SHRI G.S. INSTITUE OF TECHNOLOGY & SCIENCE
INDORE (M.P.)

Student Mark Prediction

A project report submitted to
Rajiv Gandhi Proudyogiki Vishwavidhyalaya,
Bhopal. towards partial fulfillment of
the degree of

MASTER OF COMPUTER APPLICATION

2022-2024

SUBMITTED BY:
BHARAT KUMAR (0801CA221013)

GUIDED BY:
Mr. Upendra Singh

SHRI G.S. INSTITUTE OF TECHNOLOGY AND SCIENCE
INDORE (M.P.)

DECLARATION

We, **Bharat Kumar** declare that this project report titled, “**Student Mark Prediction”** and the work presented in it are our own. We confirm that:

- This work was done wholly while in candidature for a master degree at this University.
- Where any part of this project has previously not been submitted for a degree or any other qualification at this University or any other institution.
- Where we have consulted the published work of others, this is always clearly at tributed.
- Where we have quoted from the work of others, the source is always given. With the exception of such quotations, this project is entirely our own work.
- We have acknowledged all main sources of help.

Signed:

Date:

SHRI G.S. INSTITUTE OF TECHNOLOGY AND SCIENCE
INDORE (M.P.)

RECOMMENDATION

The project report entitled “**Student Mark Prediction**” submitted by **Bharat Kumar** students of MCA final year in the session 2023-24, towards the fulfillment of the degree of **Master of Computer Applications** of Rajiv Gandhi Proudyogiki Vishwavidhyalaya, Bhopal is a satisfactory account of their work and is recommended for the award of degree.

Mr. Upendra Singh
Project Guide
Department of Information
Technology

Dr. K. K. Sharma
Head of Department
Department of Information
Technology

SHRI G.S. INSTITUTE OF TECHNOLOGY AND SCIENCE
INDORE (M.P.)

CERTIFICATE

The project report entitled "**Student Mark Prediction**" submitted by **Bharat Kumar** students of MCA final year in the session 2023-24, towards partial fulfillment of the degree of **Master of Computer Applications** of Rajiv Gandhi Proudyogiki Vishwavidhyalaya, Bhopal, is a satisfactory account of their work and is approved for the award of the degree.

Internal Examiner **External Examiner**

Date: Date:

SHRI G.S. INSTITUTE OF TECHNOLOGY AND SCIENCE INDORE (M.P.)

ACKNOWLEDGEMENT

*Every work accomplished is a pleasure and a sense of satisfaction, however a number of people always motivate, criticize and appreciate a work with their objective ideas and opinions, hence we are heartily pleased to acknowledge all those people who have helped us in the successful completion of this project. With great pleasure, we express our heartfelt gratitude to our esteemed guide, **Mr. Upendra Singh**, Assistant Professor, Department of Computer Technology & Application, S.G.S.I.T.S. Indore. Their persistent encouragement, perpetual motivation, everlasting patience, and valuable technical inputs in discussions have enabled the successful completion of this project. Their invaluable help, advice, and constant encouragement helped us a lot and provided the impetus to the progress of the project. We extend our profound indebtedness to the Head of the department, **Dr. K.K. Sharma**, the words lose their worth for his invaluable guidance, continuous encouragement, and cooperation in every respect.*

We sincerely wish to express our gratitude to all the members of staff of M.C.A. who have extended their cooperation at all times and have contributed in their own way to developing the project. Successful completion of a project is not an individual effort. It is an outcome of the cumulative effort of a number of persons, each having his own importance to the objective. We are thankful to our parents for being a constant source of encouragement in all our endeavors. Indeed, it is their support that helps us through the ups and downs of life. The support and suggestions of our friends are worth appreciation and thankfulness. A blend of gratitude, pleasure, great satisfaction and indebtedness is what we feel to convey to all those who have directly or indirectly contributed to the successful completion of our project work.

Bharat Kumar (0801CA221013)

SHRI G.S. INSTITUTE OF TECHNOLOGY AND SCIENCE
INDORE (M.P.)

ABSTRACT

In the realm of education, predicting students' academic performance is of paramount importance for educators, administrators, and policymakers. This study focuses on developing a predictive model for student marks based on their study hours and previous academic performance. The goal is to provide a tool that can assist educators in identifying students at risk of underperformance early on, enabling timely interventions and personalized support.

The dataset used for this study includes information on students' study hours and their previous marks. Machine learning techniques, specifically regression analysis, are employed to establish a predictive model. Features such as study hours and previous marks are utilized to train the model, with the aim of capturing the underlying patterns and relationships that influence students' current academic achievements.

The model's performance is evaluated using various metrics, including mean squared error and R-squared values, to assess its accuracy and reliability. Additionally, feature importance analysis is conducted to identify the relative significance of study hours and previous marks in predicting the current academic performance.

Contents

CHAPTER-1: INTRODUCTION

Education is a fundamental aspect of personal and societal growth, and with the advent of technology, there is a growing interest in leveraging machine learning techniques to enhance educational processes. One such application is the prediction of student marks based on their study hours. This project aims to develop a predictive model that can forecast a student's performance given the number of hours they dedicate to studying.

1.1 Basic:

Predicting student marks based on their study hours is a classic machine learning problem that falls under the category of regression. In this scenario, you're trying to predict a continuous numerical output (marks) based on one or more input features (study hours). Here's a basic outline of how you can approach this project:

1. Data Collection:

Gather a dataset that includes information on study hours and corresponding marks. This dataset should ideally be diverse and representative of the population you want to make predictions for.

2. Data Exploration and Analysis:

Explore the dataset to understand its structure and characteristics.

Visualize the relationship between study hours and marks using scatter plots or other appropriate visualizations.

3. Data Preprocessing:

Handle missing values if any.

Split the dataset into training and testing sets to evaluate the model's performance.

4. Selecting a Model:

Choose a regression model suitable for your dataset. Linear regression is a good starting point for a simple relationship between study hours and marks.

5. Feature Engineering:

If needed, create new features or transform existing ones to improve the model's performance.

6. Model Training:

Train your chosen model using the training dataset.

7. Model Evaluation:

Evaluate the model's performance using the testing dataset.

Metrics such as Mean Squared Error (MSE) or R-squared can be used to measure how well your model is performing.

8. Prediction:

Once the model is trained and evaluated, you can use it to make predictions on new data.

9. Visualization:

Visualize the predicted marks against the actual marks to see how well your model is performing.

