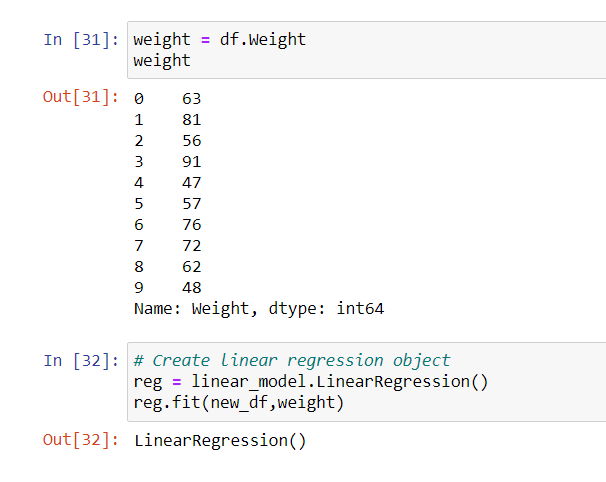
****

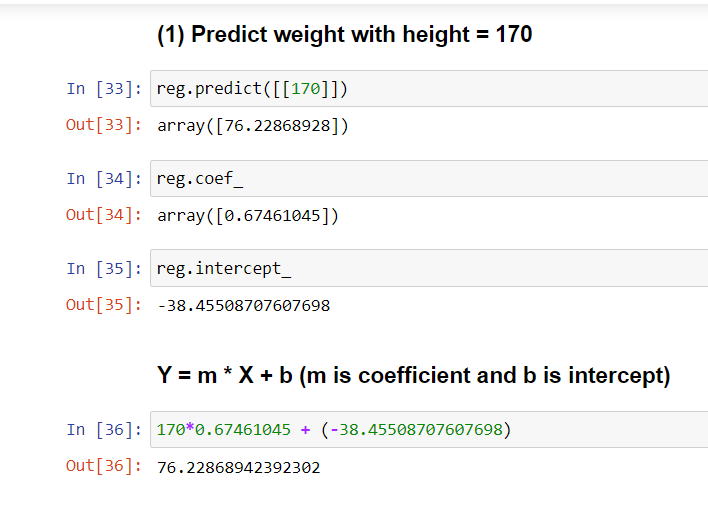
* 1. **Height Weight Prediction**

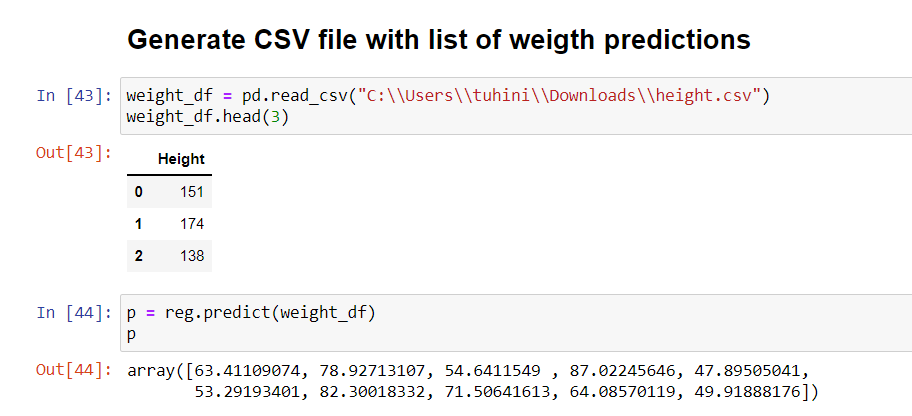
****

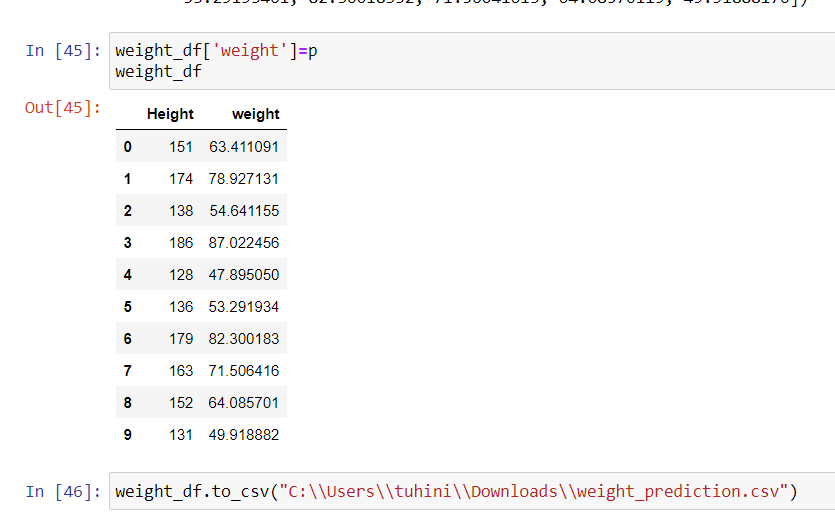
****

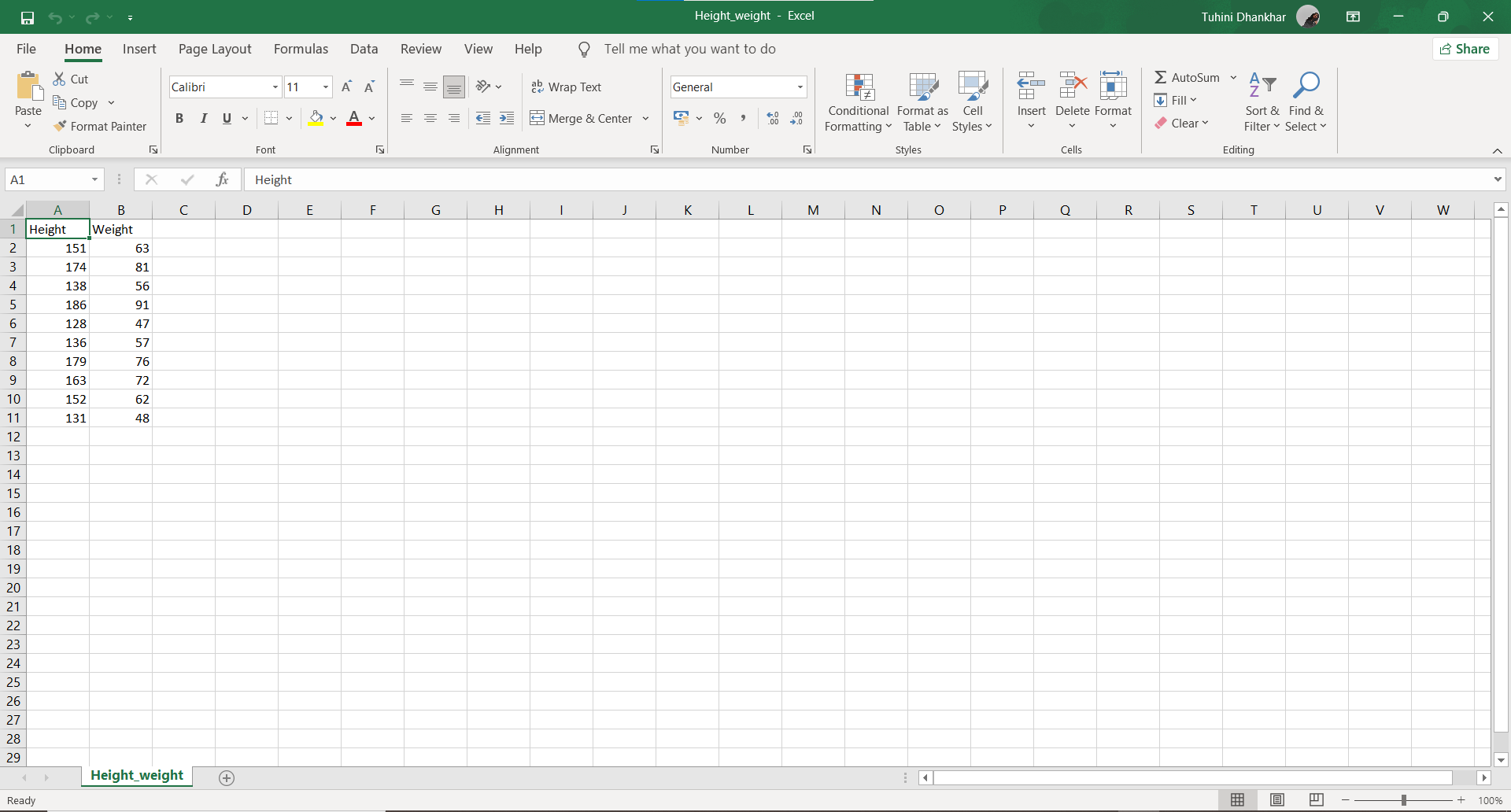
****

****

****

****

****

****

**EXPERIMENT NO. 7**

|  |
| --- |
| **Student Name and Roll Number: SUNIL KHARSU and 20csu371** |
| **Semester /Section: 5th, FSB** |
| **Link to Code:** |
| **Date: 07/10/21** |
| **Faculty Signature:** |
| **Grade:** |

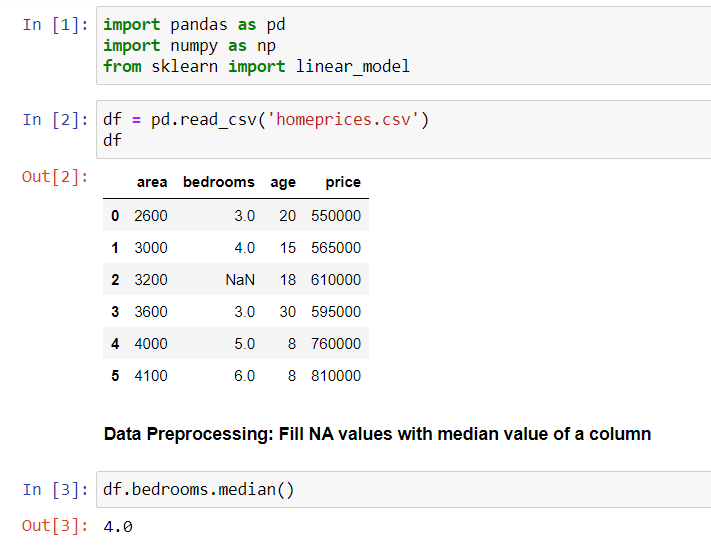
|  |
| --- |
| **Objective(s):**   * Understand mathematics behind Multiple Linear Regression (MLR). * Solving linear regression problems containing more than one independent feature using MLR. |
| **Outcome:**  Students will be familiarized with Multiple Linear Regression for solving linear regression problems. |
| **Problem Statement:**  To apply multiple linear regression on any regression dataset. |
| **Background Study:**  Multiple Linear Regressionattempts to model the relationship between two or more features and a response by fitting a linear equation to observed data. |
| **Question Bank:**  **1. What is MLR?**  **Multiple linear regression** (MLR), also known simply as multiple regression, is a statistical technique that uses several explanatory variables to predict the outcome of a response variable. Multiple regression is an extension of linear (OLS) regression that uses just one explanatory variable  **2. Differentiate between SLR and MLR?**  SLR examines the **relationship between the dependent variable and a single independent variable** . MLR examines the relationship between the dependent variable and multiple independent variables. |

