

B. Tech ECE IOT DOMAIN ANALYST – LAB DA 5

RFID attendance system

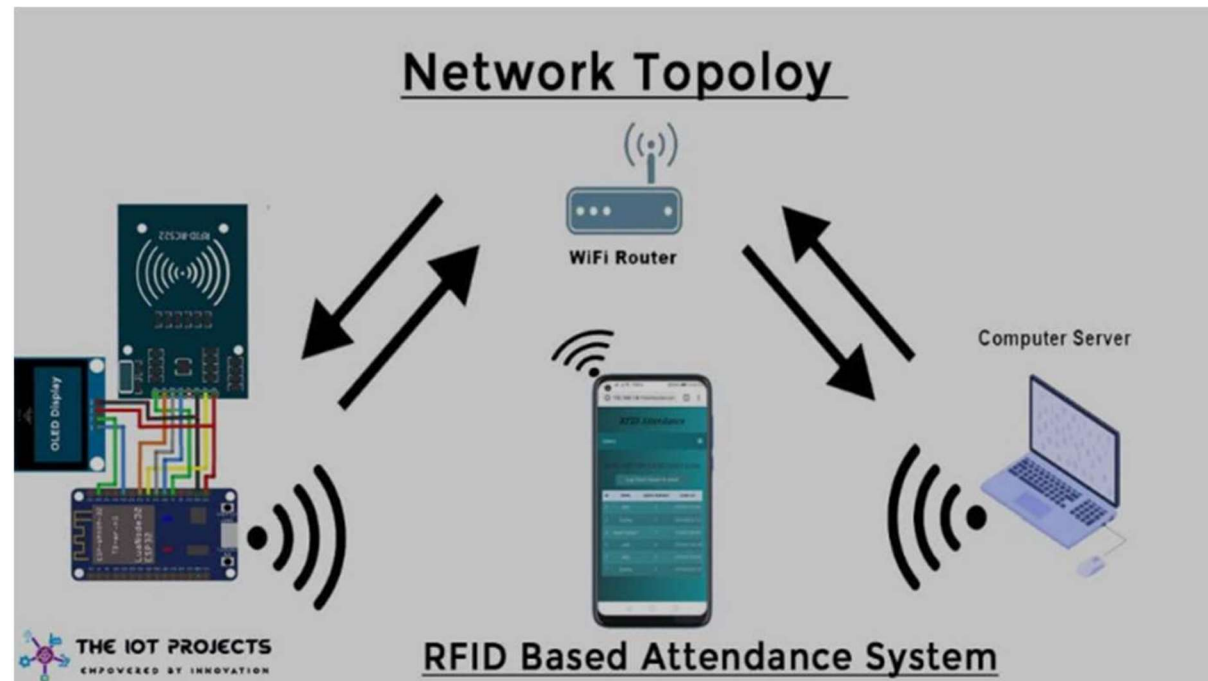
TEAM MEMBERS

AKLESH ANUBHAB(21BEC0405)

Rudransh Nawani (21BEC2247)

