A REPORT

ON

BURN RATE ANALYSIS USING LINEAR REGRESSION

By

Name of the student	Registration No.
YASWANTH SAI MANNEM	AP22110011084

Prepared in the partial fulfillment of the
Summer Internship Course

AT

EDUNET FOUNDATION

SRM UNIVERSITY, AP

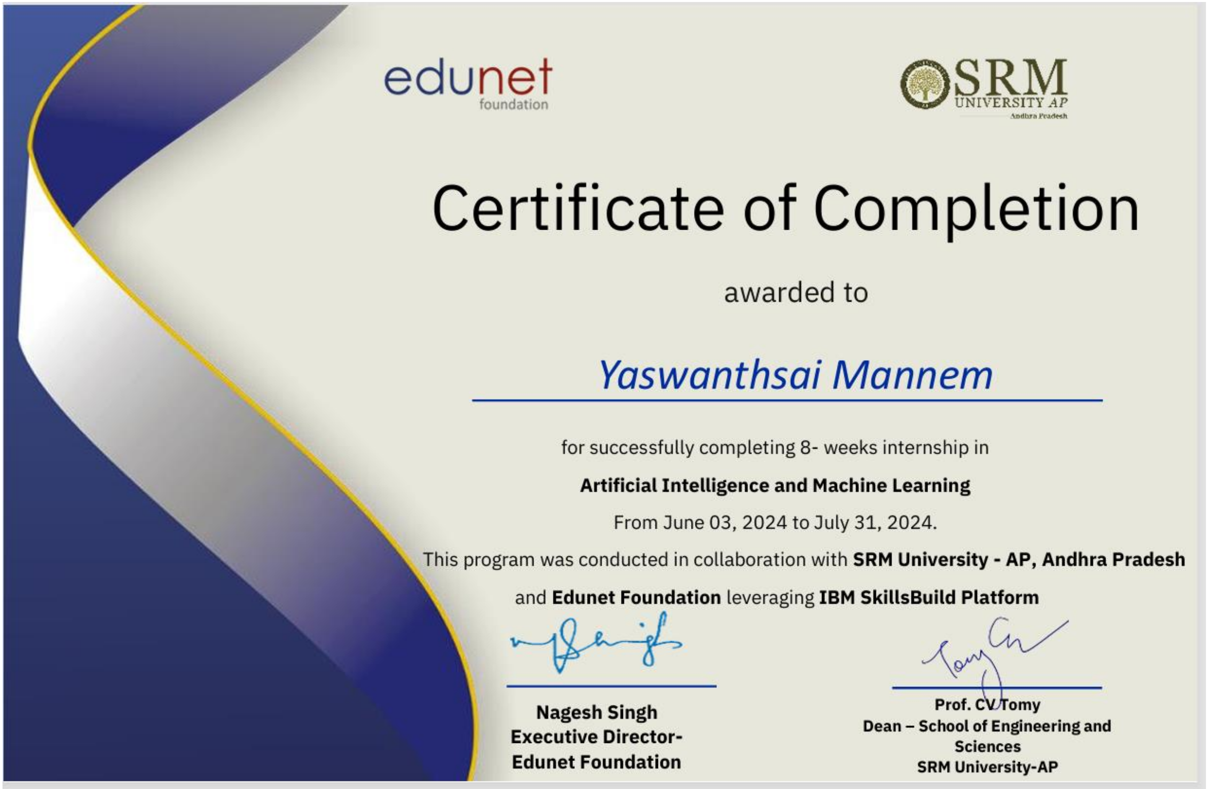
(July 2024)

Internship Completion Certificate:

CERTIFICATE

This is to certify that Summer Internship Project of YASWANTH SAI MANNEM titled BURN RATE ANALYSIS USING LINEAR REGRESSION is a record of bonafide work carried out by him under my supervision. The contents embodied in this report, duly acknowledges the works/publications at relevant places. The project work was carried during 03-06-2024 to 31-07-2024 in EDUNET FOUNDATION

Signature of Faculty Mentor	**Signature of industry Mentor/Supervisor (Not required for research internship)**
Name: SABYASACHI DUTTA	**Name: NAGESH SINGH**
Designation:	**Designation: Executive Director- Edunet Foundation**
Place: *Date:*	



edunet foundation

SRM UNIVERSITY AP Andhra Pradesh

Certificate of Completion

awarded to

Yaswanthsai Mannem

for successfully completing 8- weeks internship in

Artificial Intelligence and Machine Learning

From June 03, 2024 to July 31, 2024.

This program was conducted in collaboration with **SRM University - AP, Andhra Pradesh** and **Edunet Foundation** leveraging **IBM SkillsBuild Platform**

Nagesh Singh
Executive Director-
Edunet Foundation

Prof. CV Tomy
Dean – School of Engineering and Sciences
SRM University-AP

JOINING REPORT

Date: 20-05-2024

Name of the Student	YASWANTH SAI MANNEM
Roll No	AP22110011084
Program (BTech/ BSc/ BA/MBA)	BTech
Branch	CSE
Name and Address of the Internship Company [For research internship, it would be SRMAP]	EDUNET FOUNDATIONS. **ADDRESS:**XJ3V+64P, IBM India Private Limited, Embassy Golf Links, Off Indira Nagar-Koramangala, Intermediate Ring Rd, Embassy Golf Links Business Park, Domlur, Bengaluru, Karnataka 560071 Phone: 0124 408 0107 **E-mail** info@edunetfoundation.org
Period of Internship	**From** [03-06-2024] **to** [31-07-2024]

I hereby inform that I have joined the summer internship on Artificial Intelligence for the In-plant Training (8-week program).

Date: 20th MAY 2024 **Signature of the Student**

Yaswanthsai

ACKNOWLEDGEMENTS

I would like to express my deepest gratitude to those who have provided me with invaluable support and guidance throughout the completion of my internship in the domain of Artificial Intelligence at Edunet Foundation.

Firstly, I extend my heartfelt thanks to the Head of SRM University AP, **Prof. Manoj K Arora**, for his visionary support in encouraging students to pursue internships and providing the necessary university backing to make these opportunities possible.