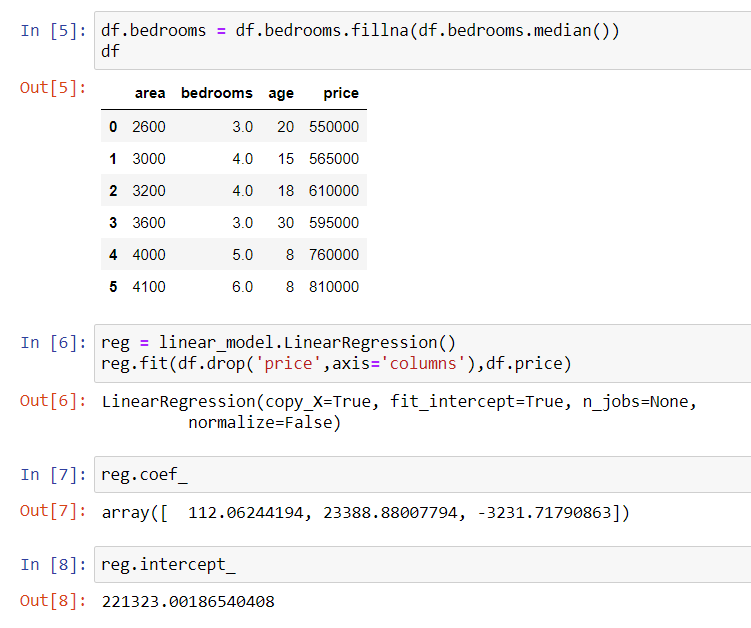
**Student Work Area**

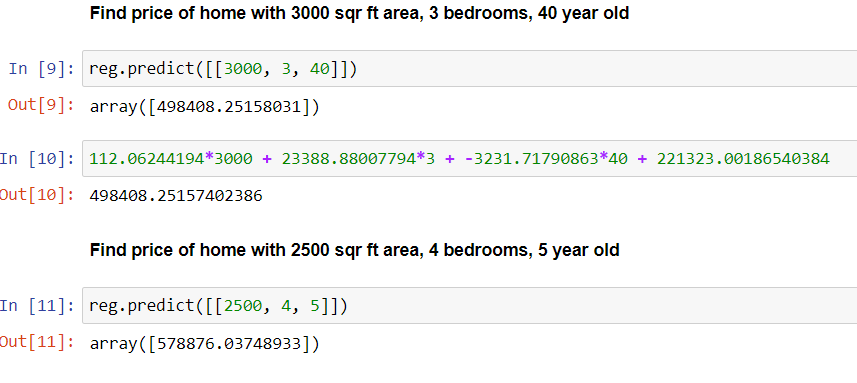
**Algorithm/Flowchart/Code/Sample Outputs**

**Multiple linear regression**

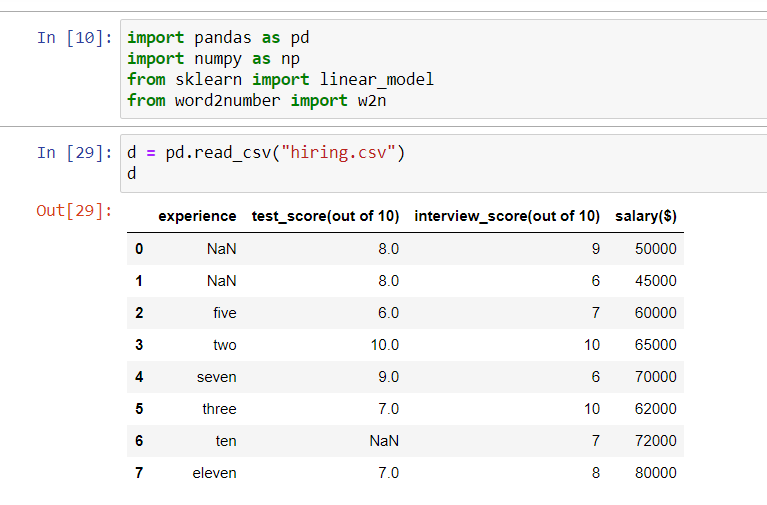
* + - 1. **House Prices**

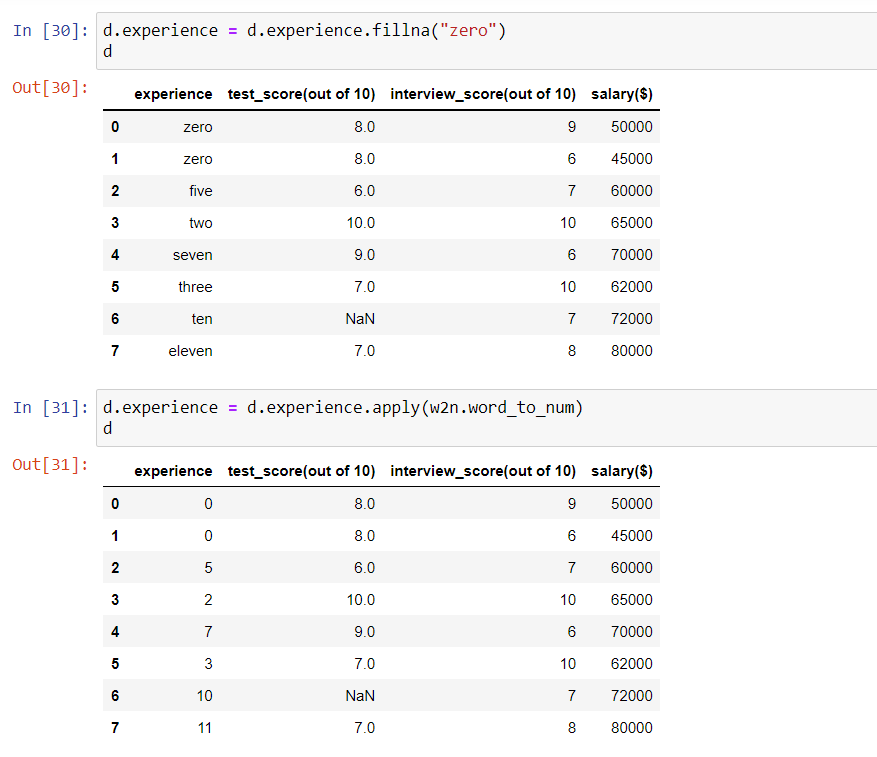
****

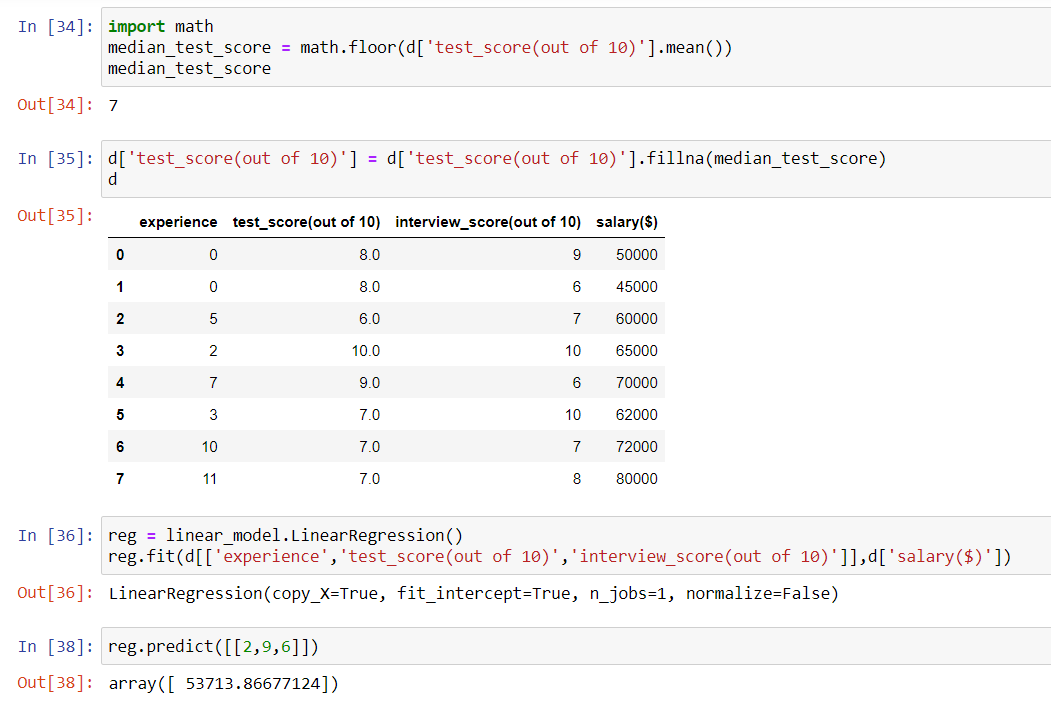
****

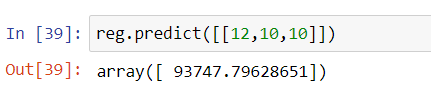
****

* + - 1. **Hiring**

****

****

****

****

**EXPERIMENT NO. 8**

|  |
| --- |
| **Student Name and Roll Number: SUNIL KHARSU and 20csu371** |
| **Semester /Section: 5th, FSB** |
| **Link to Code:** |
| **Date: 14/10/21** |
| **Faculty Signature:** |
| **Grade:** |

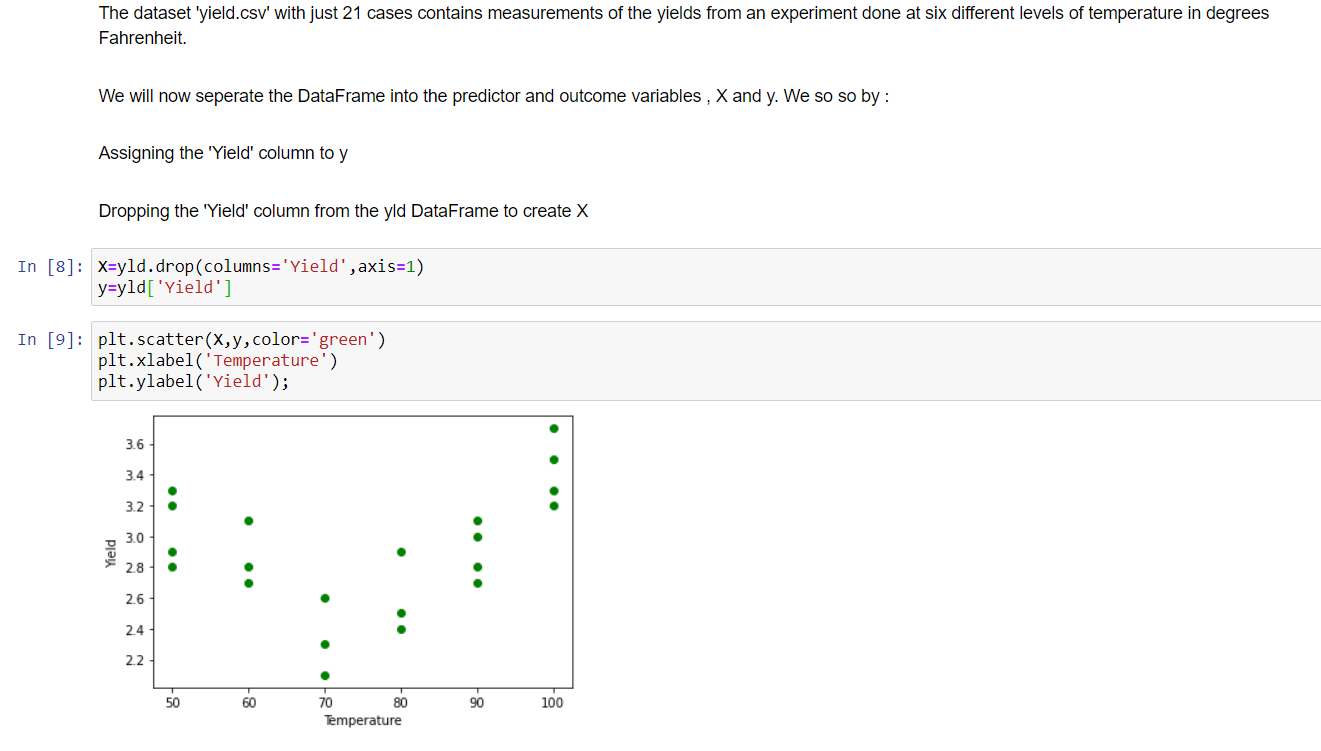
|  |
| --- |
| **Objective(s):**   * Study and understand about Polynomial Regression on non-linear regression data. * Study the mathematics behind Polynomial Regression. |
| **Outcome:**  Students will be familiarized with handling of regression data having non-linear relationship between input and output. |
| **Problem Statement:**  To apply Polynomial Linear Regression on the given dataset. |
| **Background Study:**  **Polynomial Regression**is a form of linear regression in which the relationship between the independent variable x and dependent variable y is modeled as an nth degree polynomial. Polynomial regression fits a nonlinear relationship between the value of x and the corresponding conditional mean of y, denoted E(y |x). |
| **Question Bank:**  1. What is non-linear relationship between input and output?  In a nonlinear relationship, changes in the output do not change in direct proportion to changes in any of the inputs.  2. How Polynomial Regression is used to handle non-linear relationship? |