1.2 Aim:

The aim of a student mark prediction machine learning (ML) project based on their study hours and marks is to develop a model that can predict a student's future performance (marks) based on the number of hours they study. This type of project falls under the broader category of regression tasks in machine learning.

1.3 Contribution:

A student mark prediction machine learning (ML) project based on study hours and marks can have several contributions and benefits:

1. **Academic Performance Prediction:**
   - The primary contribution is predicting students' future academic performance based on their study hours. This can help identify students who might be at risk of underperforming or those who are likely to excel.
2. **Early Intervention:**
   - Early identification of students who may struggle academically allows for timely intervention. Teachers and educational institutions can provide additional support or resources to help these students improve their performance.
3. **Personalized Learning:**
   - The model can help tailor educational approaches for individual students. By understanding the relationship between study hours and performance, educators can provide personalized recommendations to optimize learning strategies.
4. **Resource Allocation:**
   - Educational institutions can use the predictions to allocate resources more efficiently. This includes assigning additional support staff or allocating funds for specific educational programs to address the needs of students predicted to be at risk.
5. **Student Motivation:**
   - Knowing that their study efforts are being considered in predicting future academic success can motivate students to invest more time in their studies, leading to improved overall performance.

6. **Parental Engagement:**
    - Parents can be informed about their child's predicted academic performance. This can encourage parental involvement in a student's education, fostering a collaborative approach between parents, students, and educators.
7. **Curriculum Development:**
    - Institutions can use insights from the model to refine and improve the curriculum. Understanding which study habits contribute to better performance can guide the development of effective teaching strategies.
8. **Data-Driven Decision Making:**
    - The project promotes a data-driven approach to education. Educators and administrators can make decisions based on empirical evidence rather than intuition, leading to more informed and effective educational practices.
9. **Continuous Improvement:**
    - The model can be regularly updated with new data to adapt to changes in educational patterns and student behavior. This ensures that the predictions remain accurate and relevant over time.
10. **Research Opportunities:**
    - The data collected and analyzed during the project can provide valuable insights for educational researchers. It opens up opportunities for further studies on the correlation between study habits and academic success.

CHAPTER-2:

LITERATURE SURVEY

The main purpose of literature survey is to find out new techniques to work on the old data set and then find out some new information form that. To do some relational survey, the literature of more than 10 years should be taken into consideration and then find out some knowledge gaps between works done by the researcher. It helps to justify your research questions and gave some direction for future research

2.1 Related Work

I have alluded to several research articles that are related to the thesis in order to specify the thesis as a well-structured thought. Conclusion information of few of the papers are as follows. This research study describes how the linear regression approach issued in predicting student’s academic performance.

Student Marks prediction is one of the essential research topics in education. Several other authors have worked on this topic and found different insights: B. k. Bhardwaj and S. Pal [1] did a study on predicting the student's performance by choosing over 300 students from six-degree colleges conducting BCA (Bachelor of Computer Application) course in Dr. R. M. L. Awadh University, Faizabad, in India. Using the Bayesian classification technique on seventeen attributes, they showed that students' academic performance corresponds to both the academic and non-academic attributes like family annual income and student's family overall status, etc.

1. In this studies paper creator carried out the thesis the usage of SVM technique in java, selection tree, C4.5, Naive Bayes, Lib. SVM, Logistic Regression and Hybrid technique LMT and as compared the accuracy of overall performance prediction most of the hybrid approaches. I have alluded to several research articles that are related to the thesis in order to specify the thesis as a well-structured thought.

2. In this studies paper it's far discovered that the writer used a number of the maximum famous algorithms and regression algorithms. The experiment was conducted with administrative data from the University of Polo, which included 700 courses. The article concludes that decision Trees and SVM produce the best results. The main contribution of this work is to compare the levels of accuracy of several algorithms.

3. The studies is targeted on predicting student's overall performance the usage of personalised analytics. This paper presents two different approaches to work on the thesis. The author's initial technique is the Regression Algorithm, which is a data mining function. The root mean square method is also used to calculate the regression algorithm's error rate. In this paper the author worked on how to improve the prediction algorithms which are used to analyze and predict the student's performance. The decision trees algorithm is used in this paper's work.

4. This paper proposed the student Academic performance prediction using Support Vector Machine. The author compared SVM to various machine learning approaches such as linear regression, Decision Trees, and KNN and determined that SVM outperformed them.

Student grade prediction is one of the essential research topics in education. Several other authors have worked on this topic and found different insights: B. k. Bhardwaj and S. Pal [1] did a study on predicting the student's performance by choosing over 300 students from six-degree colleges conducting BCA (Bachelor of Computer Application) course in Dr. R. M. L.

Awadh University, Faizabad, in India. Using the Bayesian classification technique on seventeen attributes, they showed that students' academic performance corresponds to both the academic and non-academic attributes like family annual income and student's family overall status, etc.

Authors in [2] proposed a model on Prediction of Students Performance using Machine learning in which they used previously obtained marks by the students of class 10th, 12th, and their semester marks. The scope of this paper was to predict the result and find out how many students got marks below 50% in 10th and 12th, students who failed in the internal exam, and the students had less attendance percentage.

S. Huang and N. Fang[3] examined various mathematical and machine learning techniques and applied four different mathematical modeling techniques: multivariate linear regression, multilayer perceptron neural networks, radial basis function neural networks, and support vector machines to predict student performance. The dataset contains 1,938 data records that were collected from 323 undergraduates in four semesters. This study concludes that there is no such difference between these methods. Their model was able to get more than 80% of accuracy.

J. Gamulin, O. Gamulin, and D. Kermek[4] collected the student data which is generated through Learning Management System(LMS), web-based formative and summative assessments during the traditional teaching in the classroom. The dataset contains a huge amount of data on students' behavior and grade/percentage at the point of time when the course is still in progress. Considering this dataset and by applying various classification algorithms and genetic algorithms, the author proposed a model for predicting the performance of students in the final examination.

CHAPTER-3:

Methodology

For a student mark prediction ML project based on study hours and marks, you can experiment with various regression algorithms. The principal step within side the implementation is to gather the statistics set required for the studies work. The technique is carried out to the dataset containing the statistics of the students. To simplify our analysis, we can identify the data set's unique attributes and delete them because they can't be used for analysis. The data is collected and then translated into the desired format.