I am also deeply thankful to our **CRCS Department** for sorting the internship opportunities and providing all possible options to help us choose the best fit. Your efforts have been instrumental in guiding us toward enriching learning experiences.

I am profoundly grateful to my Industry Mentor, **Ms. Anusha Tyagi Mam**, for her continuous guidance and encouragement. And I would like to thank **Nentheni Mam** for sharing and teaching. Her expertise and insights in the field of Artificial Intelligence and machine learning have greatly enhanced my understanding and contributed significantly to the success of my project.

I would also like to acknowledge and thank my Faculty Mentor, **Sabyasachi Dutta sir**, from **[SRM University AP]**, for their unwavering support and valuable feedback. Their academic guidance, along with the support from other staff members who came to teach us, has been essential in refining my project and ensuring its successful completion.

Finally, I would like to thank all the teaching and other staff members at the Edunet Foundation who have contributed directly or indirectly to this project. Your support and cooperation have been greatly appreciated.

Thank you all for making this a memorable and enriching learning experience.

ABSTRACT

This project report presents an analysis of employee burnout using machine learning techniques. Conducted by **Yaswanth sai Mannem** under the supervision of **EDUNET** foundation, this study aims to identify key factors contributing to employee burnout and predict burnout rates using a linear regression model. The dataset, sourced from Kaggle, includes variables such as Employee ID, Date of Joining, Gender, Company Type, WFH Setup Available, Designation, Resource Allocation, Mental Fatigue Score, and Burn Rate.

The methodology involves data preprocessing, feature selection, and the application of a linear regression model to understand the relationship between the selected features and employee burnout. The results indicate significant predictors of burnout and provide insights into how various factors influence employee well-being. These findings can aid organizations in developing targeted interventions to mitigate burnout and improve employee satisfaction.

Overall, this project highlights the importance of addressing employee burnout and demonstrates the potential of machine learning in providing actionable insights. The concise nature of this abstract ensures it is accessible to a wide range of readers, summarizing the project's scope, methodology, and key findings without delving into excessive detail or technical jargon.

TABLE OF CONTENTS:

BREIF INTRODUCTION OF ORGANIZATION BUSINESS SECTOR:

Education and Skill Development Industry in India:

The education and skill development industry in India is a critical sector that plays a significant role in shaping the country's future workforce and driving economic growth. This sector encompasses a broad range of activities aimed at enhancing the educational experience, bridging the gap between academic learning and industry requirements, and fostering innovation and entrepreneurship among the youth.

Historical Context and Evolution:

India's education system has undergone significant transformation over the years, from traditional forms of learning to modern, technology-driven education. The advent of the digital age and the recognition of the importance of skill development have led to the emergence of various initiatives and organizations focused on improving employability and fostering innovation.

Key Segments and Services:

1. **Educational Institutions:** This segment includes schools, colleges, and universities that provide formal education. These institutions are crucial for foundational learning and higher education, offering a wide range of courses and degrees.

2. **Vocational Training and Skill Development:** With the evolving job market and the advent of Industry 4.0, there is an increasing demand for vocational training and skill development programs. These programs aim to equip individuals with practical skills and knowledge required for specific trades or professions, enhancing their employability.

3. **EdTech Companies:** The rise of technology in education has given birth to numerous EdTech companies that provide online learning platforms, digital content, and educational tools. These companies cater to learners of all ages, offering courses ranging from basic education to advanced professional skills.

4. **Innovation and Entrepreneurship Hubs:** Organizations and foundations focusing on innovation and entrepreneurship are pivotal in nurturing the next generation of entrepreneurs. They provide resources, mentorship, and support for young innovators to develop their ideas and bring them to market.

5. **Industry-Academia Collaboration:** Bridging the gap between academia and industry is essential for creating a workforce that meets the demands of the modern job market. Various initiatives and partnerships are aimed at fostering collaboration between educational institutions and industries to ensure that the curriculum and training programs are aligned with industry requirements.

Conclusion

In summary, the education and skill development industry in India is a dynamic and evolving sector that plays a crucial role in the nation's development. By bridging the academia-industry divide, enhancing employability, and fostering innovation and entrepreneurship, organizations like Edunet Foundation are instrumental in preparing India’s youth for the challenges and opportunities of the future. This sector’s continued growth and adaptation to new technologies and industry demands are vital for maintaining India’s competitive edge in the global economy.

OVERVIEW OF EDUNET FOUNDATIONS

Brief history:

Edunet Foundation is a social enterprise which was founded in 2015 and focuses on bridging the academia-industry divide, enhancing student employability, promoting innovation and creating an entrepreneurial ecosystem in India. Working primarily with emerging technologies, and striving to leverage them to augment, upgrade the knowledge ecosystem and equip the beneficiaries to become contributors themselves, we work extensively to build a workforce with an IR 4.0 enabled career

Business size:

Edunet Foundation has had a significant influence on India. According to the latest data from LinkedIn, the organization employs over 201 to 500 individuals, including educators, software engineers, data scientists, and support workers. These individuals collaborate across departments to create and provide new educational solutions. The foundation has a countrywide presence and serves both urban and rural areas. Its programs serve tens of thousands of learners annually, demonstrating its broad reach and effectiveness.

Although the foundation is non-profit and does not engage in stock trading.

Product lines:

Content Curation:

An in-house team of subject matter experts that have developed tailor-made pieces for over 200+ universities, courses, and programs to achieve desired learning outcomes.

Technical Development:

A specialized tech team comprised of developers and testers that has delivered dedicated program sites, supplementary platforms, and learning management systems to facilitate effective learning.

Program Management:

Overseeing the full program deployment cycle right from program design through orientation, execution, and delivery, the experienced program management team is responsible for monitoring, evaluating, and scheduling all elements of a program.