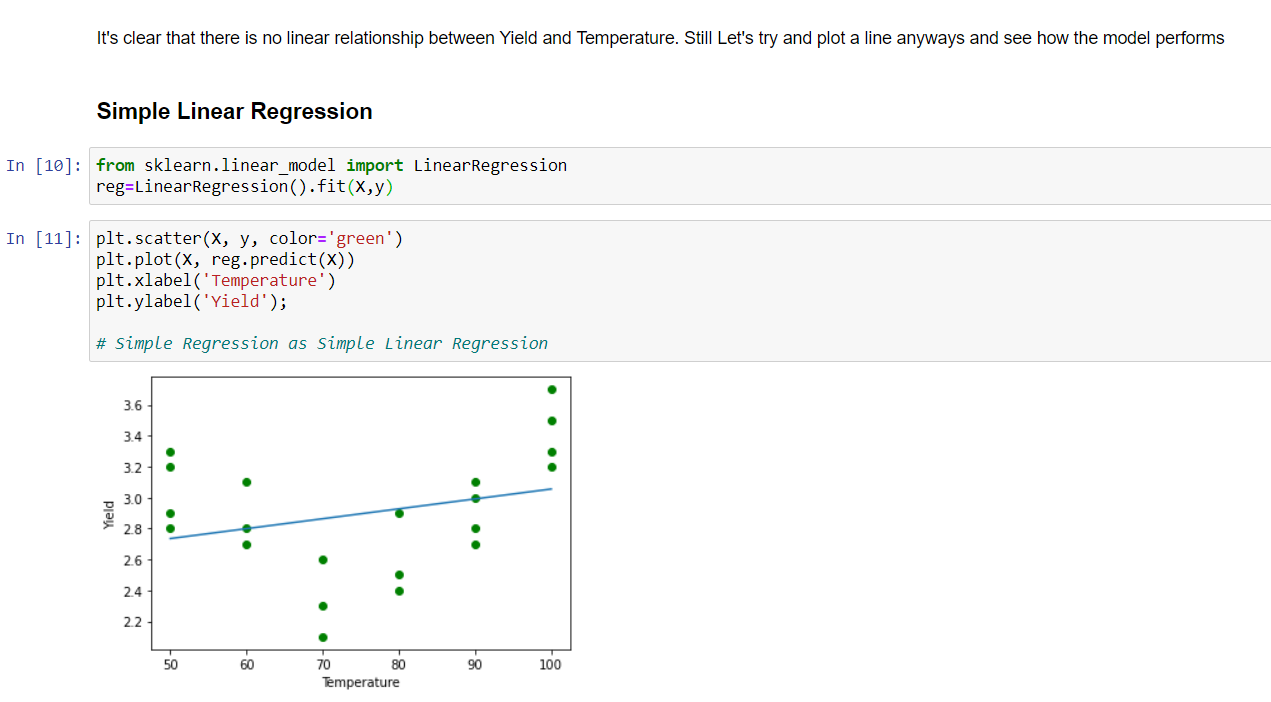
**Student Work Area**

**Algorithm/Flowchart/Code/Sample Outputs**

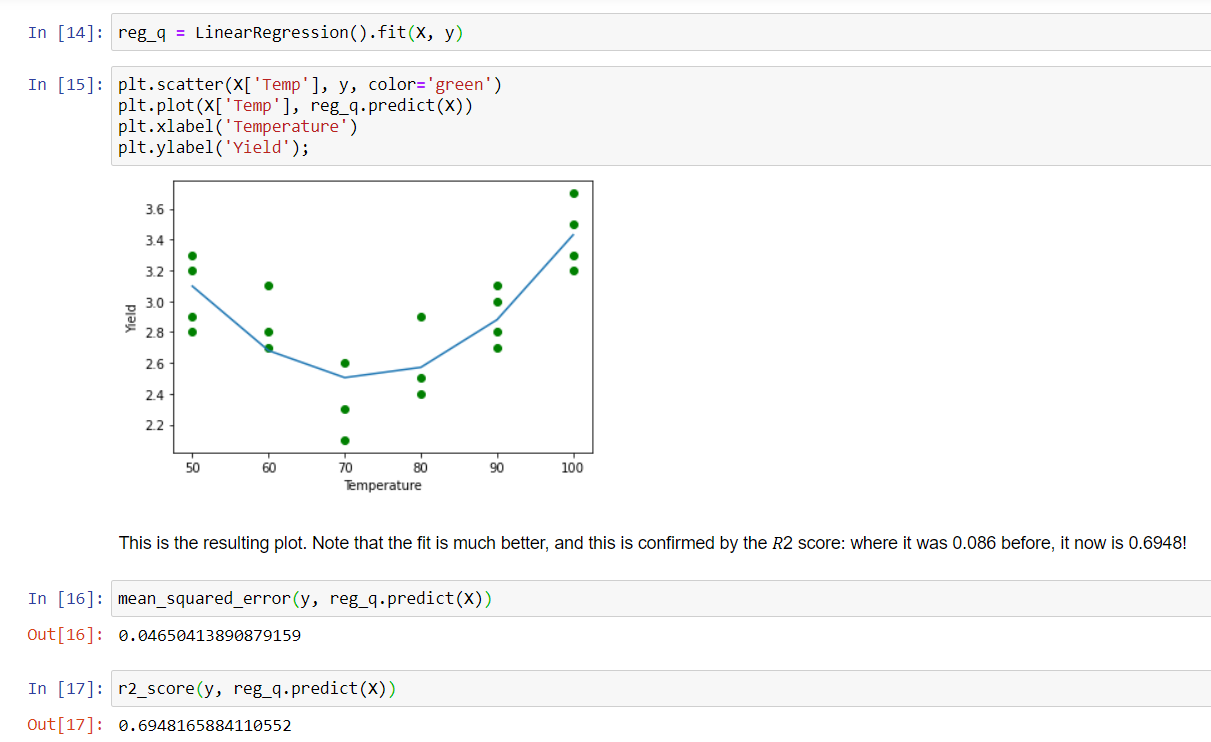
**Polynomial Regression**

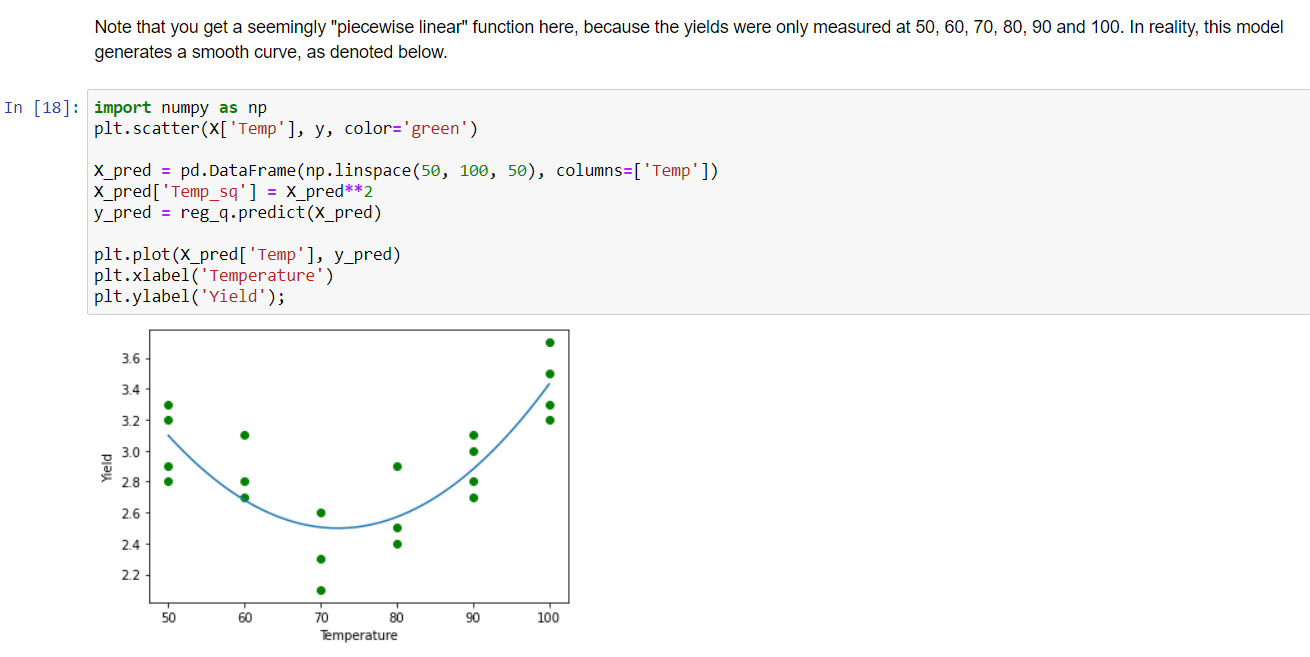
****

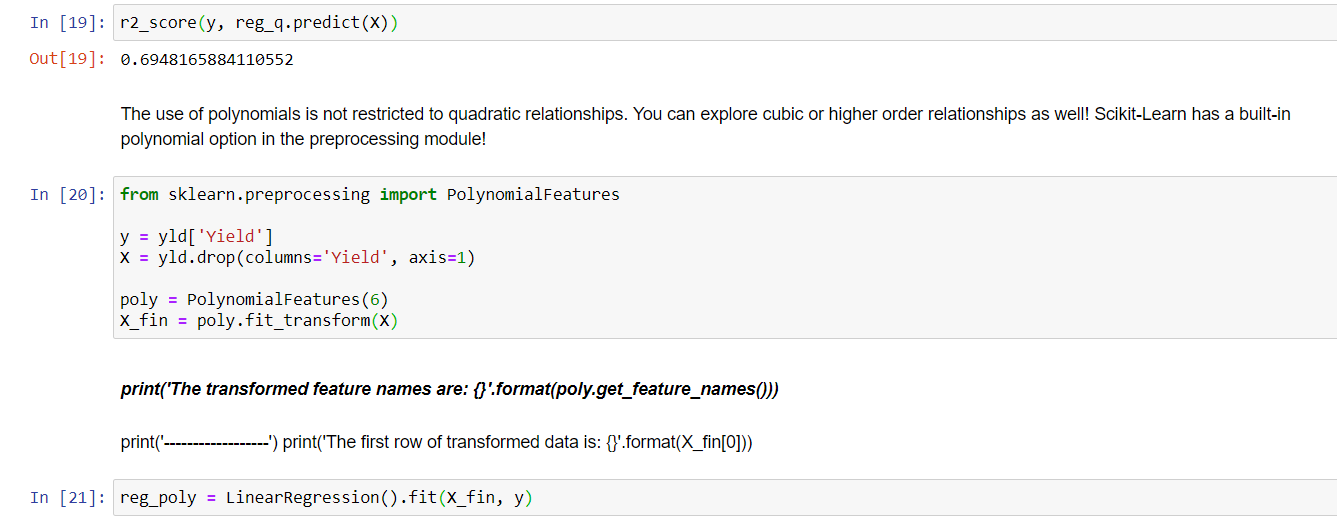
****

****

****

****

****

****

****

**EXPERIMENT NO. 9**

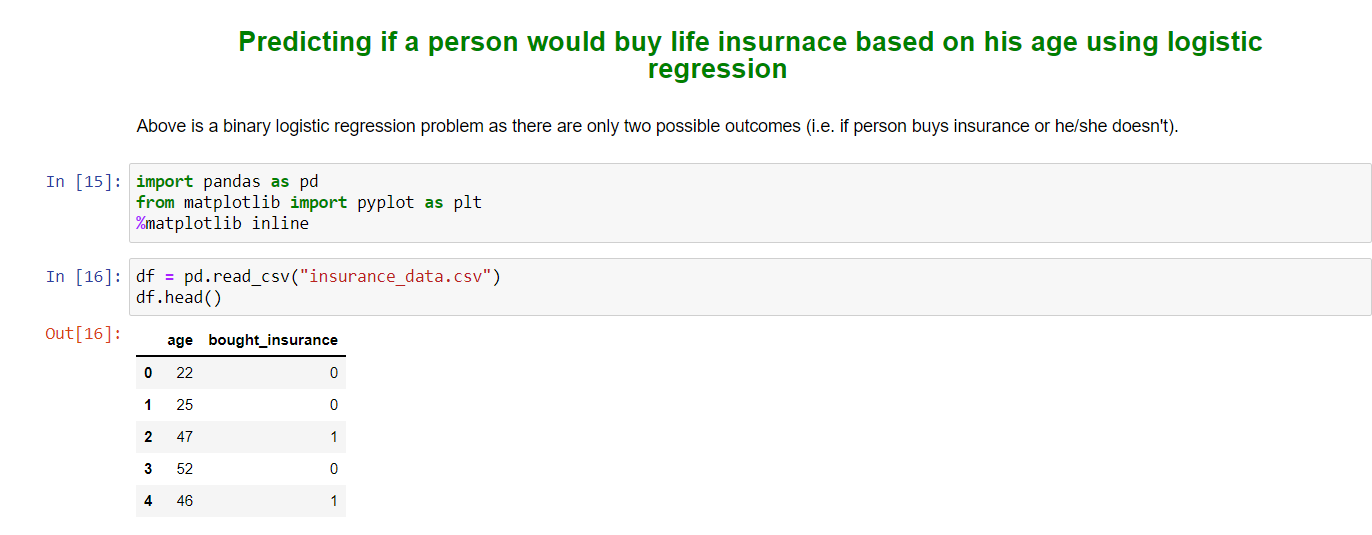
|  |
| --- |
| **Student Name and Roll Number: SUNIL KHARSU and 20csu371** |
| **Semester /Section: 5th, FSB** |
| **Link to Code:** |
| **Date: 21/10/21** |
| **Faculty Signature:** |
| **Grade:** |

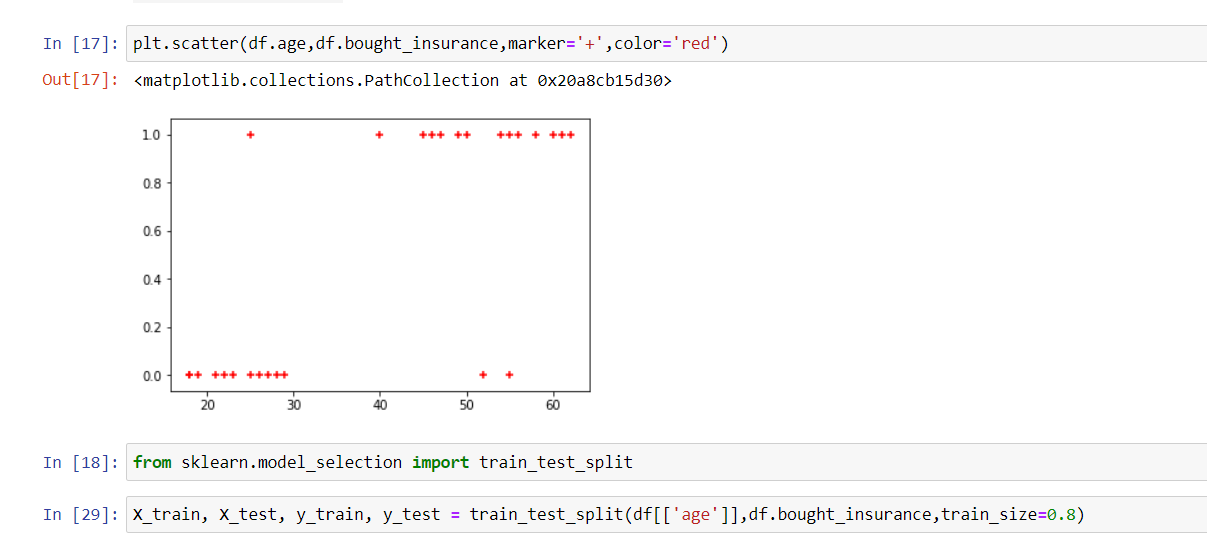
|  |
| --- |
| **Objective(s):**   * Study Logistic Regression. * How Logistic Regression is used to solve classification problems. |
| **Outcome:**  Students will be familiarized with Logistic Regression and performance metrics to calculate its performance on the given dataset. |
| **Problem Statement:**  To solve classification problems using Logistic Regression. |
| **Background Study:**  Logistic regression is a classification technique which helps to predict the probability of an outcome that can only have two values. Logistic Regression is used when the dependent variable (target) is categorical. A logistic regression produces a logistic curve, which is limited to values between 0 and 1. |