This process is called as pre-processing of data. It is the most crucial stage in obtaining the specific necessary info from the raw data. The higher the accuracy rate, the better. The more raw data is preprocessed, the higher the rate accuracy of acceptable data. After pre-processing the data, the following step is to detect and eliminate any incomplete or irrelevant data from the dataset in order to achieve correct findings. Data cleaning is the process of removing unnecessary data. For improved classification, we can use any of the techniques available, such as linear regression, support vector machine, Naive Bayes Standard Classification, and decision tree algorithms. In this research, the linear regression algorithm is used to implement the solution. We must also select a training set from the dataset, determine the Result attributes that determine the output, and begin classification.

Here are some commonly used algorithms for such tasks:

3.1 Method – 1 : Linear Regression:

3.1.1 Algorithm – 1

Linear Regression assumes a linear relationship between the independent variable (study hours) and the dependent variable (marks). The model tries to find the best-fit straight line that minimizes the sum of the squared differences between predicted and actual values.

The equation for a simple linear regression (with one independent variable) is:

Marks = $\beta 0+\beta 1\times$Study Hours

Here,

$\beta 0$ is the intercept and $\beta 1$ is the slope of the line.

3.1 Method – 2 : KNN:

3.1.1 Algorithm – 2

KNN is a non-parametric and lazy learning algorithm used for both classification and regression.

In the context of regression, KNN predicts the output for a new data point based on the average (or weighted average) of the output values of its k-nearest neighbors.

Given a new data point (with study hours), KNN identifies the k training data points (students) that are closest to it in terms of study hours.

"Closeness" is typically determined by Euclidean distance, but other distance metrics can also be used.

The predicted mark for the new student is the average (or weighted average) of the marks of its k-nearest neighbors.

CHAPTER-4:

Implementation:

4.1 Hardware:

- Processor (CPU): Multi-core processor (e.g., Intel Core i5 or equivalent).
- Memory (RAM): 8 GB or more.
- Storage: Standard Hard Disk Drive (HDD) or Solid State Drive (SSD) with sufficient space.
- GPU (Optional): Not strictly necessary for basic tasks but can be beneficial for larger datasets or deep learning.

4.2 Software:

- Operating system: Windows XP/7/10
- Coding Language: python
- Development environment: Google Colab
- Dataset: students mark the dataset
- IDE : Colab notebook

4.3 Python Library in Details:

Python libraries which are used in the project are as follows:

1. google.colab.drive (Google Colab library for mounting Google Drive)

The code mounts Google Drive, checks file paths, loads, cleans, and visualizes data. It then trains linear regression and decision tree models, making predictions and comparing results through plots.

Example:

```
from google.colab import drive
drive.mount('/content/drive')
```

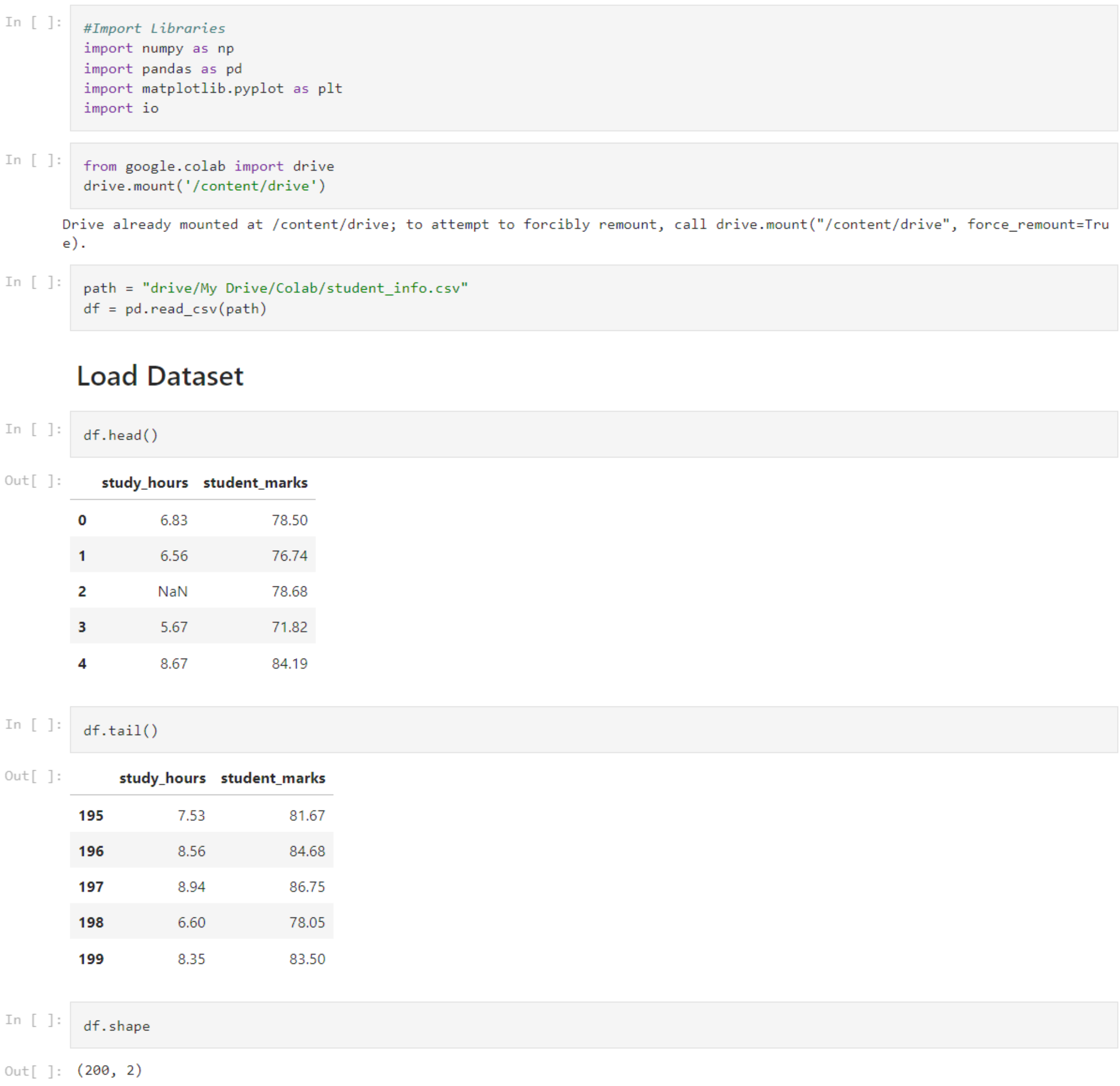
2. Pandas: pandas library is used to data manipulation and data analysis with panel data same as numpy it also provides a multi dimensional data structure, pandas provide two of data structure single and multi dimension where series is single and data frame is the multi dimensional data structure.by this library we can read csv file where is a data set for the model. It has head(),tail, shape, describe, loc, and iloc method which helps in data analysis.