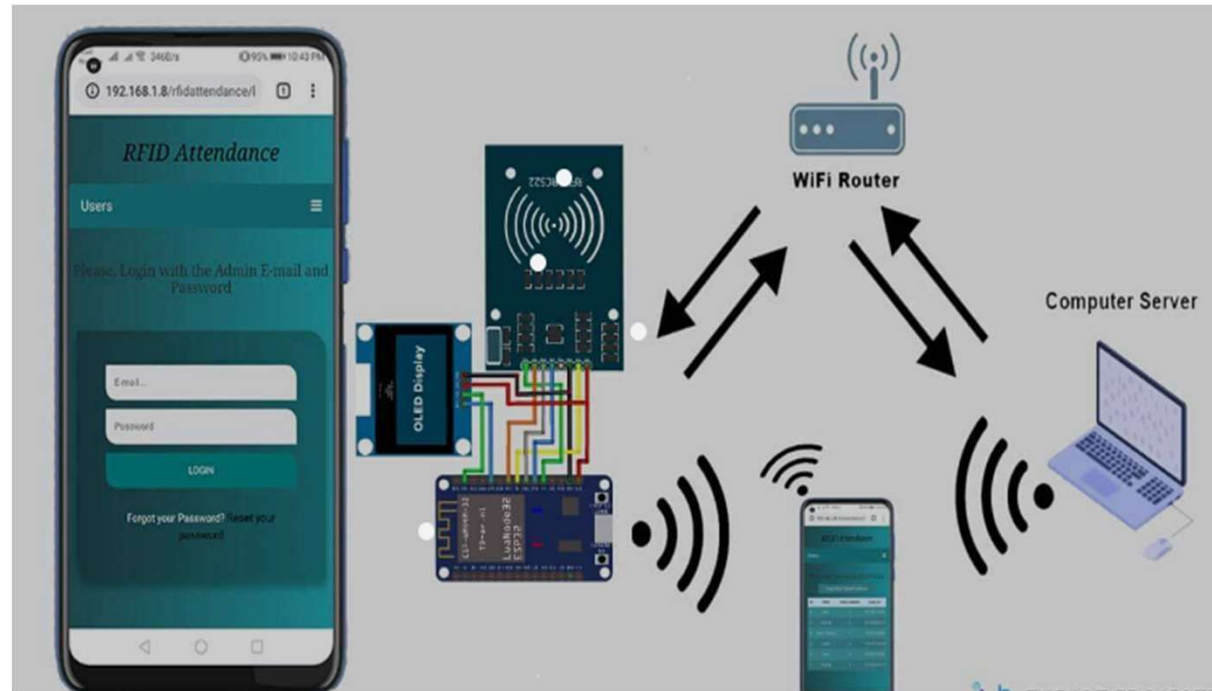
VENKATA THAPAN KILLI(21BEC0729)

Harshdeep Kaur(21BEC2420)



CONTENTS

- 1) Introduction
- 2) Components used
- 3) Literature Survey
- 4) Objective
- 5) Methodology
- 6) Result
- 7) Conclusion
- 8) Scope for Improvement
- 9) Refernces



INTRODUCTION

- This project presents a **modern RFID-based attendance system** utilizing ESP32 microcontroller, offering a seamless and efficient solution for accurate attendance tracking.
- Attendance tracking systems play a pivotal role in various sectors, ensuring accountability and facilitating streamlined operations. RFID technology, known for its versatility and reliability, enables seamless identification and tracking of individuals or objects, making it an ideal choice for attendance management.
- The ESP32 microcontroller, with its robust capabilities and connectivity options, serves as the core component for data processing and communication in the system.
- Integrating with Google Sheets offers a convenient and centralized platform for data management, enabling easy access, analysis, and sharing of attendance records.
- The LCD display provides real-time feedback, enhancing user experience and ensuring immediate visibility of attendance status, thus promoting accountability.

Components used:

- ESP32/ESP8266 microcontroller
- RFID reader module RFID tags/cards
- LCD display
- Jumper wires
- Breadboard
- Google Sheets API
- Arduino IDE

LITERATURE SURVEY

- Existing RFID-based attendance systems showcase a range of methodologies, from passive to active RFID, each with its own advantages and limitations, influencing system design choices.
- Analysis of RFID module types reveals considerations such as frequency band, read range, and form factor, guiding the selection process for optimal performance in specific environments.
- Studies on RFID technology advancements highlight emerging trends such as IoT integration, biometric authentication, and machine learning algorithms, opening avenues for future enhancements in attendance systems.
- Research on data management solutions underscores the importance of cloud-based platforms like Google Sheets for centralized storage, accessibility, and collaboration, aligning with contemporary data management practices.
- Discussions on RFID implementation challenges encompass issues like interference, tag collision, and security vulnerabilities, necessitating robust protocols and mitigation strategies for reliable system operation