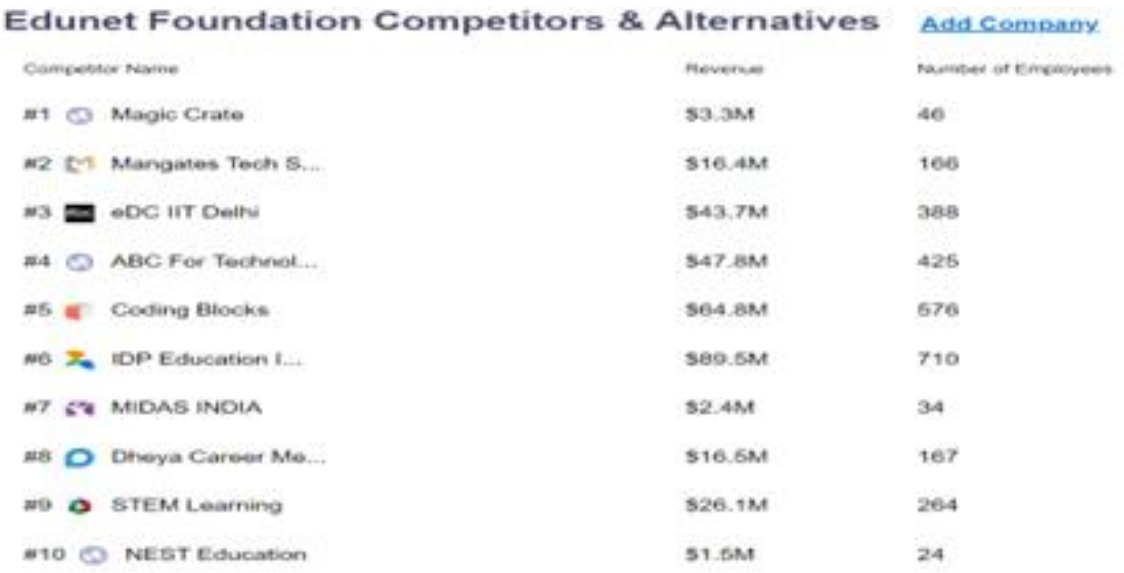
Expert Training:

On-ground specialists who execute the learning and training aspects of the program with deliberation. A well-connected network of trainers from all regions of the country ensures a healthy learning experience for the beneficiaries.

Internships:

Edunet Foundation offers internships to undergraduate students from diverse universities in various courses. By partnering with other companies, it provides online learning programs and mentorship for internship projects.

Competitor's:



Edunet Foundation Competitors & Alternatives

Competitor Name	Revenue	Number of Employees
#1 Magic Crate	$3.3M	46
#2 Mangates Tech S...	$16.4M	166
#3 eDC IIT Delhi	$43.7M	388
#4 ABC For Technol...	$47.8M	425
#5 Coding Blocks	$64.8M	576
#6 IDP Education I...	$89.5M	710
#7 MIDAS INDIA	$2.4M	34
#8 Dheya Career Me...	$16.5M	167
#9 STEM Learning	$26.1M	264
#10 NEST Education	$1.5M	24

PLAN OF MY INTERNSHIP

A brief introduction of the branch/department when i performed my internship:

I performed my internship in the AI & ML (Artificial Intelligence and Machine Learning) department at Edunet Foundation. This department focuses on advancing AI and ML technologies through research, development, and practical application. It provides comprehensive training programs, fosters industry collaborations, and supports innovative projects, preparing students to excel in the field of AI and ML.

START AND END DATE OF INTERNSHIP:

START DATE: 03-JUNE-2024
END DATE: 31-JULY-2024

Duties and responsibilities performed:

As my internship was virtual, I completed my AI learning plan on the IBM SkillsBuild platform. During this time, I learned about machine learning concepts and practiced coding that I learned through virtual classes. Additionally, I was assigned a project that required solving a problem using machine learning techniques.

The project I was assigned involves predicting employee burnout rates using linear regression. This task requires analyzing various factors related to employee well-being and performance to determine how they contribute to burnout. By applying linear regression techniques, I aim to build a model that can forecast burnout levels based on input features such as mental fatigue scores, resource allocation, and other relevant factors. The goal is to develop a predictive tool that can help organizations identify employees at risk of burnout and implement timely interventions to improve their well-being and productivity.

Background and description of the problem

Problem Statement:

- **Objective**: The project aims to develop a predictive model using linear regression to analyze employee burnout based on various factors such as demographics, work setup, and mental fatigue.

- **Use of the Project**: By pinpointing the primary factors contributing to burnout, the project aims to assist organizations in implementing proactive measures to manage and reduce employee burnout. This predictive capability enables targeted interventions, fostering a healthier and more productive workplace environment.

- **Definition and Importance of Burnout Analysis**: Burnout is a state of emotional, physical, and mental exhaustion caused by excessive and prolonged stress. Analyzing burnout is crucial as it helps organizations identify and address the root causes of burnout, promoting healthier work environments and enhancing employee well-being.

- **Increasing Incidence of Burnout**: Employee burnout is becoming increasingly common in various industries, leading to reduced productivity, higher turnover rates, and significant mental and physical health issues among employees.

- **Lack of Early Detection**: Organizations often fail to identify the early signs of burnout, resulting in prolonged stress and decreased overall employee well-being.

- **Limited Predictive Tools**: Existing tools and methods for detecting and managing burnout are often reactive rather than proactive, lacking the ability to predict burnout before it severely impacts employees.

- **Impact on Organizational Performance**: Unaddressed employee burnout can lead to significant financial and reputational costs for organizations, making it imperative to develop effective solutions to mitigate this issue.

ITS VALUE PROPOSITION:

- **Enhanced Employee Well-being**: The predictive model enables organizations to proactively manage employee burnout by identifying at-risk employees early. This allows HR and management teams to implement targeted interventions, improving overall employee well-being and satisfaction, and reducing turnover rates.

- **Strategic Resource Allocation**: By providing data-driven insights into the factors contributing to burnout, the model helps organizations make informed decisions about resource allocation, work policies, and employee support programs. This strategic approach enhances productivity, fosters a healthier work environment, and supports the organization's long-term success.

WHO ARE END USERS?