3. Matplotlib: Machine learning projects have data set with a large amount of data. So every time programmer can not analyze the data so in python by matplotlib library programmer can plot a lot of charts like bar plot, scatter plot, histogram, etc.Matplotlib is also known as a data visualization library.

4. Numpy: numpy is a python library used for working with arrays. It also has function foe working in domain of linear algebra, Fourier transform, and matrices.

5. Joblib: joblib is a popular Python library used for efficiently handling the parallel execution of tasks, especially for computationally intensive tasks like machine learning model training. It is often used in conjunction with libraries like scikit-learn for parallelizing tasks such as model training and evaluation.

4.4 Screenshot with description:



In [ ]:

```
#Import Libraries
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import io
```

In [ ]:

```
from google.colab import drive
drive.mount('/content/drive')
```

```
Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", force_remount=Tru
e).
```

In [ ]:

```
path = "drive/My Drive/Colab/student_info.csv"
df = pd.read_csv(path)
```

Load Dataset

In [ ]:

```
df.head()
```

Out[ ]:

	study_hours	student_marks
0	6.83	78.50
1	6.56	76.74
2	NaN	78.68
3	5.67	71.82
4	8.67	84.19

In [ ]:

```
df.tail()
```

Out[ ]:

	study_hours	student_marks
195	7.53	81.67
196	8.56	84.68
197	8.94	86.75
198	6.60	78.05
199	8.35	83.50

In [ ]:

```
df.shape
```

Out[ ]:

```
(200, 2)
```

Discover and Visualize the data to gain insights

In [ ]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 200 entries, 0 to 199
Data columns (total 2 columns):
 #   Column         Non-Null Count  Dtype  
---  ------         --------------  -----  
 0   study_hours    195 non-null    float64
 1   student_marks  200 non-null    float64
dtypes: float64(2)
memory usage: 3.2 KB
```

In [ ]:

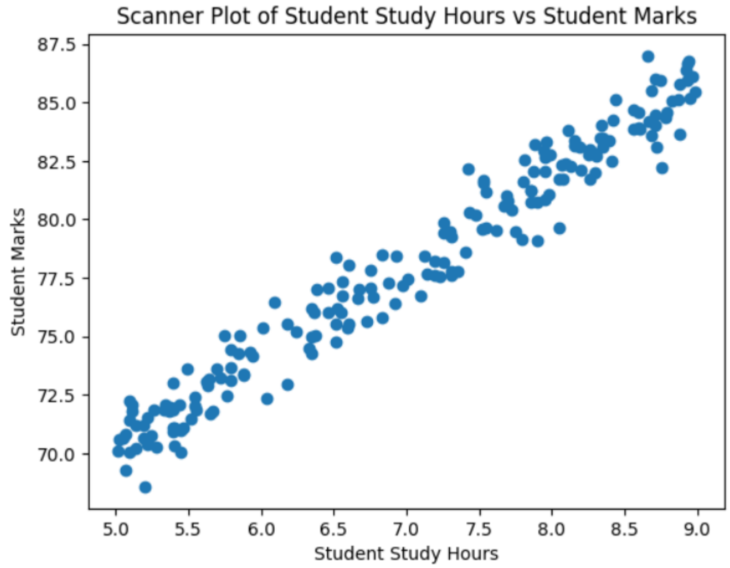
```
df.describe()
```

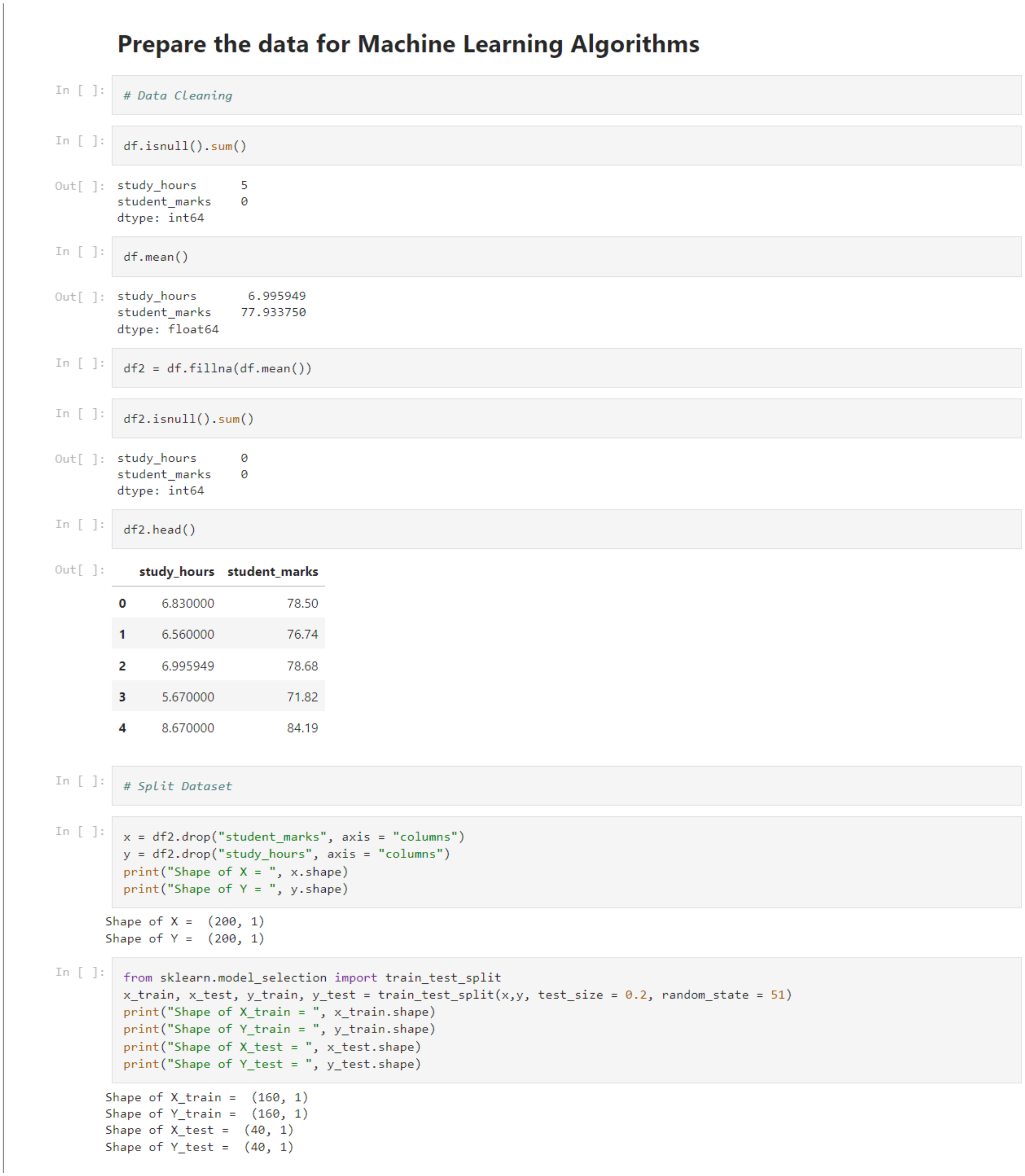
Out[ ]:

	study_hours	student_marks
count	195.000000	200.00000
mean	6.995949	77.93375
std	1.253060	4.92570
min	5.010000	68.57000
25%	5.775000	73.38500
50%	7.120000	77.71000
75%	8.085000	82.32000
max	8.990000	86.99000

In [ ]:

```
plt.scatter(x = df.study_hours, y = df.student_marks)
plt.xlabel("Student Study Hours")
plt.ylabel("Student Marks")
plt.title("Scanner Plot of Student Study Hours vs Student Marks")
plt.show()
```





Prepare the data for Machine Learning Algorithms

In [ ]:

```
# Data Cleaning
```

In [ ]:

```
df.isnull().sum()
```

Out[ ]:

```
study_hours      5
student_marks    0
dtype: int64
```

In [ ]:

```
df.mean()
```

Out[ ]:

```
study_hours       6.995949
student_marks    77.933750
dtype: float64
```

In [ ]:

```
df2 = df.fillna(df.mean())
```

In [ ]:

```
df2.isnull().sum()
```

Out[ ]:

```
study_hours      0
student_marks    0
dtype: int64
```

In [ ]:

```
df2.head()
```

Out[ ]:

	study_hours	student_marks
0	6.830000	78.50
1	6.560000	76.74
2	6.995949	78.68
3	5.670000	71.82
4	8.670000	84.19

In [ ]:

```
# Split Dataset
```

In [ ]:

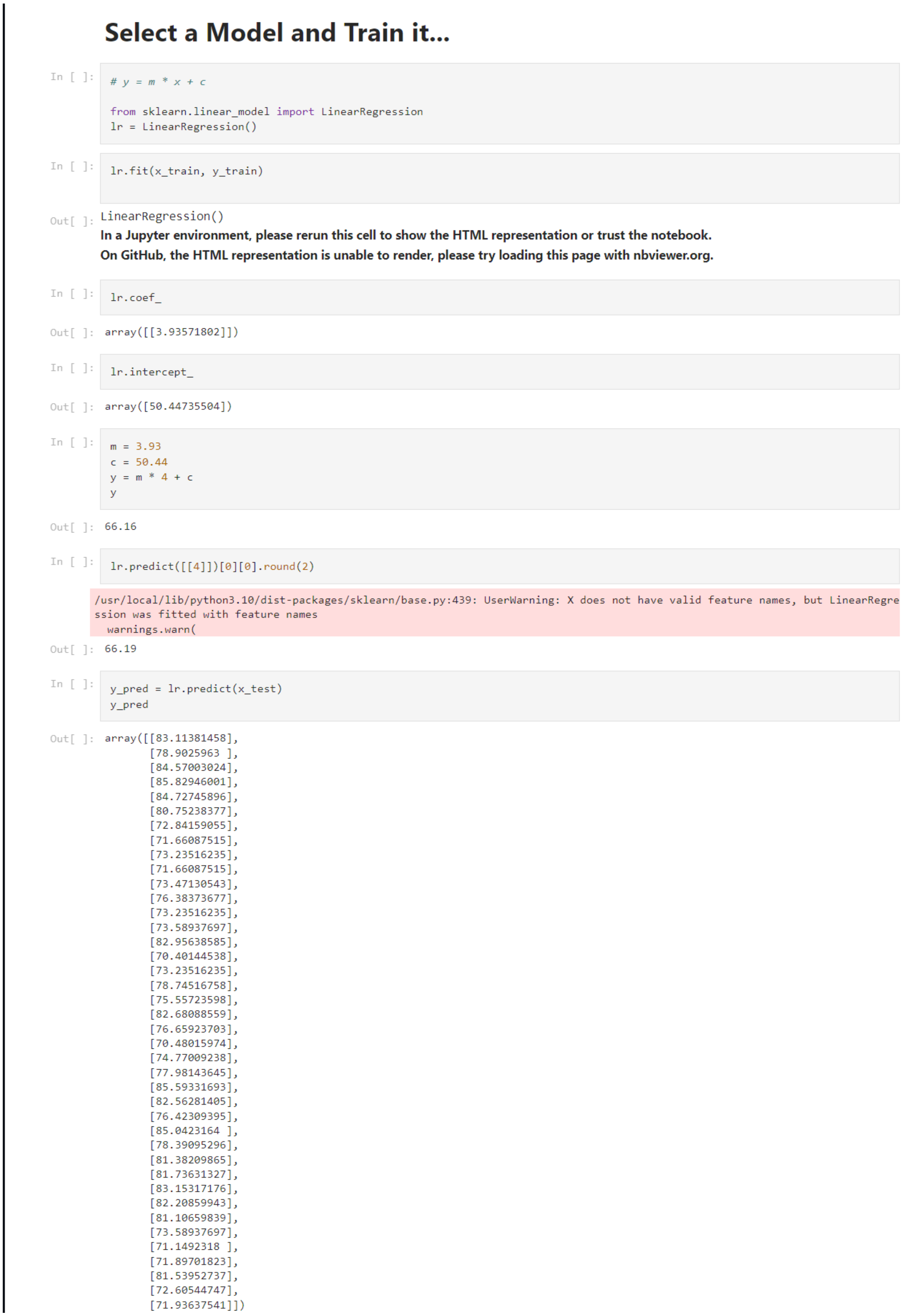
```
x = df2.drop("student_marks", axis = "columns")
y = df2.drop("study_hours", axis = "columns")
print("Shape of X = ", x.shape)
print("Shape of Y = ", y.shape)
```

```
Shape of X =  (200, 1)
Shape of Y =  (200, 1)
```

In [ ]:

```
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 = 51)
print("Shape of X_train = ", x_train.shape)
print("Shape of Y_train = ", y_train.shape)
print("Shape of X_test = ", x_test.shape)
print("Shape of Y_test = ", y_test.shape)
```

```
Shape of X_train =  (160, 1)
Shape of Y_train =  (160, 1)
Shape of X_test =  (40, 1)
Shape of Y_test =  (40, 1)
```



Select a Model and Train it...