| **Question Bank:**  **1. What is Logistic Regression?**  Logistic regression is **a statistical model that** in its basic form uses a logistic function to model a binary dependent variable, although many more complex extensions exist. In regression analysis, logistic regression (or logit regression) is estimating the parameters of a logistic model (a form of binary regression).  **2. How Logistic Regression is used for solving classification problems?**  The basis of logistic regression is the logistic function, also called the sigmoid function, which takes in any real valued number and maps it to a value between 0 and 1. Logistic regression model takes **a linear equation as input and use logistic function and log odds to perform** a binary classification task  **3. Why sigmoid function is used in it?**  The main reason why we use sigmoid function is **because it exists between (0 to 1)**. Therefore, it is especially used for models where we have to predict the probability as an output. Since probability of anything exists only between the range of 0 and 1, sigmoid is the right choice. The function is differentiable. |

**Student Work Area**

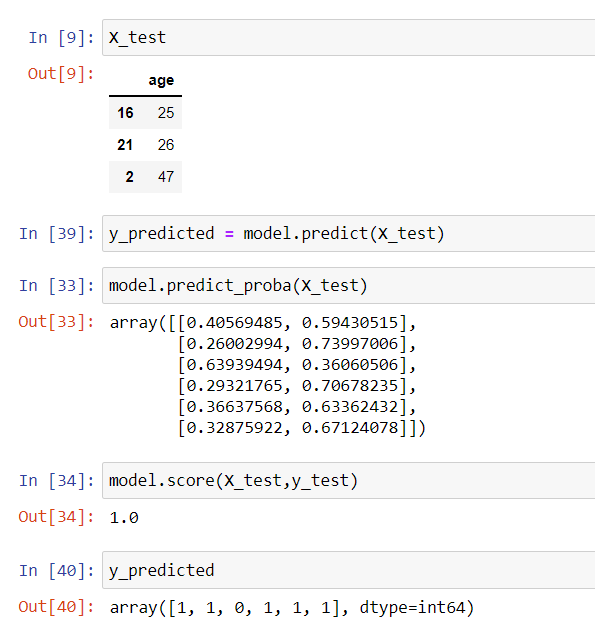
**Algorithm/Flowchart/Code/Sample Outputs**