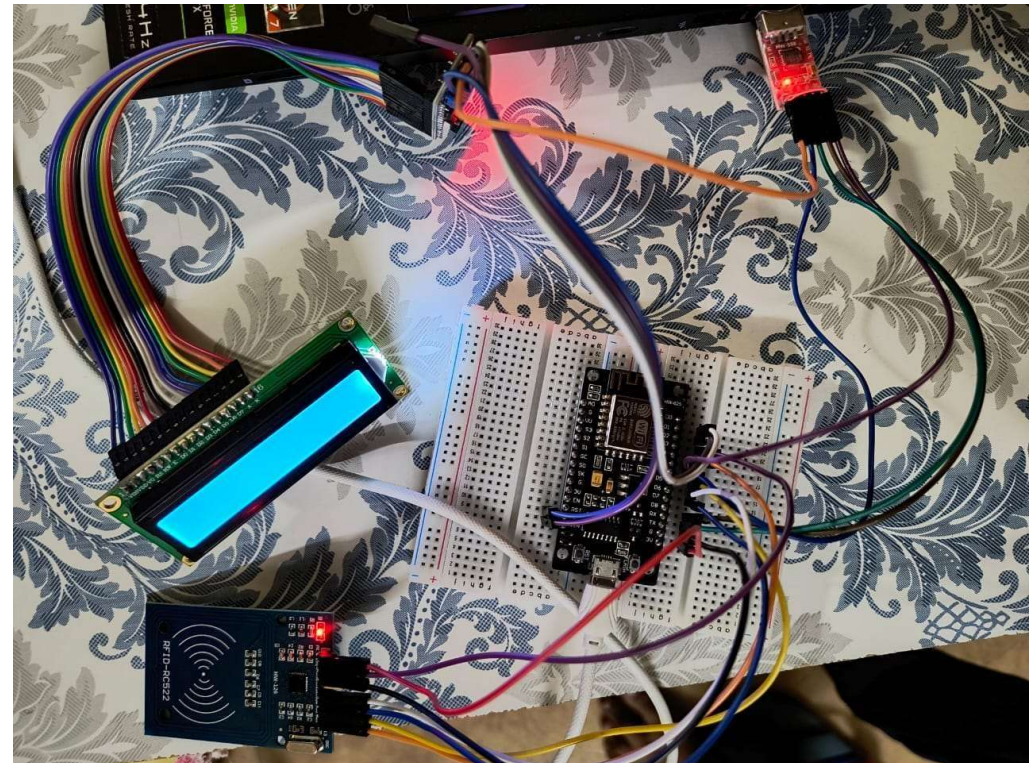
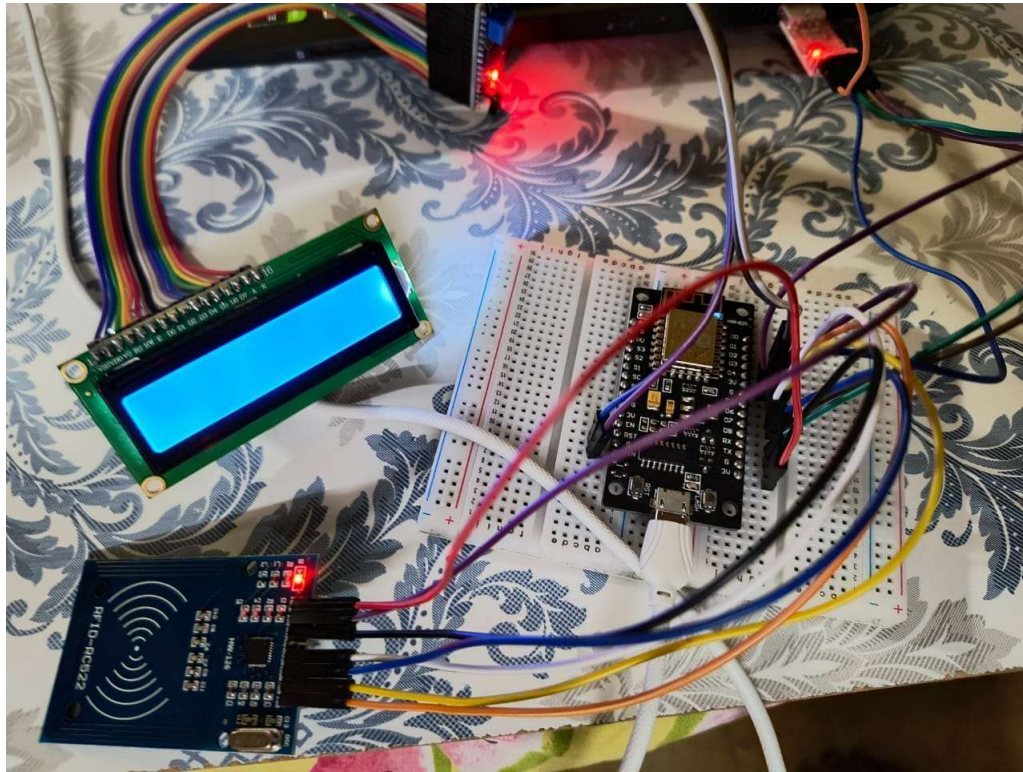
OBJECTIVE