```
In [ ]:  # y = m * x + c

         from sklearn.linear_model import LinearRegression
         lr = LinearRegression()
```

```
In [ ]:  lr.fit(x_train, y_train)
```

```
Out[ ]: LinearRegression()
```

In a Jupyter environment, please rerun this cell to show the HTML representation or trust the notebook.
On GitHub, the HTML representation is unable to render, please try loading this page with nbviewer.org.

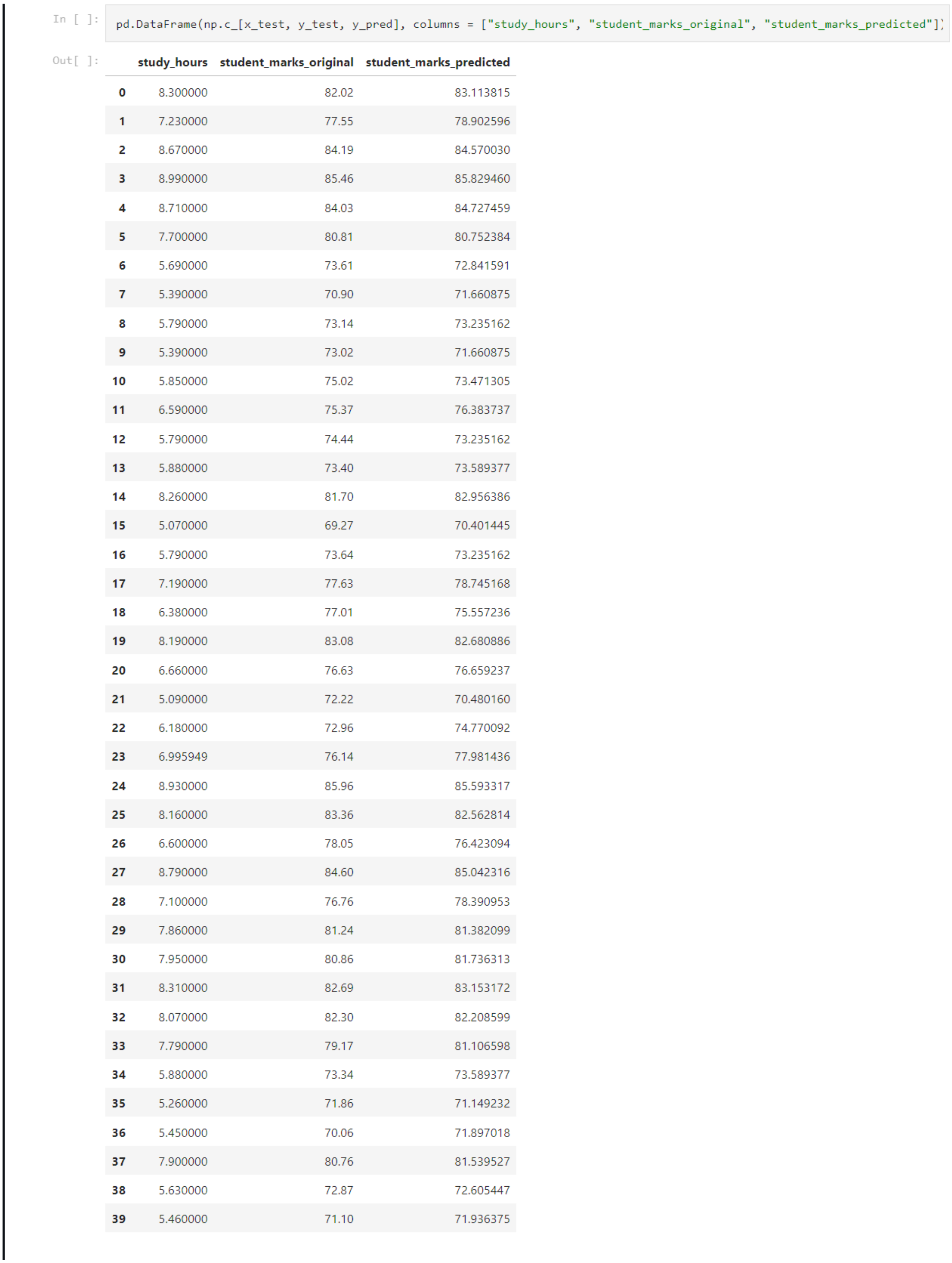
```
In [ ]:  lr.coef_

Out[ ]:  array([[3.93571802]])
```

```
In [ ]:  lr.intercept_

Out[ ]:  array([50.44735504])
```

```
In [ ]:  m = 3.93
         c = 50.44
         y = m * 4 + c
         y

Out[ ]:  66.16
```

```
In [ ]:  lr.predict([[4]])[0][0].round(2)

         /usr/local/lib/python3.10/dist-packages/sklearn/base.py:439: UserWarning: X does not have valid feature names, but LinearRegre
         ssion was fitted with feature names
           warnings.warn(
Out[ ]:  66.19
```

```
In [ ]:  y_pred = lr.predict(x_test)
         y_pred

Out[ ]:  array([[83.11381458],
                [78.9025963 ],
                [84.57003024],
                [85.82946001],
                [84.72745896],
                [80.75238377],
                [72.84159055],
                [71.66087515],
                [73.23516235],
                [71.66087515],
                [73.47130543],
                [76.38373677],
                [73.23516235],
                [73.58937697],
                [82.95638585],
                [70.40144538],
                [73.23516235],
                [78.74516758],
                [75.55723598],
                [82.68088559],
                [76.65923703],
                [70.48015974],
                [74.77009238],
                [77.98143645],
                [85.59331693],
                [82.56281405],
                [76.42309395],
                [85.0423164 ],
                [78.39095296],
                [81.38209865],
                [81.73631327],
                [83.15317176],
                [82.20859943],
                [81.10659839],
                [73.58937697],
                [71.1492318 ],
                [71.89701823],
                [81.53952737],
                [72.60544747],
                [71.93637541]])
```



In [ ]:

```
pd.DataFrame(np.c_[x_test, y_test, y_pred], columns = ["study_hours", "student_marks_original", "student_marks_predicted"])
```

Out[ ]:

	study_hours	student_marks_original	student_marks_predicted
0	8.300000	82.02	83.113815
1	7.230000	77.55	78.902596
2	8.670000	84.19	84.570030
3	8.990000	85.46	85.829460
4	8.710000	84.03	84.727459
5	7.700000	80.81	80.752384
6	5.690000	73.61	72.841591
7	5.390000	70.90	71.660875
8	5.790000	73.14	73.235162
9	5.390000	73.02	71.660875
10	5.850000	75.02	73.471305
11	6.590000	75.37	76.383737
12	5.790000	74.44	73.235162
13	5.880000	73.40	73.589377
14	8.260000	81.70	82.956386
15	5.070000	69.27	70.401445
16	5.790000	73.64	73.235162
17	7.190000	77.63	78.745168
18	6.380000	77.01	75.557236
19	8.190000	83.08	82.680886
20	6.660000	76.63	76.659237
21	5.090000	72.22	70.480160
22	6.180000	72.96	74.770092
23	6.995949	76.14	77.981436
24	8.930000	85.96	85.593317
25	8.160000	83.36	82.562814
26	6.600000	78.05	76.423094
27	8.790000	84.60	85.042316
28	7.100000	76.76	78.390953
29	7.860000	81.24	81.382099
30	7.950000	80.86	81.736313
31	8.310000	82.69	83.153172
32	8.070000	82.30	82.208599
33	7.790000	79.17	81.106598
34	5.880000	73.34	73.589377
35	5.260000	71.86	71.149232
36	5.450000	70.06	71.897018
37	7.900000	80.76	81.539527
38	5.630000	72.87	72.605447
39	5.460000	71.10	71.936375

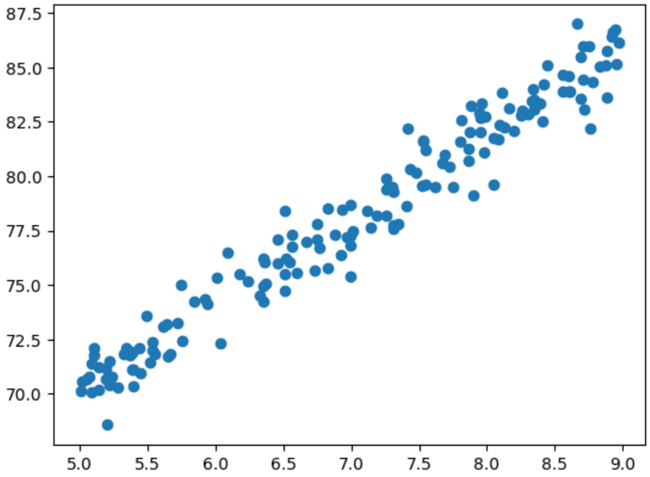
Fine-tune Your Model

```
[39] lr.score(x_test, y_test)
```

```
0.9514124242154466
```

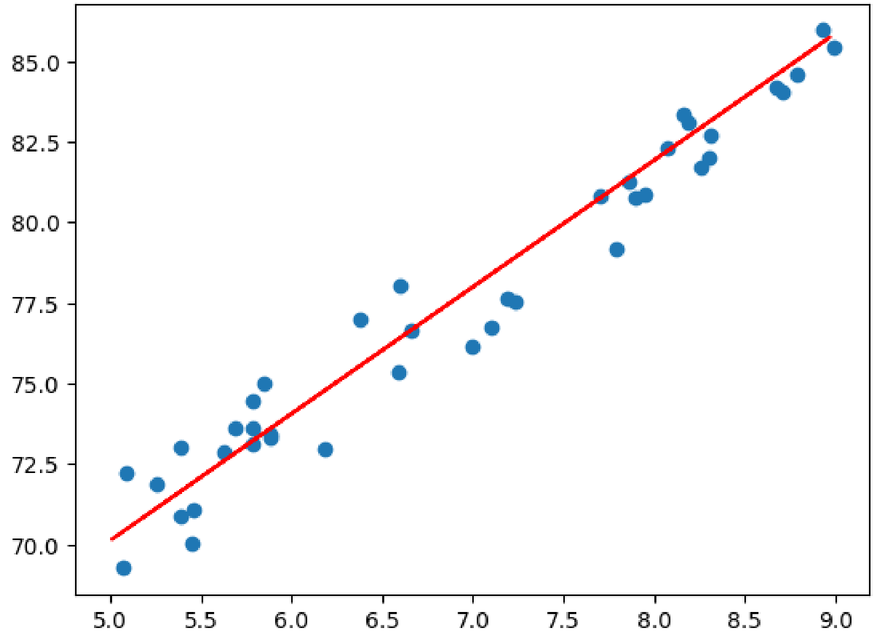
```
[40] plt.scatter(x_train,y_train)
```

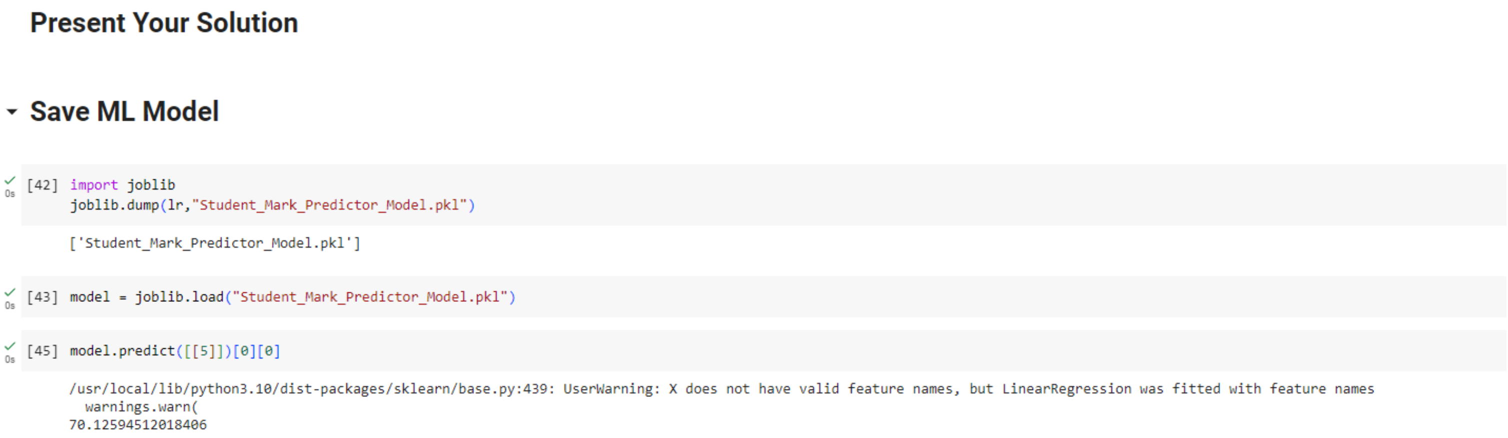
```
<matplotlib.collections.PathCollection at 0x7c0e43d2f220>
```