- **Human Resources (HR) Professionals**: HR professionals are primary end users of the burnout analysis model. They can utilize the insights to proactively address employee burnout, improve work conditions, enhance employee satisfaction, and reduce turnover rates. By identifying high-risk individuals or groups, HR can implement targeted interventions to maintain a healthier, more productive workforce.
- **Management and Executive Teams**: Management and executive teams can use burnout predictions to inform strategic decision-making. Understanding burnout trends and risk factors allows them to allocate resources effectively, design better policies, and foster a supportive work culture. This helps in improving overall organizational performance and maintaining a positive company reputation.
- And finally, Employees themselves.

Employee Burnout Prediction Analysis using Linear Regression Model

Introduction:

Employee burnout is a significant issue in today's fast-paced work environments, characterized by emotional exhaustion, depersonalization, and reduced personal accomplishment. Predicting burnout can help organizations implement preventive measures and support employee well-being. This report details the methodology and results of using a linear regression model to predict employee burnout. The analysis was conducted as part of an internship with Edunet Foundations.

Assumptions Made:

- **Linearity:** The relationship between predictor variables and burnout levels is linear.
- **Independence:** Predictor variables are independent of each other (no multicollinearity).
- **Homoscedasticity:** The residuals (errors) of the model are uniformly distributed.
- **Normality:** All The data in every column is normally distributed.

These assumptions are crucial for the validity of the linear regression model. Violations of these assumptions could affect the accuracy of the predictions.

Data Collection:

I had used the data set that was found in Kaggle website, that data was collected in Stanford research, Consisting Of a sample space 22750 rows and 9 columns.

Kaggle excel file link: https://www.kaggle.com/datasets/vijaysubhashp/employee-burnout-prediction

ALGORITHM:

- **Data Preprocessing**: Cleaning and preparing the dataset by handling missing values, encoding categorical variables, and normalizing numerical features to ensure accurate and reliable predictions.

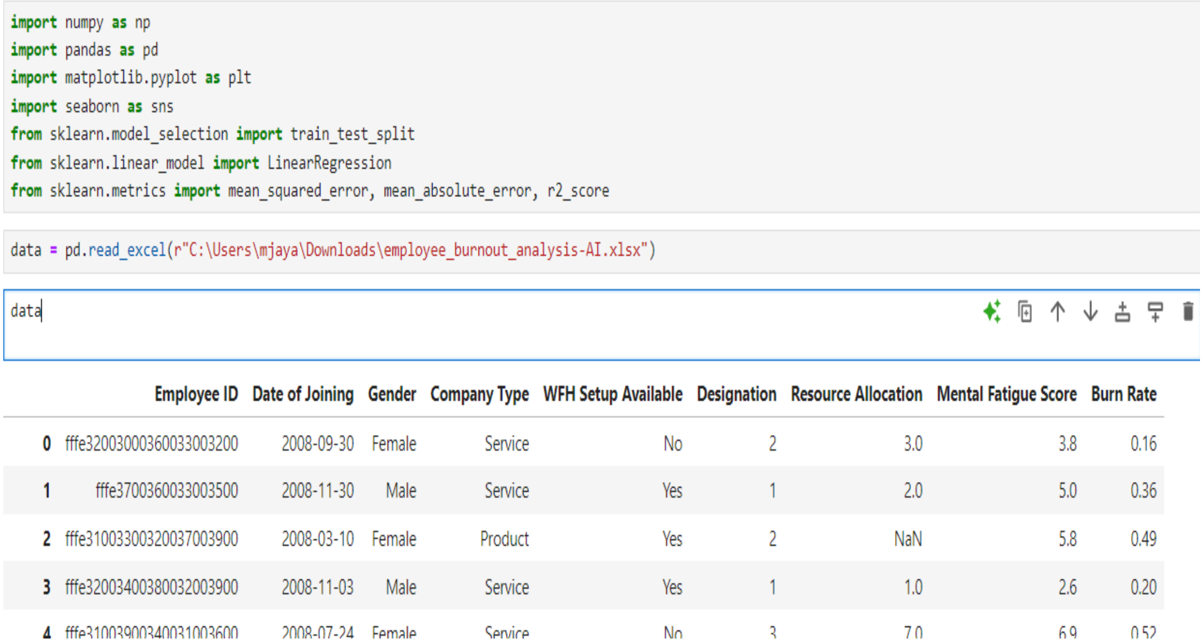
- **Model Development**: Utilize a linear regression model to predict the Burn Rate of employees by analyzing the relationships between the input features (such as Mental Fatigue Score, Resource Allocation) and the target variable (Burn Rate).

- **Evaluation and Validation**: Assess the model's performance using evaluation metrics such as R-squared and Mean Squared Error (MSE) and validate its accuracy with a test dataset to ensure reliability.

- **Implementation and Insights**: Implement the model to predict the burnout rate of employees, focusing on key predictive factors identified through the analysis.

METHODOLOGY:

Importing libraries & Loading data: Importing libraries (i.e. NumPy, pandas, matplotlib, ski kit) and loading the data set



```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, mean_absolute_error, r2_score
```

```
data = pd.read_excel(r"C:\Users\mjaya\Downloads\employee_burnout_analysis-AI.xlsx")
```

```
data
```

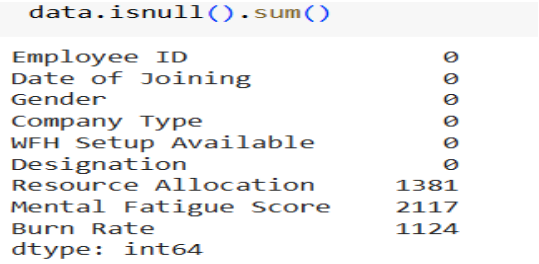
	Employee ID	Date of Joining	Gender	Company Type	WFH Setup Available	Designation	Resource Allocation	Mental Fatigue Score	Burn Rate
0	fffe32003000360033003200	2008-09-30	Female	Service	No	2	3.0	3.8	0.16
1	fffe3700360033003500	2008-11-30	Male	Service	Yes	1	2.0	5.0	0.36
2	fffe31003300320037003900	2008-03-10	Female	Product	Yes	2	NaN	5.8	0.49
3	fffe32003400380032003900	2008-11-03	Male	Service	Yes	1	1.0	2.6	0.20
4	fffe31003900340031003600	2008-07-24	Female	Service	No	3	7.0	6.9	0.52

Figure 1:importing libraries and loading data.