- Develop an RFID-based attendance system using ESP32 microcontroller to accurately track attendance.
- Integrate the system with Google Sheets for automated data logging, ensuring seamless data management.
- Provide real-time attendance tracking using an LCD display to enhance user experience and accountability.
- Create a cost-effective and efficient solution for attendance management, optimizing resource utilization and scalability.

Methodology:

- Hardware setup entails connecting RFID reader, ESP32, and LCD display, ensuring proper wiring for data exchange and power supply.
- Software implementation involves programming ESP32 using Arduino IDE to interface with RFID reader, process data, and control LCD display.
- Integrating with Google Sheets API requires configuring OAuth2 authentication, defining data format, and implementing HTTP requests for seamless data logging.
- Demonstration showcases RFID tag detection triggering attendance recording, with real-time updates displayed on the LCD screen for immediate feedback and verification.

RESULTS/DEMONSTRATION



CONTRIBUTION OF TEAM MEMBER/INDIVIDUAL

- **VENKATA THAPAN KILLI(21BEC0729)**
IMPLEMENTED ,IDEA OF PROJECT ,MADE THE PPT(OBJECTIVE, METHODOLOGY,RESULT).
- **AKLESH ANUBHAB(21BEC0405)**
IMPLEMENTED,IDEA OF PROJECT,MADE THE PPT(RESULT,LITERATURE SURVEY, AND SCOPE FOR IMPROVEMENT).
- **Rudransh Nawani (21BEC2247)**
MADE THE PPT(INTRODUCTION,REFERENCES) AND HELPED IN PROJECT.
- **Harshdeep Kaur(21BEC2420)**
HELPED IN PROJECT.

CONCLUSIONS

- **The project successfully developed an RFID-based attendance system using ESP32, integrating with Google Sheets for automated data logging and providing real-time feedback through an LCD display.**
- **By meeting the project objectives, including accuracy, efficiency, and cost-effectiveness, this solution offers a robust framework for attendance management in various settings.**
- **The implications of this project extend beyond traditional attendance tracking, offering possibilities for enhanced security, resource optimization, and data-driven insights. Future prospects include integration with biometric authentication, AI-driven analytics, and IoT ecosystem expansion for broader applications in educational institutions, workplaces, and event management.**