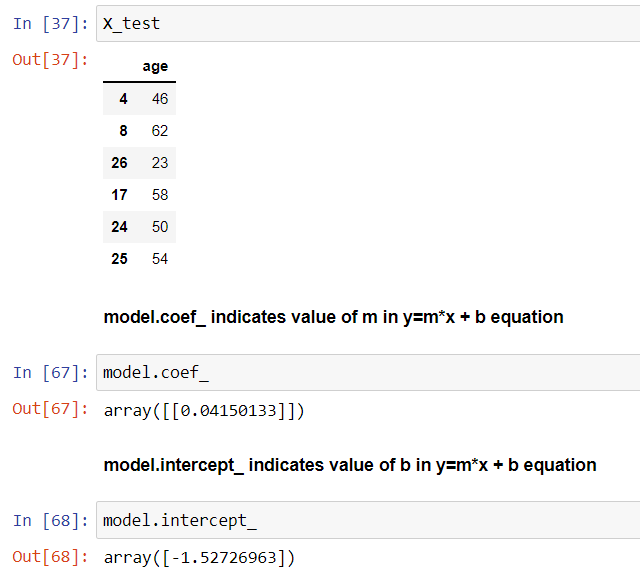
1. **Logistic Regression**

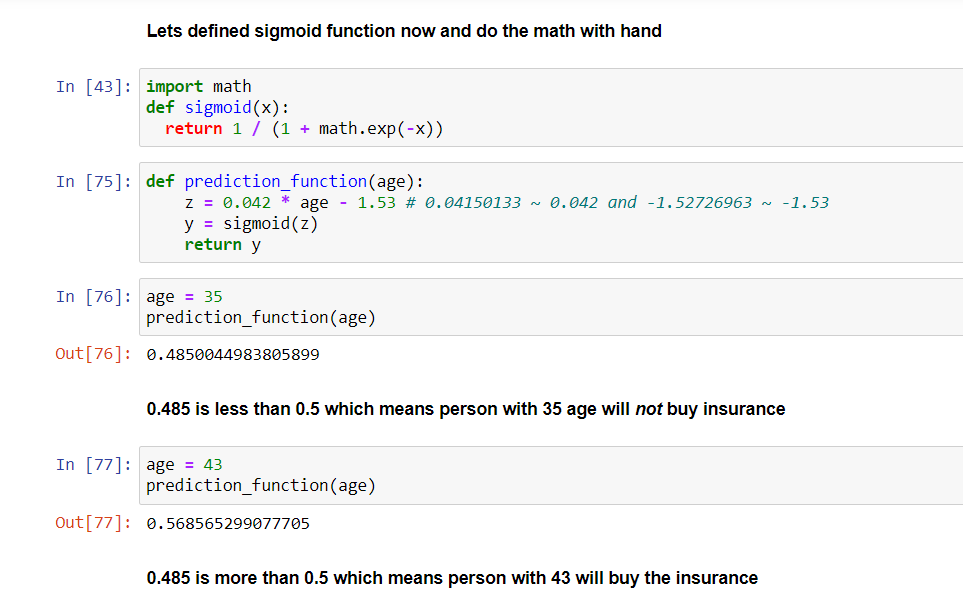
****

****

****

****

****

****

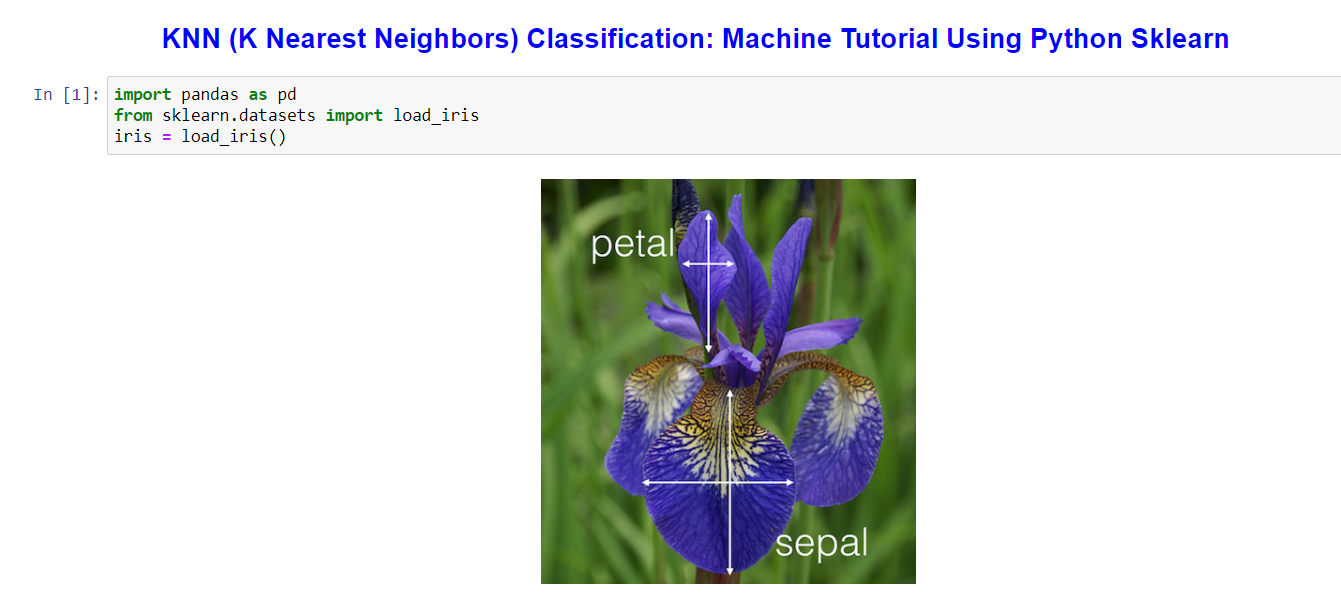
**EXPERIMENT NO. 10**

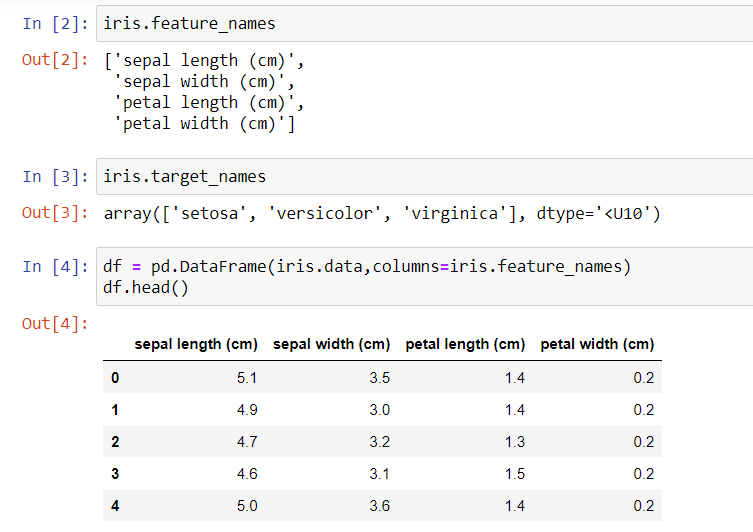
|  |
| --- |
| **Student Name and Roll Number: SUNIL KHARSU and 20csu371** |
| **Semester /Section: 5th, FSB** |
| **Link to Code:** |
| **Date: 28/10/21** |
| **Faculty Signature:** |
| **Grade:** |

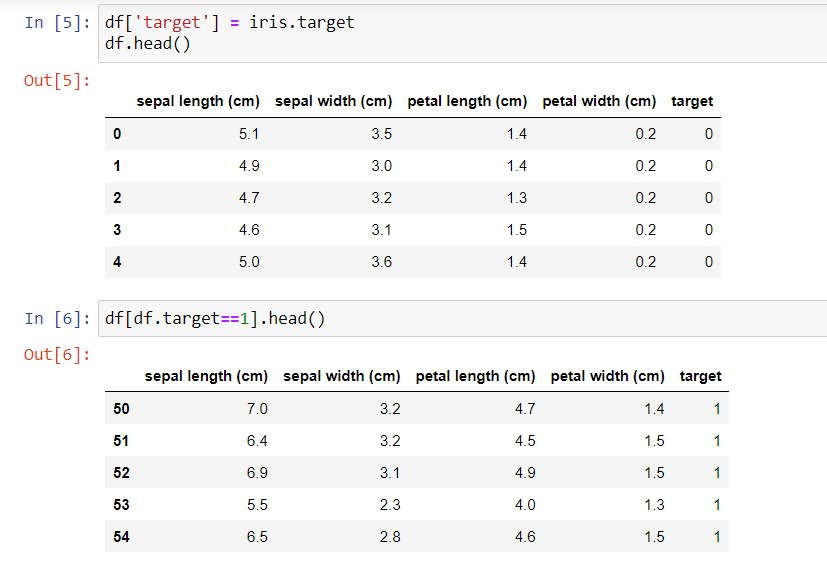
|  |
| --- |
| **Objective(s):**   * Study K-Nearest Neighbour algorithm (KNN). * Understand the working principle behind KNN. |
| **Outcome:**  Students will be familiarized with classification technique using KNN. |
| **Problem Statement:**  To solve classification problems using KNN classification. |
| Background Study:  K-nearest neighbors (KNN) algorithm is a type of supervised ML algorithm which can be used for both classification as well as regression predictive problems. However, it is mainly used for classification predictive problems in industry. The following two properties would define KNN well −   * Lazy learning algorithm − KNN is a lazy learning algorithm because it does not have a specialized training phase and uses all the data for training while classification. * Non-parametric learning algorithm − KNN is also a non-parametric learning algorithm because it doesn’t assume anything about the underlying data.   K-nearest neighbors (KNN) algorithm uses ‘feature similarity’ to predict the values of new datapoints which further means that the new data point will be assigned a value based on how closely it matches the points in the training set. The most common parameter used to perform matching is Euclidean distance between the points. |
| **Question Bank:**  **1. What is KNN classifier?**  K-Nearest Neighbors (KNN) is **one of the simplest algorithms used in Machine Learning for regression and classification problem** . KNN algorithms use data and classify new data points based on similarity measures (eg distance function). Classification is done by a majority vote to its neighbors.  **2. How KNN makes use of Euclidean distance to calculate nearest neighbor?**  The formula to calculate Euclidean distance is: **For each dimension, we subtract one point's value from the other's to get the length of that “side” of the triangle in that dimension, square it, and add it to our running total**. The square root of that running total is our Euclidean distance.  **3. What are the other distances that can be used for nearest neighbor?**  specifically, four different distance functions, which are **Euclidean distance, cosine similarity measure, Minkowsky, correlation, and Chi square**, are used in the k-NN classifier respectively  **4. What are the various performance metrics used for classification problems?**  There are many ways for measuring classification performance. **Accuracy, confusion matrix, log-loss, and AUC-ROC** are some of the most popular metrics. Precision-recall is a widely used metrics for classification problems. |

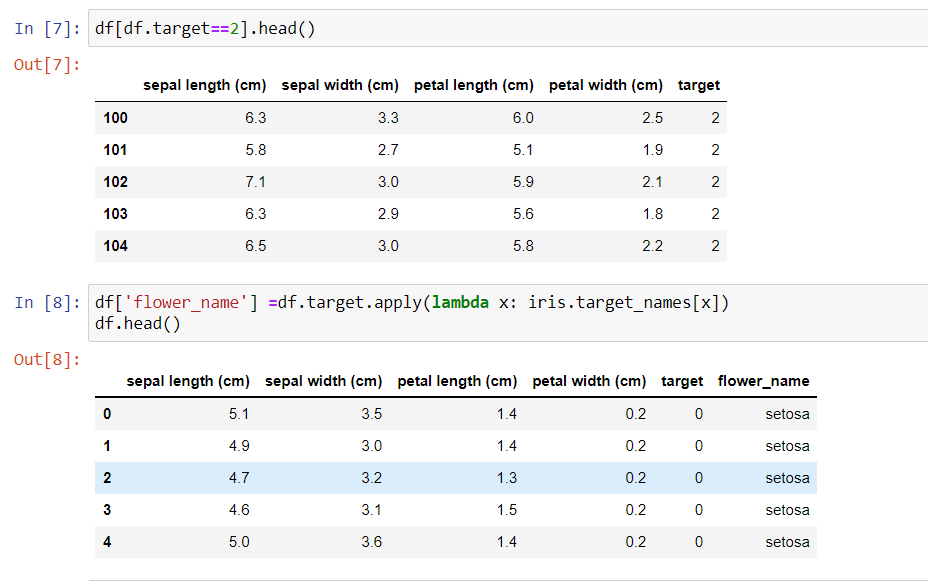
**Student Work Area**

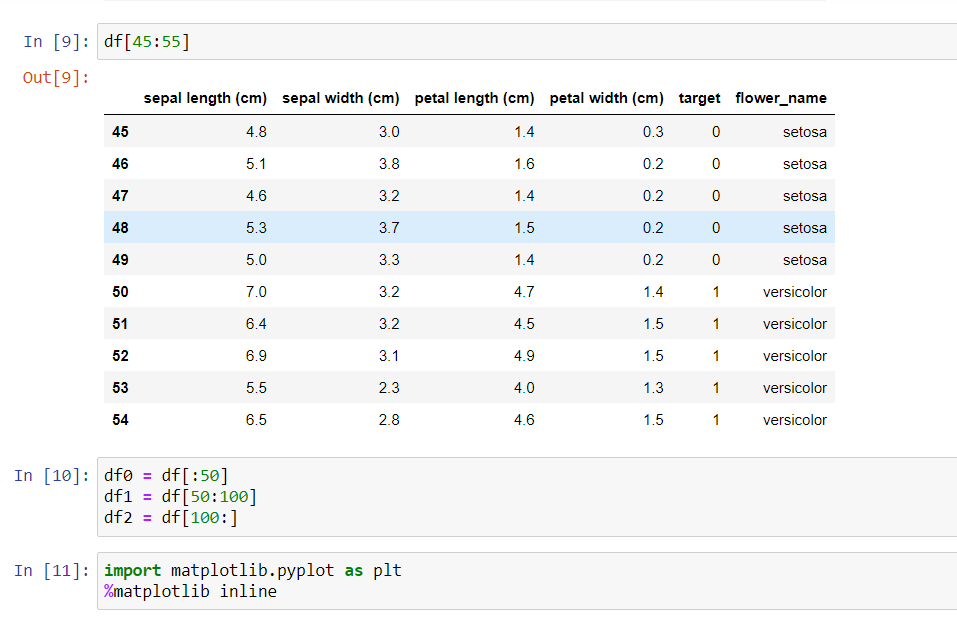
**Algorithm/Flowchart/Code/Sample Outputs**