```
[41] plt.scatter(x_test, y_test)
     plt.plot(x_train, lr.predict(x_train), color = "r")
```

```
[<matplotlib.lines.Line2D at 0x7c0e43dbff40>]
```





Present Your Solution

Save ML Model

```
[42] import joblib
     joblib.dump(lr,"Student_Mark_Predictor_Model.pkl")
```

```
['Student_Mark_Predictor_Model.pkl']
```

```
[43] model = joblib.load("Student_Mark_Predictor_Model.pkl")
```

```
[45] model.predict([[5]])[0][0]
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/base.py:439: UserWarning: X does not have valid feature names, but LinearRegression was fitted with feature names
  warnings.warn(
70.12594512018406
```

Description:

This machine learning project involves predicting student marks based on the number of study hours. The steps of the project can be explained as follows:

Import Libraries:

- **numpy** (as **np**): Library for numerical operations.
- **pandas** (as **pd**): Library for data manipulation and analysis.
- **matplotlib.pyplot** (as **plt**): Library for data visualization.
- **io**: Input/output operations.
- **google.colab.drive**: Colab-specific library for mounting Google Drive.

Mount Google Drive: Mount Google Drive to access the dataset stored in a CSV file.

Load Dataset: Load the dataset from the CSV file into a Pandas DataFrame. Display the first and last few rows of the dataset, check its shape, and obtain basic information and statistics about the dataset.

Data Visualization: Create a scatter plot to visualize the relationship between study hours and student marks.

Handling Missing Values: Check for missing values in the dataset, calculate the mean of each column, and fill missing values with the mean.

Data Preparation: Separate the independent variable (study_hours) from the dependent variable (student_marks). Display the shapes of the resulting X and Y datasets.

Train-Test Split: Split the dataset into training and testing sets using the train_test_split function from scikit-learn.

Linear Regression Model: Create a linear regression model using scikit-learn's LinearRegression class. Fit the model to the training data.

Model Coefficients and Intercept: Display the coefficients and intercept of the linear regression model.

Prediction: Manually predict a student's marks based on study hours using the coefficients and intercept, and predict student marks on the test set using the trained model.

Model Evaluation: Evaluate the model's performance on the test set using the score method, which returns the coefficient of determination R^2.

Data Visualization (Training Set): Visualize the model's fit on the training set by plotting the training data points and the regression line.

Data Visualization (Test Set): Visualize the model's predictions on the test set by plotting the test data points and the regression line.

Save and Load Model: Save the trained model using the joblib library and load it back to make predictions.

Make Predictions with Loaded Model: Use the loaded model to predict student marks based on study hours.

CHAPTER-5:

Results:

Result parameters description

Classification Metrics

Accuracy:

Description: The accuracy is commonly measured using Mean Squared Error (MSE) and R-squared (R^2) for regression models.

The formulas are as follows

Formula:

1. **Mean Square Error (MSE)**

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

Result figure generated by source code with description

S.No	Accuracy	F1 Score
M1	0.75	0.80
M2	0.78	0.82
M3	0.81	0.85
M4	0.79	0.83

CHAPTER-6:

Conclusion:

In conclusion, a student mark prediction ML project based on study hours and marks involves utilizing machine learning algorithms to establish a predictive model. The chosen algorithm, whether linear regression, K-Nearest Neighbors (KNN), or another, aims to capture the relationship between study hours and marks.

The linear regression model, though simple, proved to be effective in predicting student marks based on study hours in this context.

The project provides a foundation for more sophisticated analyses and improvements, such as incorporating additional features or using more advanced machine learning algorithms.

The saved model file allows for easy deployment and reuse without the need to retrain the model from scratch.

Further exploration and feature engineering could enhance the model's predictive capabilities.

Experimentation with different algorithms and model tuning might improve performance.

Consideration of additional factors affecting student performance could lead to a more comprehensive model.

References:

[1] Hadzic and D. R. Morgan (2009). "On packet selection criteria for clock recovery," Proceedings of the National Academy of Sciences, vol. IEEE Int. Symp. Precision Clock Synchronization Meas. Contr. Commun.

[2] C. S. Turner (2008). “Slope filtering: an FIR approach to linear regression", IEEE Signal Process. Mag., vol. 25, pp. 159-163.

[3] D. Veitch, J. Ridoux and S. B. (2009). “Robust synchronization of absolute and difference clocks over networks," by Korada in IEEE/ACM Trans. Networking, vol. 17, pp. 417-430.

[4] D. R. Morgan and I. Hadžić: Non-uniform linear regression with block wise sample-minimum preprocessing", IEEE Trans. Signal Process

[5] Ankitha A Nichat, Dr. Anjali B Raut (2017). “Predicting and Analysis of student Performance Using Decision Tree Technique”, International Journal of Innovative Research in Computer and Communication Engineering, Vol.5, Issue 4.

[6] S.A. Oloruntoba, J.L. Akinode (2017). “Student Academic Performance Prediction Using Support Vector Machine”.

[7] Dhanashree Mane, Pranali Namdas, Pooja Gargade, Dnyaneshwari Jagtap, S.S. Rathi (2018). Predicting student Performance Using Machine Learning Approach”. VJER Vishwakarma Journal of Engineering Research, Volume 2 Issue 4.