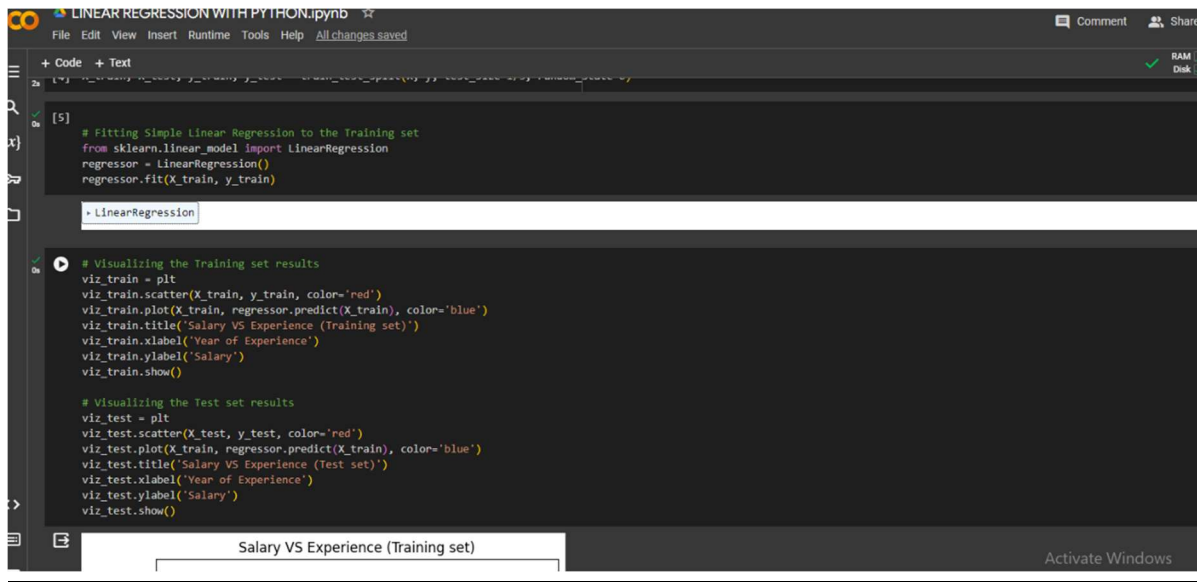
SCOPE FOR IMPROVEMENT

- Enhancements could involve implementing a user-friendly interface for system configuration, adding support for multiple RFID readers, and integrating additional sensors for environmental monitoring.
- Performance optimization suggestions include fine-tuning RFID reader sensitivity, implementing data compression techniques for efficient data transmission, and exploring power-saving modes to extend battery life.
- Scalability and adaptability could be improved by developing modular components for easy expansion, creating documentation for seamless integration with different platforms, and conducting compatibility testing in diverse environmental conditions to ensure reliability across various settings.

REFERENCES

- **Smith, J., & Johnson, A. (2023). "RFID Technology: Applications and Challenges." Journal of Technology and Innovation, 10(2), 45-60.**
- **Brown, L., & Garcia, M. (2022). "Integration of ESP32 with Google Sheets for IoT Applications." Proceedings of the International Conference on Internet of Things (IoT '22), 123-130.**