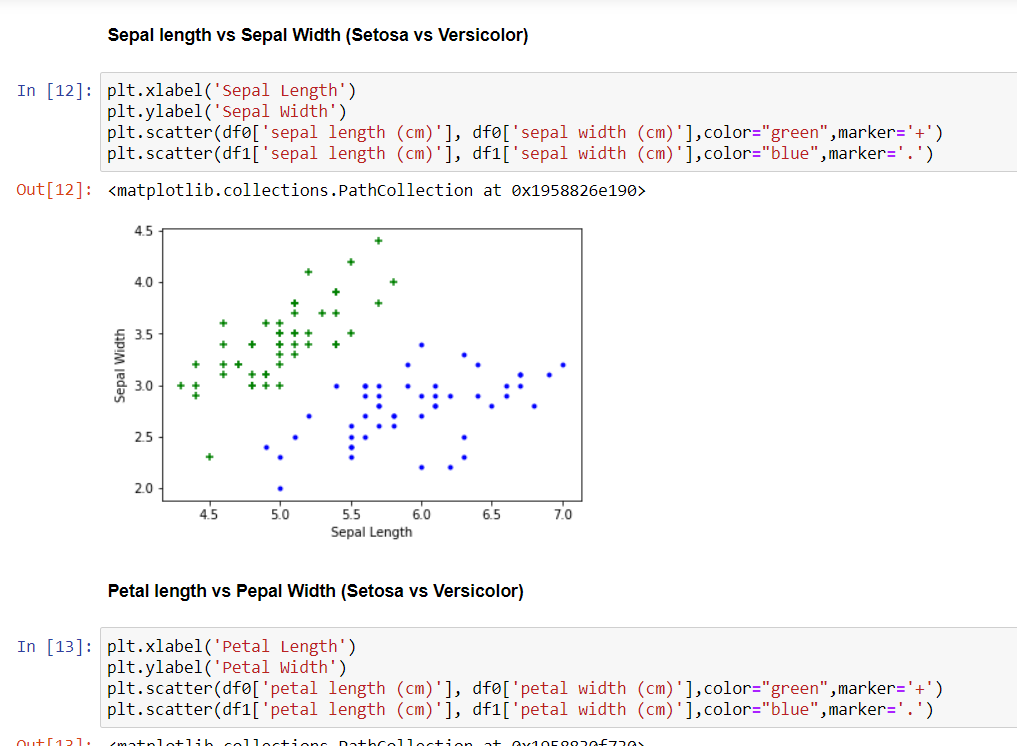
****

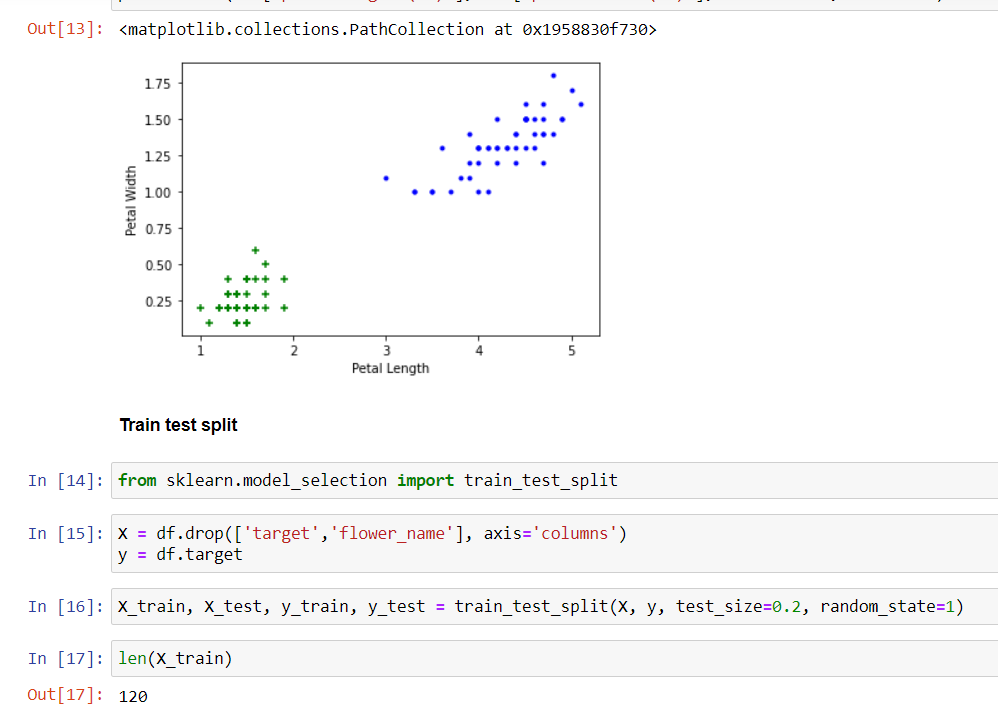
****

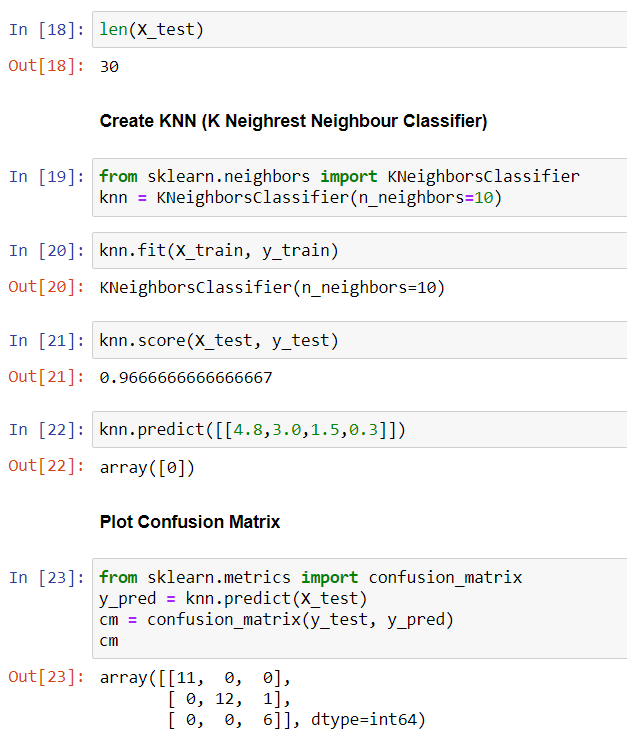
****

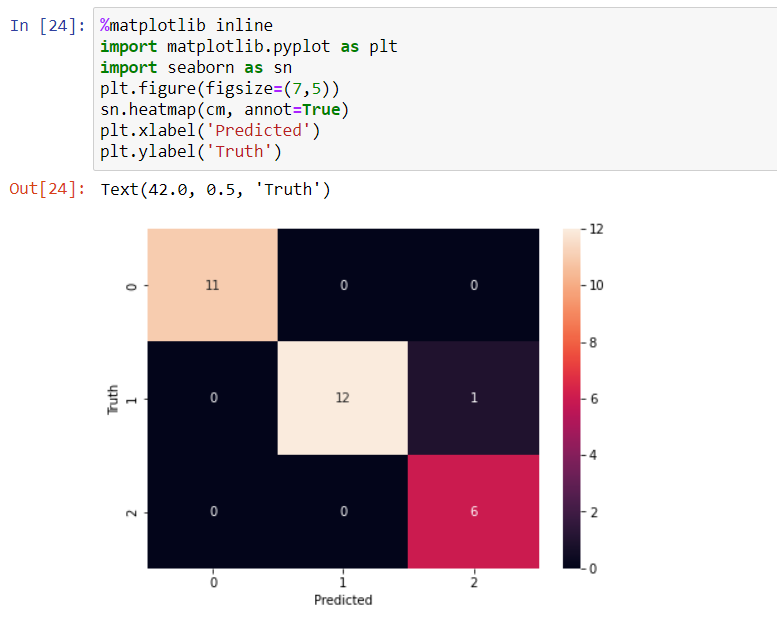
****

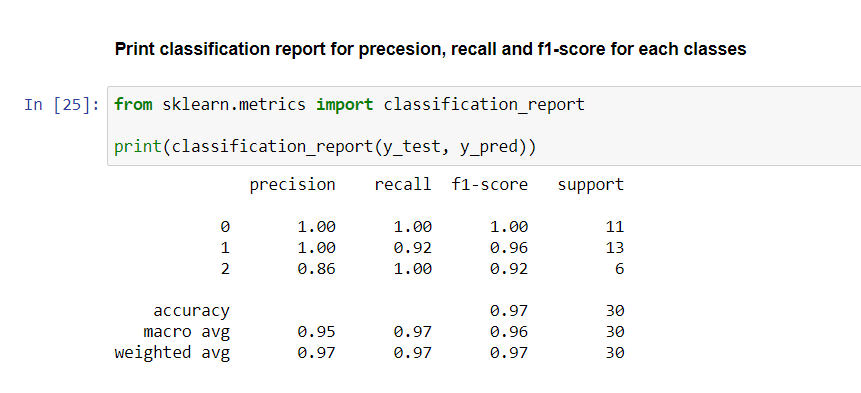
****

****

****

****

****

****

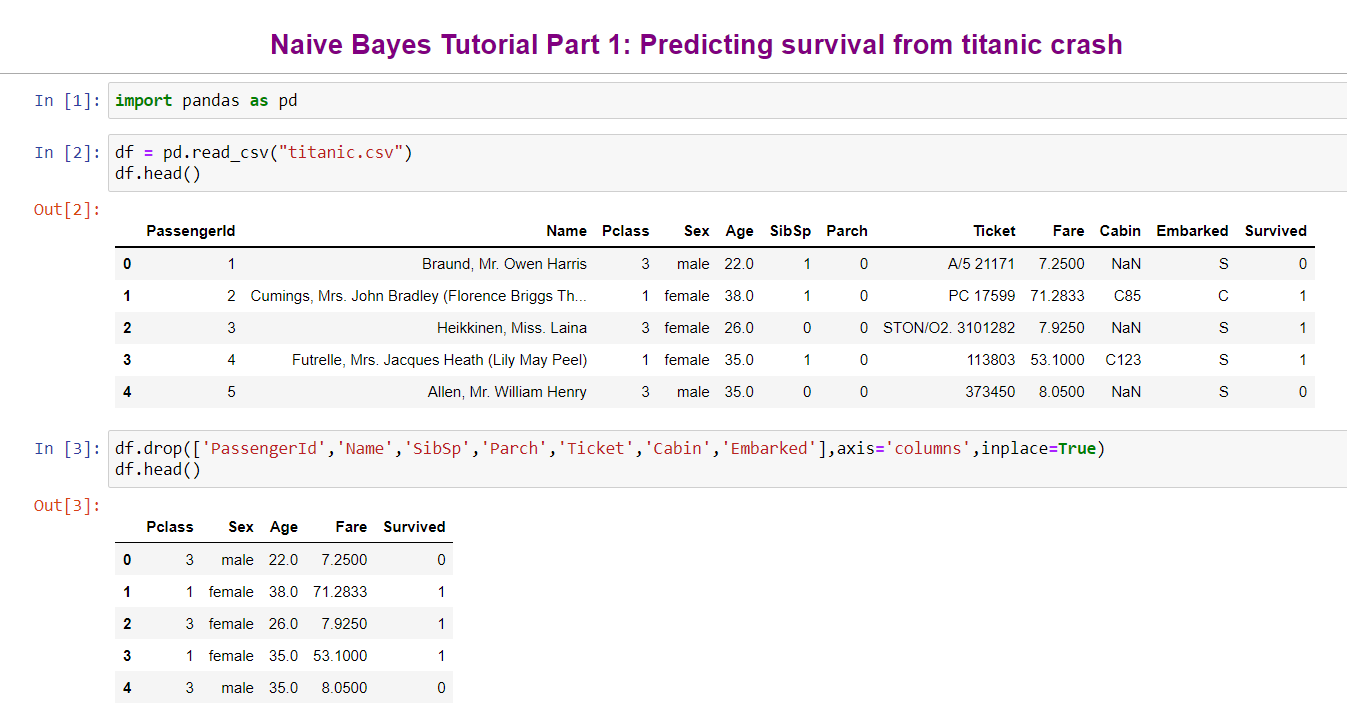
**EXPERIMENT NO. 11**

|  |
| --- |
| **Student Name and Roll Number: SUNIL KHARSU and 20csu371** |
| **Semester /Section: 5th, FSB** |
| **Link to Code:** |
| **Date: 11/11/21** |
| **Faculty Signature:** |
| **Grade:** |