DATA SET INFORMATION:

- Employee ID: The unique ID allocated for each employee (example: **fffe390032003000**)
- Date of Joining: The date-time when the employee has joined the organization (example: **2008-12-30**)
- Gender: The gender of the employee (**Male/Female**)
- Company Type: The type of company where the employee is working (**Service/Product**)
- WFH Setup Available: Is the work from home facility available for the employee (**Yes/No**)
- Designation: The designation of the employee of work in the organization.
  - In the range of **[0.0, 5.0]** bigger is higher designation.
- Resource Allocation: The amount of resources allocated to the employee to work, ie. number of working hours.
  - In the range of **[1.0, 10.0]** (higher means more resource)
- Mental Fatigue Score: The level of fatigue mentally the employee is facing.
  - In the range of **[0.0, 10.0]** where 0.0 means no fatigue and 10.0 means completely fatigue.
- Burn Rate: The value we need to predict for each employee telling the rate of Bur out while working.
  - In the range of **[0.0, 1.0]** where the higher the value is more is the burn out.

Handling missing values:



```
data.isnull().sum()
```

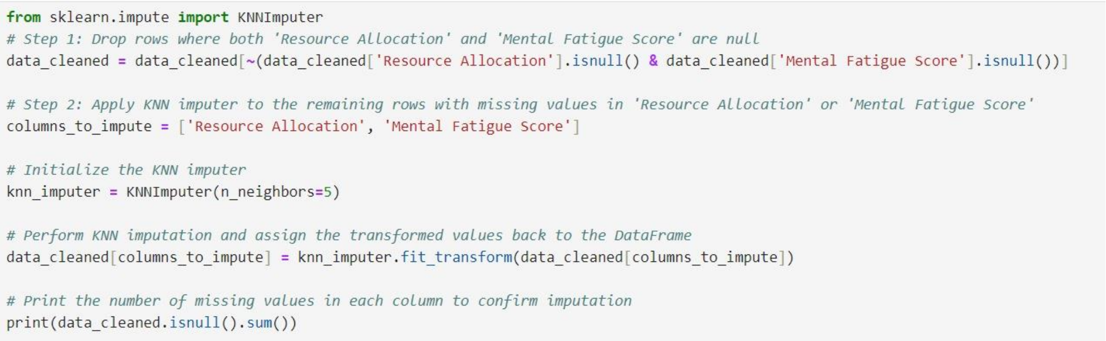
```
Employee ID                0
Date of Joining            0
Gender                     0
Company Type               0
WFH Setup Available        0
Designation                0
Resource Allocation     1381
Mental Fatigue Score    2117
Burn Rate               1124
dtype: int64
```

Figure 2:Count of Null values in data

1) We have observed that there are 1381 ,2117,1124 rows of null values in 3 columns respectively i.e. Resource allocation, Mental Fatigue Score, Burn rate.

2)Firstly, we drop of all rows that contains Burn rate as null values as it is a dependent variable and we can’t fill it with some false value and predict the same at the last, it will be just similar like teaching a non-accurate answer to a child, So we drop off 1124 rows and we will be remained with 21626 rows now.

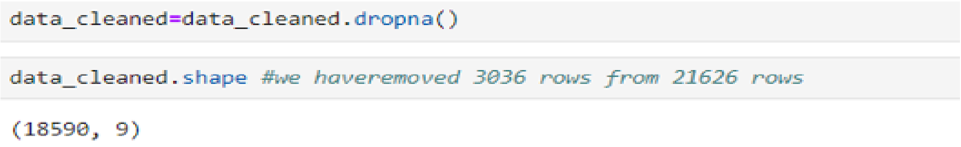
3)Now we are still left with 3036 rows that still contain null values that is 14% of data , Our first assumption now is we cant miss out this 14% of data so I tried to fill these null values with mean, mode, median, KNN imputer .After building linear regression with all these techniques the values filled by KNN imputer after dropping a row if more than one column have null value(i.e. we dropped 156 rows nearly) performed better with 90.41% accuracy by r-square score.



```
from sklearn.impute import KNNImputer
# Step 1: Drop rows where both 'Resource Allocation' and 'Mental Fatigue Score' are null
data_cleaned = data_cleaned[~(data_cleaned['Resource Allocation'].isnull() & data_cleaned['Mental Fatigue Score'].isnull())]

# Step 2: Apply KNN imputer to the remaining rows with missing values in 'Resource Allocation' or 'Mental Fatigue Score'
columns_to_impute = ['Resource Allocation', 'Mental Fatigue Score']

# Initialize the KNN imputer
knn_imputer = KNNImputer(n_neighbors=5)

# Perform KNN imputation and assign the transformed values back to the DataFrame
data_cleaned[columns_to_impute] = knn_imputer.fit_transform(data_cleaned[columns_to_impute])

# Print the number of missing values in each column to confirm imputation
print(data_cleaned.isnull().sum())
```

Figure 3:USAGE OF KNN IMPUTER TO FILL NULL DATA

4)But we have found that we got more accuracy by dropping all the rows that contains NULL values, this is because even after dropping all the rows with NULL values we are still left with 18590 rows which is enough to train the machine rather than polluting 14% of data with false data. So, we dropped all the Rows with NULL values and proceeded forward.



```
data_cleaned=data_cleaned.dropna()
```

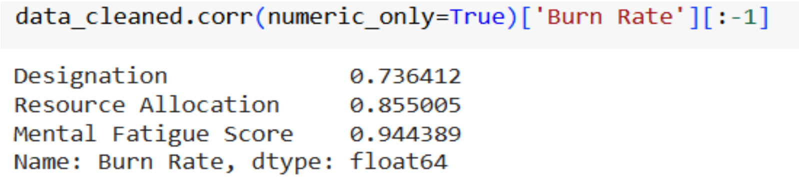
```
data_cleaned.shape #we haveremoved 3036 rows from 21626 rows
```

```
(18590, 9)
```

Figure 4:Shape of data after dropping all NULL values

Finding Correlation:

1.Correlation says how much an entity depends on the other. But we can only find this correlation between the columns with numerical data type. So, we chose the columns Designation, Resource allocation, Mental Fatigue Score and found their correlation on Burn rate.



```
data_cleaned.corr(numeric_only=True)['Burn Rate'][:-1]
```

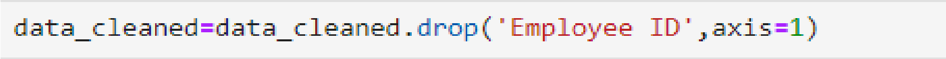
```
Designation             0.736412
Resource Allocation     0.855005
Mental Fatigue Score    0.944389
Name: Burn Rate, dtype: float64
```

Figure 5:Finding Correlation

2)As their correlation value is close to 1, we can conclude that these columns values are closely affecting Burn rate, with this conclusion we move further to check about other columns how strong they are related to Burn rate.

Analyzing categorical data:

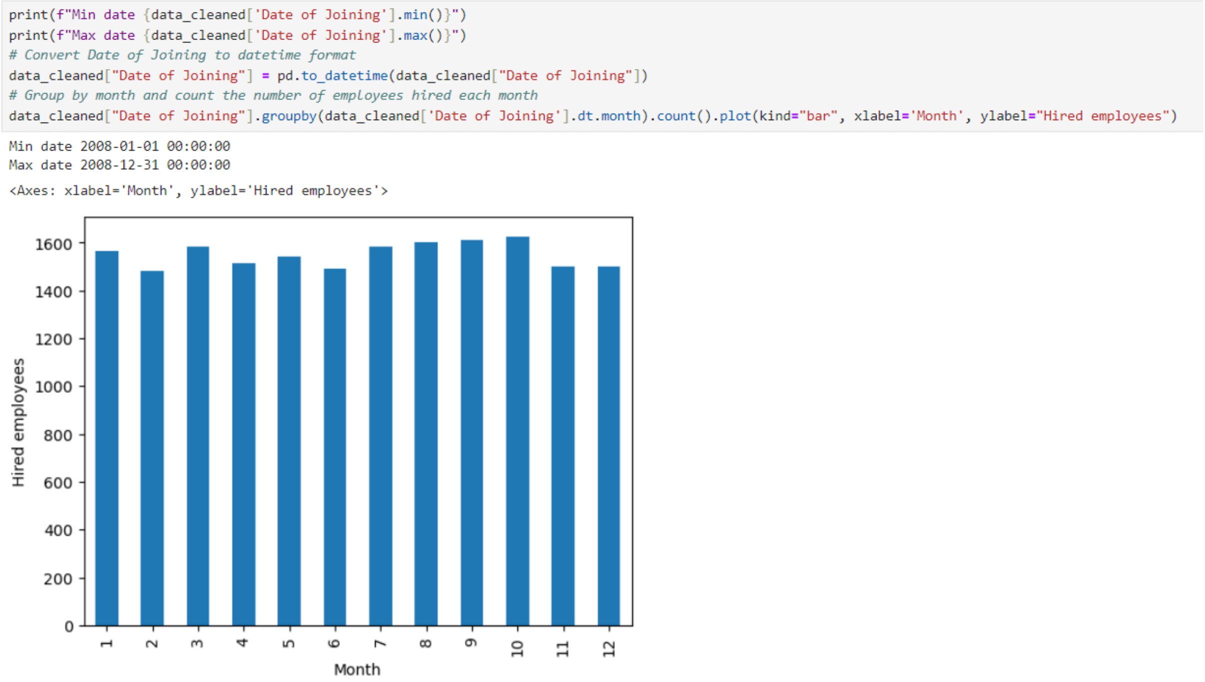
1.Coming to categorical data We have 5 rows Employee ID, Date of Joining, Gender, Company type, WFH setup. In this we know that Employee ID Doesn't show any effect on Burn rate So we simply **Drop that Column Employee ID**.



```
data_cleaned=data_cleaned.drop('Employee ID',axis=1)
```

Figure 6:Analyzed Catagorical data 1

2.Later coming to Date of joining it may affect the burn rate as the days they worked for company may come into scene so firstly we wanted to check the time range of the employees in data to check what's the min and max dates are and check we have the data spread normally over the range of time.

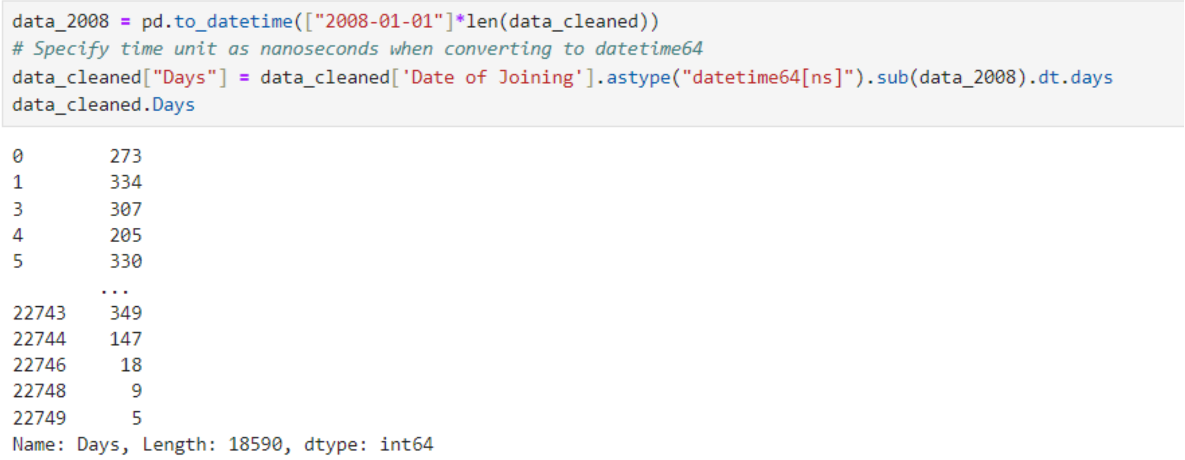


```
print(f"Min date {data_cleaned['Date of Joining'].min()}")
print(f"Max date {data_cleaned['Date of Joining'].max()}")
# Convert Date of Joining to datetime format
data_cleaned["Date of Joining"] = pd.to_datetime(data_cleaned["Date of Joining"])
# Group by month and count the number of employees hired each month
data_cleaned["Date of Joining"].groupby(data_cleaned['Date of Joining'].dt.month).count().plot(kind="bar", xlabel='Month', ylabel="Hired employees")
```

```
Min date 2008-01-01 00:00:00
Max date 2008-12-31 00:00:00
<Axes: xlabel='Month', ylabel='Hired employees'>
```

3)Here we can see that the min date is start of the year 2008 and the max date is end of the year 2008, so the data is hired employees in single year, and we here have the data of all hired employees spread normally all over the year, so we have enough data to proceed with it.