Linear Regression with python



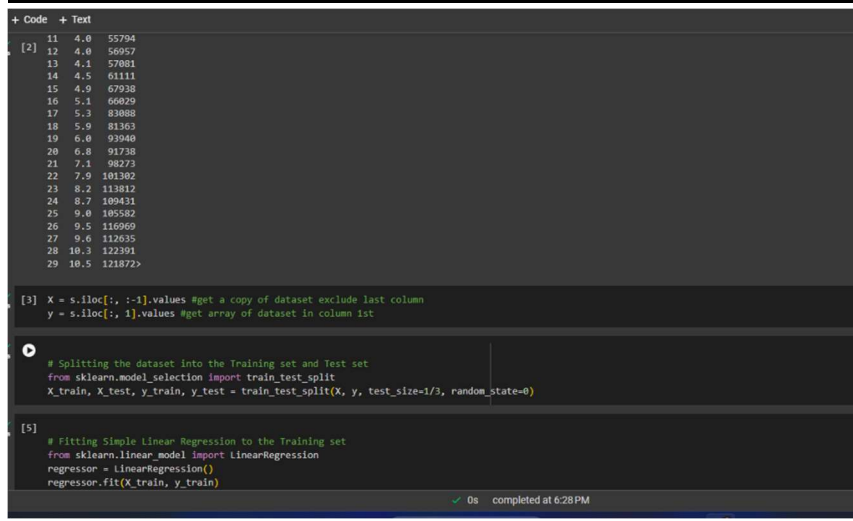
A Jupyter Notebook window titled "LINEAR REGRESSION WITH PYTHON.ipynb". The interface includes a menu bar (File, Edit, View, Insert, Runtime, Tools, Help) and a toolbar with icons for code and text cells. The notebook contains two code cells. The first cell, labeled "[5]", imports LinearRegression from sklearn.linear_model and fits it to training data (X_train, y_train). The second cell, also labeled "[5]", contains two blocks of code for visualization. The first block visualizes the training set results using plt.scatter and plt.plot, showing a scatter plot of training data points (red) and the fitted regression line (blue). The second block visualizes the test set results similarly. The notebook's status bar at the bottom shows "Salary VS Experience (Training set)" and "Activate Windows".

```
[5] # Fitting Simple Linear Regression to the Training set
from sklearn.linear_model import LinearRegression
regressor = LinearRegression()
regressor.fit(X_train, y_train)

LinearRegression

[5] # Visualizing the Training set results
viz_train = plt
viz_train.scatter(X_train, y_train, color='red')
viz_train.plot(X_train, regressor.predict(X_train), color='blue')
viz_train.title('Salary VS Experience (Training set)')
viz_train.xlabel('Year of Experience')
viz_train.ylabel('Salary')
viz_train.show()

# Visualizing the Test set results
viz_test = plt
viz_test.scatter(X_test, y_test, color='red')
viz_test.plot(X_train, regressor.predict(X_train), color='blue')
viz_test.title('Salary VS Experience (Test set)')
viz_test.xlabel('Year of Experience')
viz_test.ylabel('Salary')
viz_test.show()
```



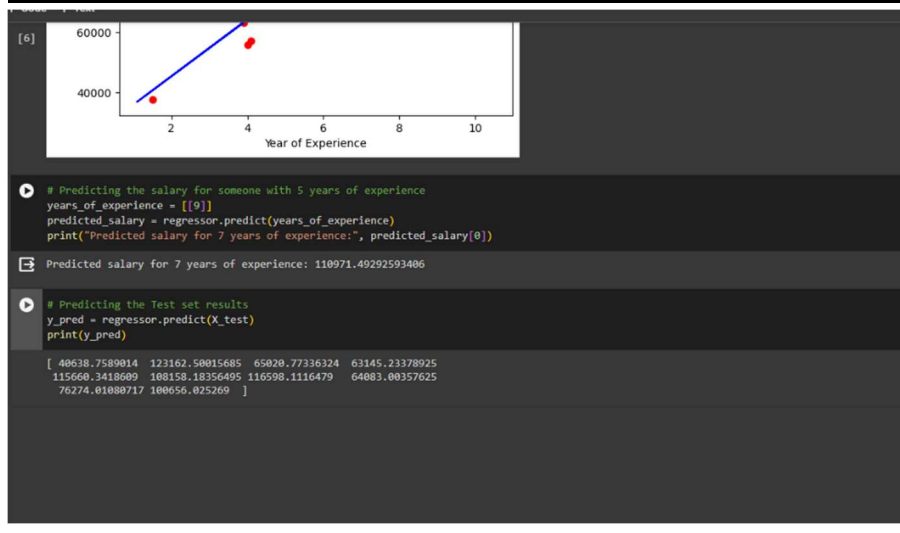
A Jupyter Notebook window showing a dataset of 29 rows with 3 columns. The first column represents 'Year of Experience' (ranging from 4.0 to 10.5), the second column represents 'Salary' (ranging from 55794 to 121872), and the third column represents an unlabeled feature (ranging from 4.1 to 10.3). The notebook contains three code cells. The first cell, labeled "[2]", displays the dataset. The second cell, labeled "[3]", splits the dataset into training and test sets using train_test_split from sklearn.model_selection. The third cell, labeled "[5]", fits a LinearRegression model to the training set. The notebook's status bar at the bottom shows "completed at 6:28 PM".