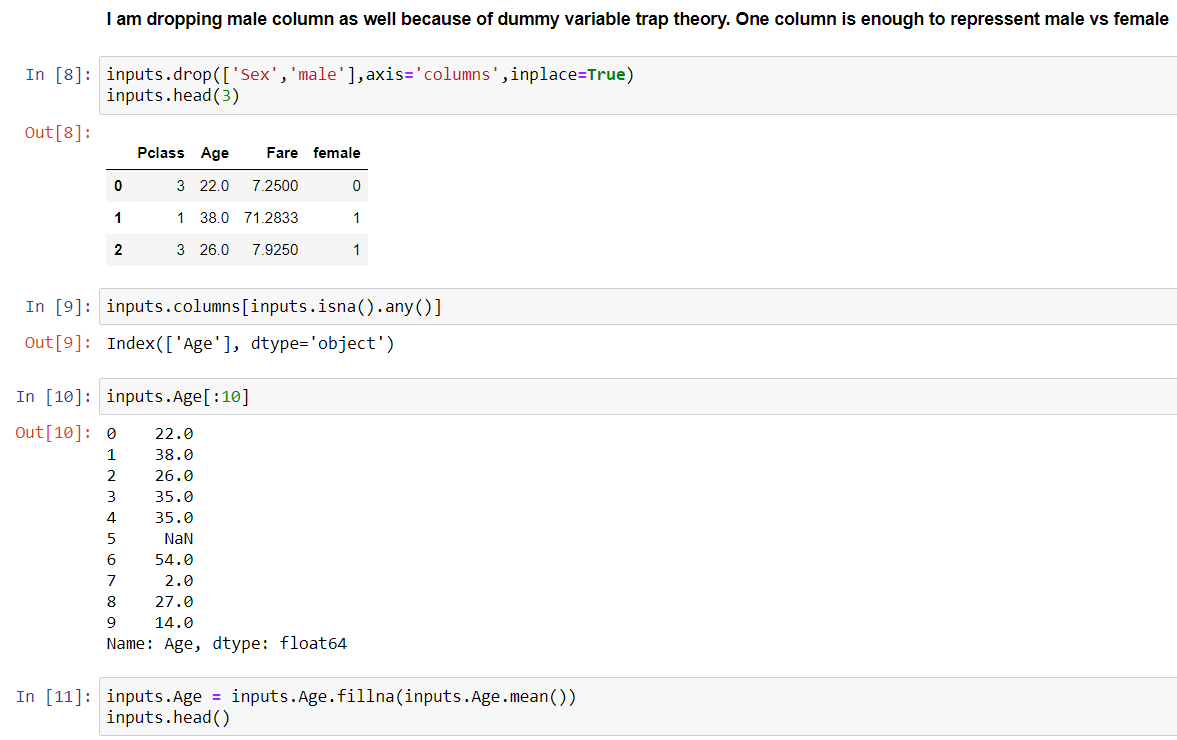
|  |
| --- |
| **Objective(s):**   * Understand and study Naïve Bayes (NB) Classifier. * Understand Naïve Bayes theorem behind it. |
| **Outcome:**  Students will be familiarized with NB classification technique. |
| **Problem Statement:**  To solve classification problems using Naïve Bayes. |
| **Background Study:**  Naïve Bayes Classifier is a probabilistic classifier and is based on Bayes Theorem.  In Machine learning, a classification problem represents the selection of the Best Hypothesis given the data.  Given a new data point, we try to classify which class label this new data instance belongs to. The prior knowledge about the past data helps us in classifying the new data point. The Naïve Bayes theorem:    gives us the probability of Event A to happen given that event B has occurred. |
| **Question Bank:**  **1. What is Bayes theorem?**  Bayes' theorem is **a way to figure out conditional probability** . ... For example, your probability of getting a parking space is connected to the time of day you park, where you park, and what conventions are going on at any time.  **2. How Naïve Bayes classifier helps for solving classification problems?**  Naïve Bayes Classifier is one of the simple and most effective classification algorithms which helps in building the fast machine learning models that can make quick predictions. It is a **probabilistic classifier** , which means it predicts on the basis of the probability of an object.  **3. What is the condition on features that should be fulfilled for successful application of Naïve Bayes method?**  The naive bayes classifers don't offer an intrinsic method to evaluate feature importances. Naïve Bayes methods **work by determining the conditional and unconditional probabilities associated with the features and predict** the class with the highest probability.   * If continuous features do not have normal distribution, we should use transformation or different methods to convert it in normal distribution. * If test data set has zero frequency issue, apply smoothing techniques “Laplace Correction” to predict the class of test data set. * Remove correlated features, as the highly correlated features are voted twice in the model and it can lead to over inflating importance. |

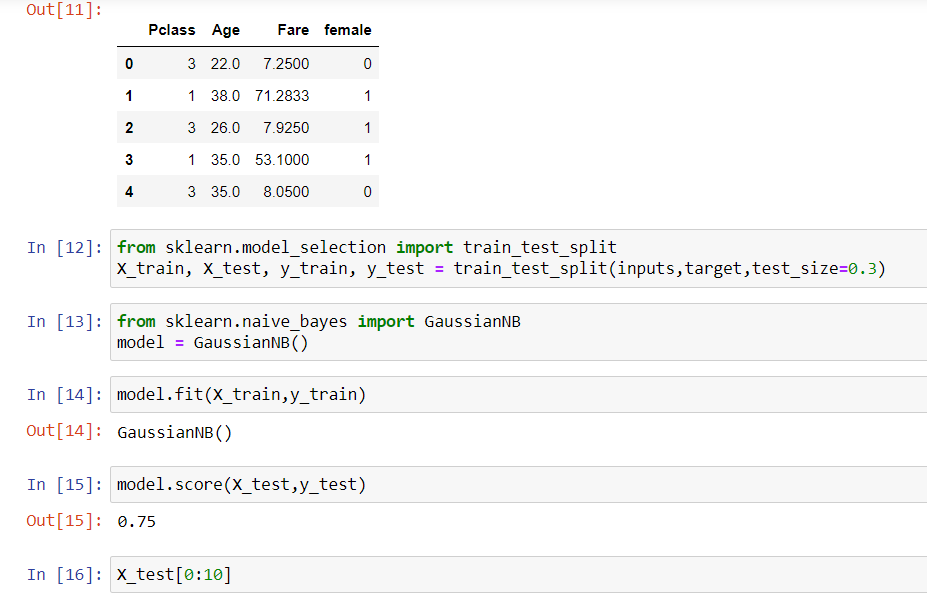
**Student Work Area**

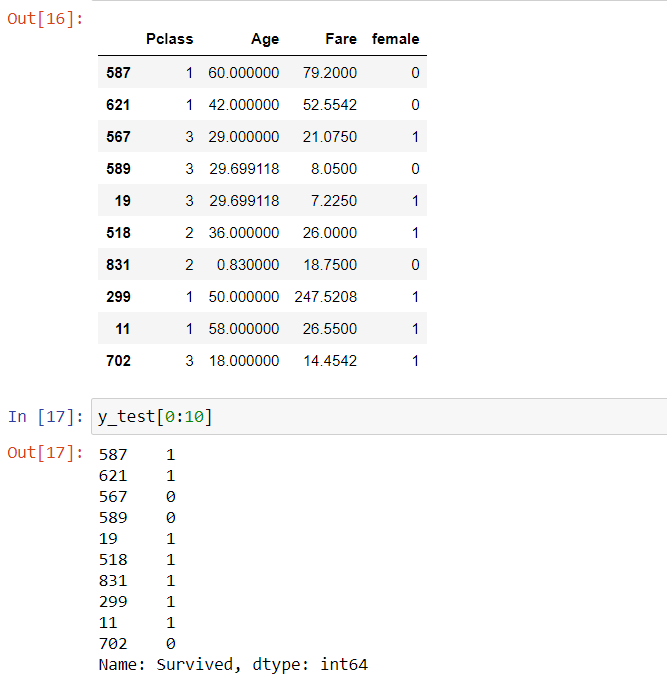
**Algorithm/Flowchart/Code/Sample Outputs**

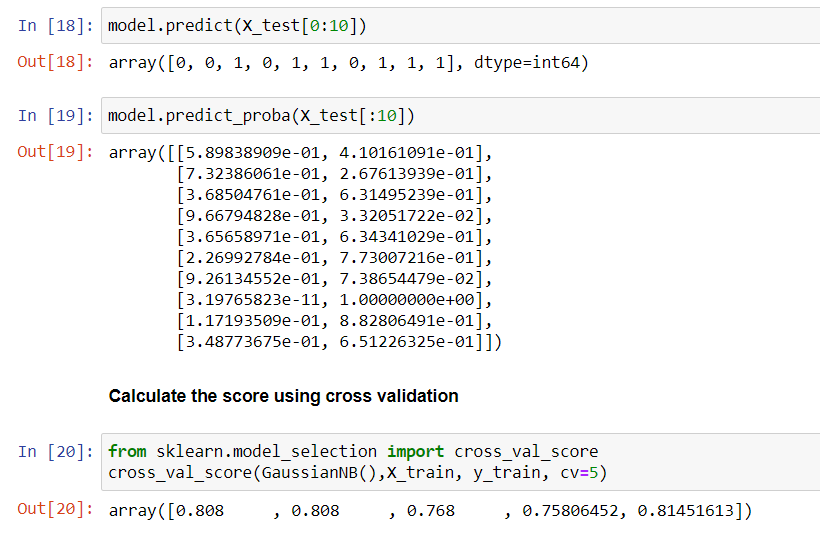












**EXPERIMENT NO. 12**

|  |
| --- |
| **Student Name and Roll Number: SUNIL KHARSU and 20csu371** |
| **Semester /Section: 5/ FSB** |
| **Link to Code:** |
| **Date:** |
| **Faculty Signature:** |
| **Grade:** |

|  |
| --- |
| **Objective(s):**   * Understand and study Support Vector Machines (SVM). * To study how linear hyperplane is calculated to differentiate between two classes. * Basic understanding of the different variants of SVM. |
| **Outcome:**  Students will be familiarized with Support Vector Machines classifier. |
| **Problem Statement:**  To solve classification problems using SVM. |
| **Background Study:**  In machine learning, support-vector machines (SVMs, also support-vector networks) are supervised learning models with associated learning algorithms that analyze data for classification and regression analysis.  Developed at AT&T Bell Laboratories by Vladimir Vapnik with colleagues, SVMs are one of the most robust prediction methods, being based on statistical learning frameworks or VC theory proposed by Vapnik (1982, 1995) and Chervonenkis (1974).  Given a set of training examples, each marked as belonging to one of two categories, an SVM training algorithm builds a model that assigns new examples to one category or the other, making it a non-probabilistic binary linear classifier. SVM maps training examples to points in space so as to maximise the width of the gap between the two categories. New examples are then mapped into that same space and predicted to belong to a category based on which side of the gap they fall. |
| **Question Bank:**  1. What is SVM?  2. What are the advantages of using SVM over other classifiers?  3. What do you mean by support vectors? |