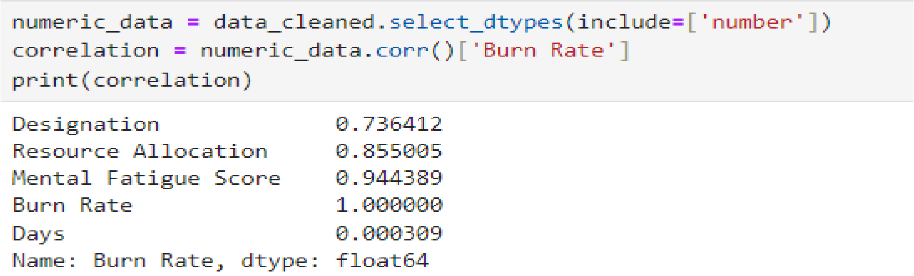
4)So now we will convert this categorical data into numerical data by converting the date of joining to date time format and find how many days they have worked in that year and store it in a new column "Days" and find its correlation with Burn rate as it is now a numerical data.



```
data_2008 = pd.to_datetime(["2008-01-01"]*len(data_cleaned))
# Specify time unit as nanoseconds when converting to datetime64
data_cleaned["Days"] = data_cleaned['Date of Joining'].astype("datetime64[ns]").sub(data_2008).dt.days
data_cleaned.Days
```

```
0        273
1        334
3        307
4        205
5        330
        ... 
22743    349
22744    147
22746     18
22748      9
22749      5
Name: Days, Length: 18590, dtype: int64
```

Figure 7:Obtaining no of days they worked from the date of joining and storing in a new column "Days"

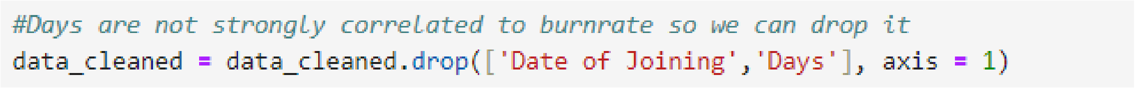


```
numeric_data = data_cleaned.select_dtypes(include=['number'])
correlation = numeric_data.corr()['Burn Rate']
print(correlation)
```

```
Designation             0.736412
Resource Allocation     0.855005
Mental Fatigue Score    0.944389
Burn Rate               1.000000
Days                    0.000309
Name: Burn Rate, dtype: float64
```

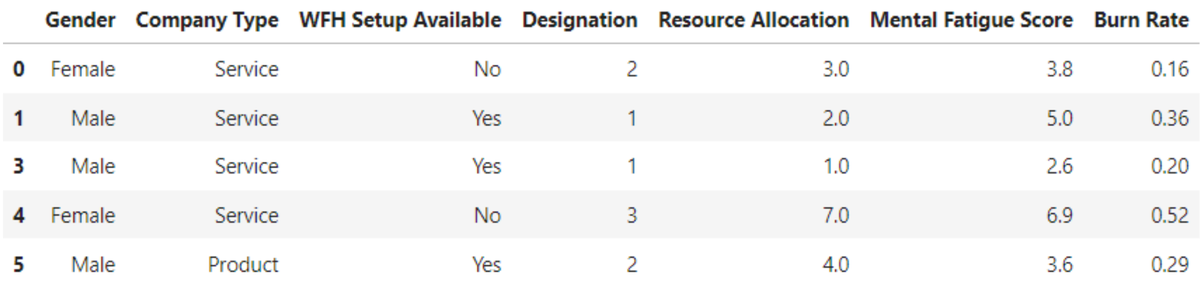
Figure 8:Finding correlation of days to Burn rate

5)After converting it into days we checked the correlation of it with burn rate as it is in numerical format as well; after checking we got much less correlation with burn rate (i.e. 0.000309), so we **dropped the column Date of Joining**.



```
#Days are not strongly correlated to burnrate so we can drop it
data_cleaned = data_cleaned.drop(['Date of Joining','Days'], axis = 1)
```

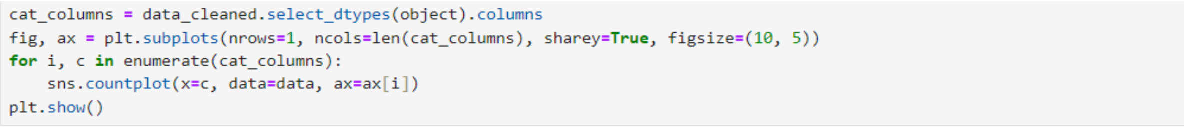
Figure 9:Analyzed categorical value 2



	Gender	Company Type	WFH Setup Available	Designation	Resource Allocation	Mental Fatigue Score	Burn Rate
0	Female	Service	No	2	3.0	3.8	0.16
1	Male	Service	Yes	1	2.0	5.0	0.36
3	Male	Service	Yes	1	1.0	2.6	0.20
4	Female	Service	No	3	7.0	6.9	0.52
5	Male	Product	Yes	2	4.0	3.6	0.29

Figure 10:Now we are left with 3 more columns consisting of categorical variables.