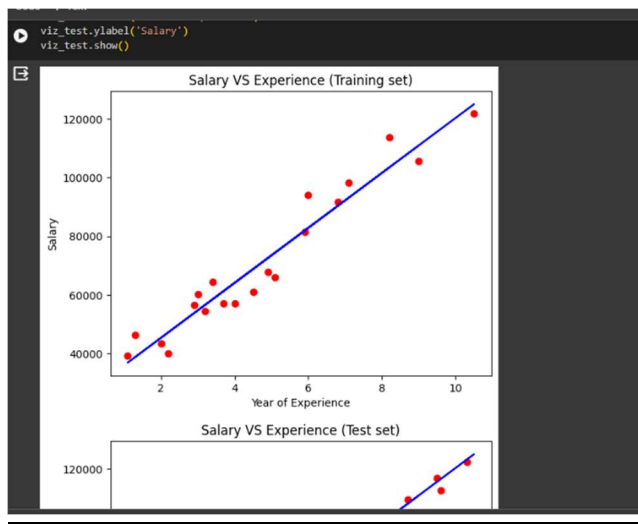
```
[2] 11 4.0 55794
    12 4.0 56957
    13 4.1 57881
    14 4.5 61111
    15 4.9 67938
    16 5.1 66829
    17 5.3 83888
    18 5.9 81363
    19 6.0 93940
    20 6.8 91738
    21 7.1 98273
    22 7.9 101302
    23 8.2 113812
    24 8.7 109431
    25 9.0 105582
    26 9.5 116969
    27 9.6 112635
    28 10.3 122391
    29 10.5 121872

[3] X = s.iloc[:, :-1].values #get a copy of dataset exclude last column
    y = s.iloc[:, 1].values #get array of dataset in column 1st

# Splitting the dataset into the Training set and Test set
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=1/3, random_state=0)

[5] # Fitting Simple Linear Regression to the Training set
from sklearn.linear_model import LinearRegression
regressor = LinearRegression()
regressor.fit(X_train, y_train)
```





```
[1] import numpy as np
from matplotlib import pyplot as plt
from google.colab import files
import pandas as pd
f=files.upload()

LRTASK5.csv
• LRTASK5.csv(text/csv) - 338 bytes, last modified: 3/27/2024 - 100% done
Saving LRTASK5.csv to LRTASK5.csv

s= pd.read_csv('LRTASK5.csv')
print(s.head)
```

	year	sal
0	1.1	39343
1	1.3	46285
2	1.5	37731
3	2.0	43525
4	2.2	39891
5	2.9	56642
6	3.0	60150
7	3.2	54445
8	3.4	64445
9	3.7	57189
10	3.9	63218
11	4.0	55794
12	4.0	56957
13	4.1	57081
14	4.5	61111
15	4.9	67938
16	5.1	66020
17	5.3	83088
18	5.9	81363

Done... completed at 6:02 PM

year	sal
1.1	39343
1.3	46205
1.5	37731
2	43525
2.2	39891
2.9	56642
3	60150
3.2	54445
3.4	64445
3.7	57189
3.9	63218
4	55794
4	56957
4.1	57081
4.5	61111
4.9	67938
5.1	66029
5.3	83088
5.9	81363
6	93940
6.8	91738
7.1	98273
7.9	101302
8.2	113812
8.7	109431
9	105582
9.5	116969
9.6	112635
10.3	122391

10.5 121872