**Student Work Area**

**Algorithm/Flowchart/Code/Sample Outputs**

**EXPERIMENT NO. 13**

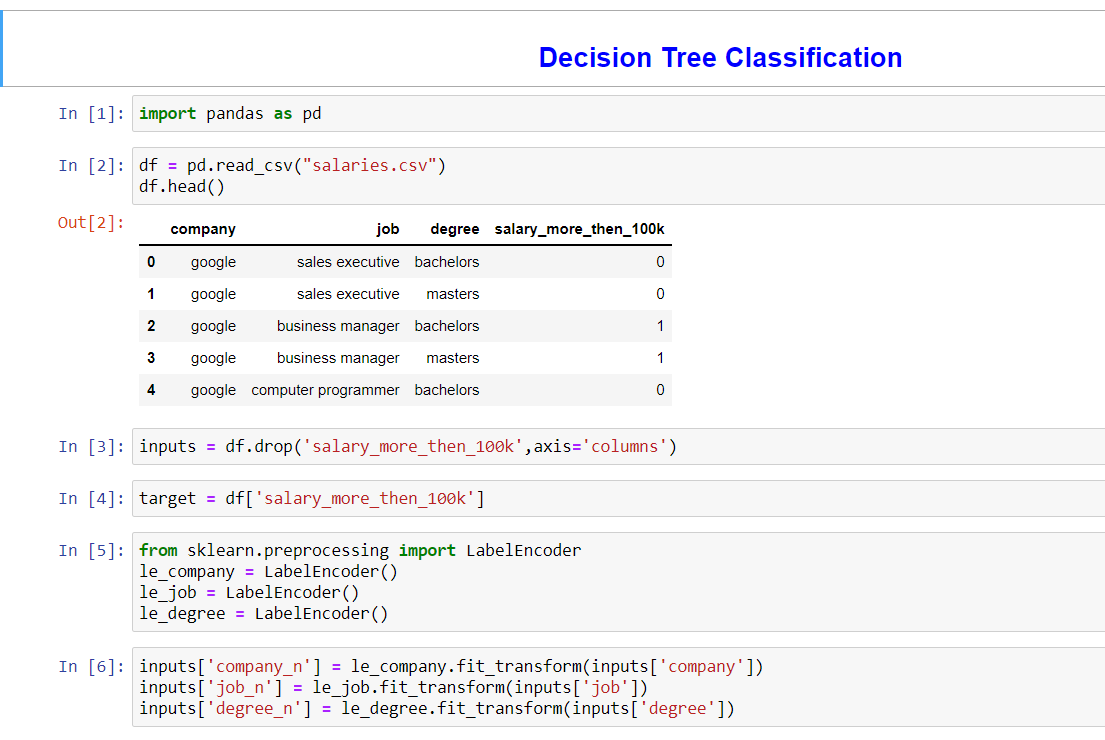
|  |
| --- |
| **Student Name and Roll Number: SUNIL KHARSU and 20csu371** |
| **Semester /Section: 5th, FSB** |
| **Link to Code:** |
| **Date: 18/11/21** |
| **Faculty Signature:** |
| **Grade:** |

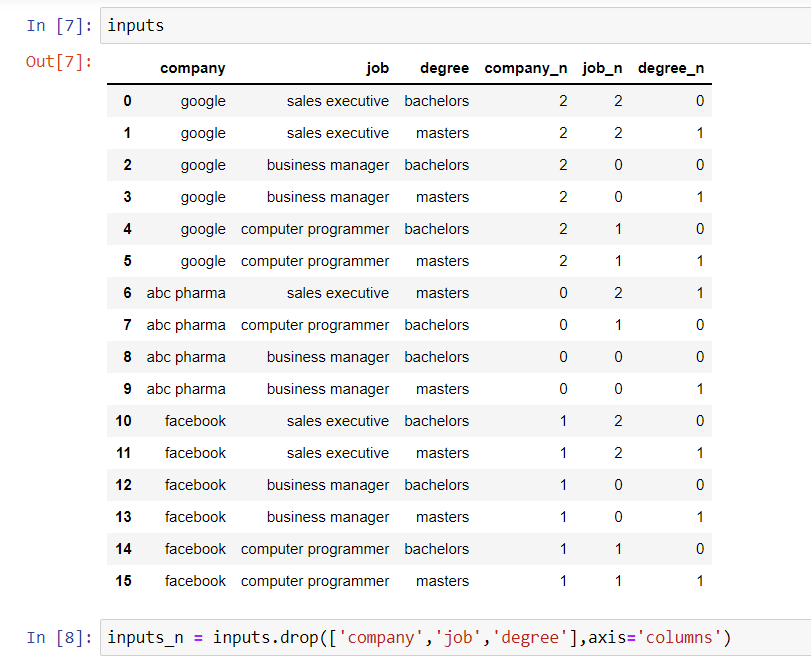
|  |
| --- |
| **Objective(s):**   * Understand and study Decision Trees for classification problems. * Study about the information gain used to create decision trees. |
| **Outcome:**  Students will be familiarized with creation of decision trees. |
| **Problem Statement:**  Apply Decision Tree classifier for solving classification problems. |
| Decision tree analysis is a predictive modelling tool that can be applied across many areas. Decision trees can be constructed by an algorithmic approach that can split the dataset in different ways based on different conditions. Decision trees are the most powerful algorithms that falls under the category of supervised algorithms.  They can be used for both classification and regression tasks. The two main entities of a tree are decision nodes, where the data is split and leaves, where we got outcome. |
| **Question Bank:**  **1. What is a decision tree?**  A decision tree is **a decision support tool that uses a tree-like model of decisions and their possible consequences**, including chance event outcomes, resource costs, and utility. It is one way to display an algorithm that only contains conditional control statements.  **2. How decision tree is created to solve problems?**  Decision trees help you **to evaluate your options**. Decision Trees are excellent tools for helping you to choose between several courses of action. They provide a highly effective structure within which you can lay out options and investigate the possible outcomes of choosing those options.  **3. List out the advantages and disadvantages of Decision Tree Classifiers?**  Decision tree often involves higher time to train the model. Decision tree training is relatively expensive as the complexity and time has taken are more. The Decision Tree algorithm **is inadequate for applying regression and predicting continuous values**. |

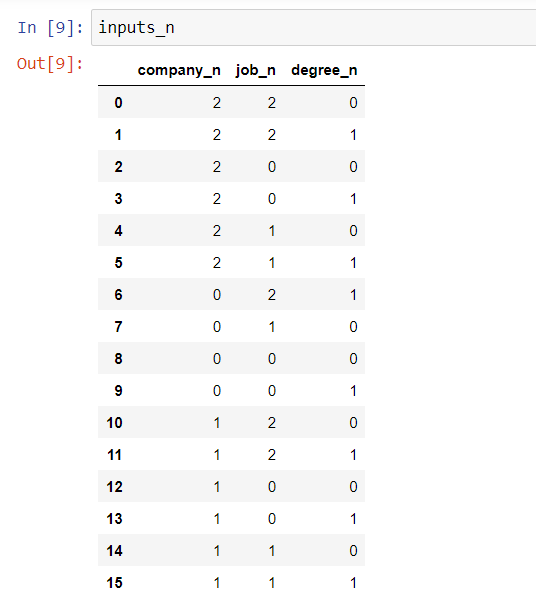
**Student Work Area**

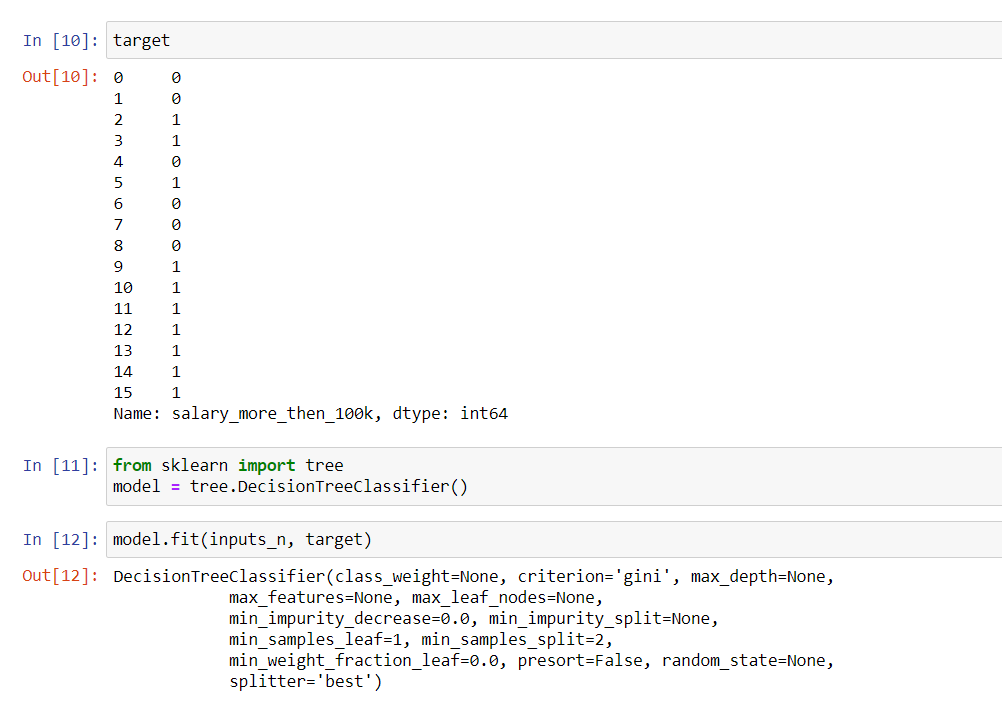
**Algorithm/Flowchart/Code/Sample Outputs**

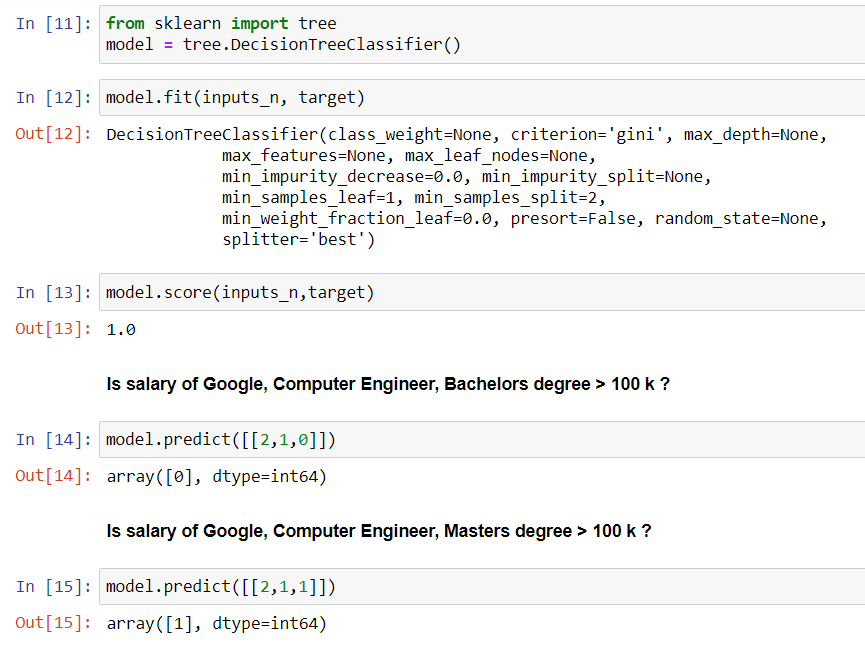
1. **Decision Tree**

****

****

****

****

****

**Submitted By: Sunil Kharsu(20csu371)**

**Submitted To: Dr. Nitin Malik**