7)Now we are left to analyze 3 more columns, here each column consists only two unique values in it, and they serve importance on Burn rate as we know. Now what we need to find is these values spread normally so that no data is missing out and to ensure that we have enough data in every category to proceed.



```
cat_columns = data_cleaned.select_dtypes(object).columns
fig, ax = plt.subplots(nrows=1, ncols=len(cat_columns), sharey=True, figsize=(10, 5))
for i, c in enumerate(cat_columns):
    sns.countplot(x=c, data=data, ax=ax[i])
plt.show()
```

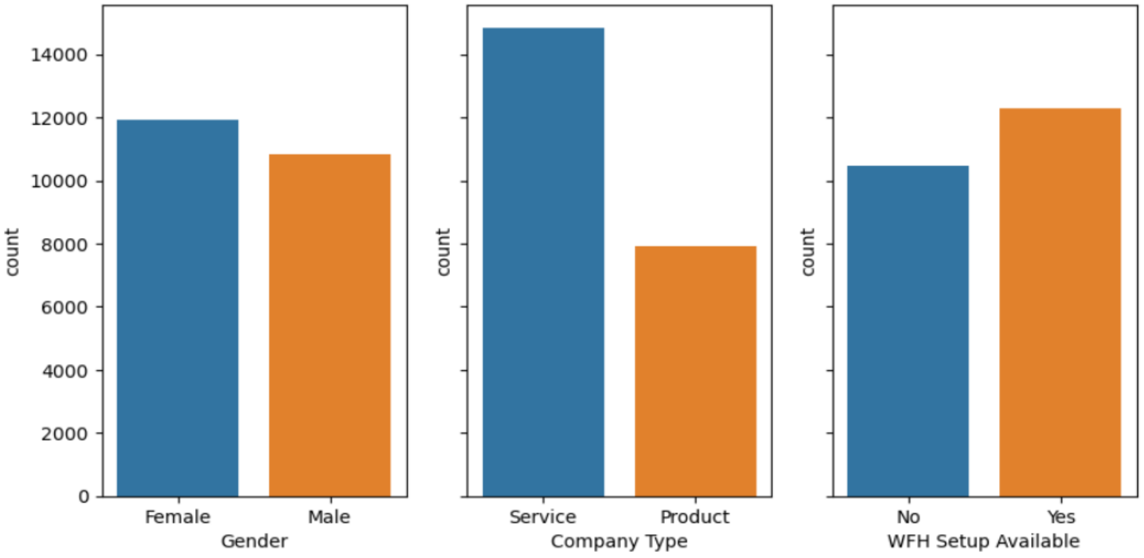
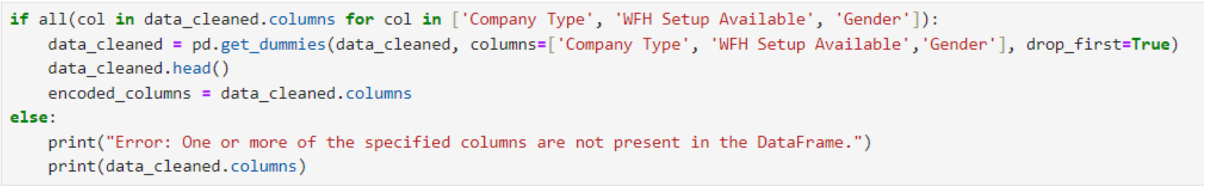


Figure 11:Analyzed all categorical columns

<u>Encoding categorical data</u>: From Figure 10 we can conclude that we have enough information for every variable, that means our data is spread normally. Now our task is to convert this categorical data into numerical data as they are needed to me understand by machine learning model, so to do this **we use one-hot encoding for Gender, Company type, WFH setup available columns.**



```
if all(col in data_cleaned.columns for col in ['Company Type', 'WFH Setup Available', 'Gender']):
    data_cleaned = pd.get_dummies(data_cleaned, columns=['Company Type', 'WFH Setup Available','Gender'], drop_first=True)
    data_cleaned.head()
    encoded_columns = data_cleaned.columns
else:
    print("Error: One or more of the specified columns are not present in the DataFrame.")
    print(data_cleaned.columns)
```

Figure 12:Used One-hot encoding on 3 columns (i.e. In this process we Encode the categorical values with dummy numerical values.)

With this we have finished analyzing our categorical data and it’s now time to proceed to our next step.

Splitting dataset and scaling:

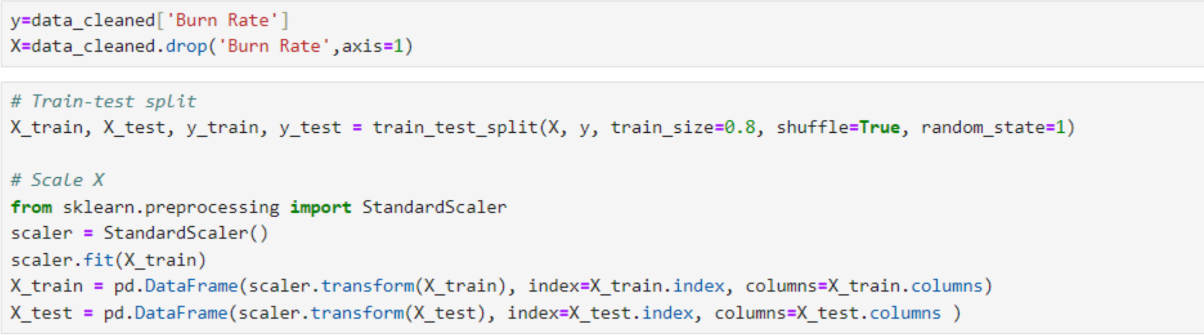
1)Here we split the data into 4 parts where the first two parts are X and y, after that we further divide them both in 7:3 ratio called training and testing set,7 parts are for training and 3 parts for testing.

2)X contains the features (independent variables) that will be used to make predictions.

3) y contains the target variable (Burn Rate) that you want to predict.

4)**Splitting into "X_train", "X_test", "y_train", and "y_test": train_test_split function divides the data** so that you can train your model on X_train and y_train (70% of the data) and then test its performance on "X_test" and "y_test" (30% of the data). This helps in evaluating how well the model generalizes to unseen data.

4)And to scale the data we used StandardScaler.

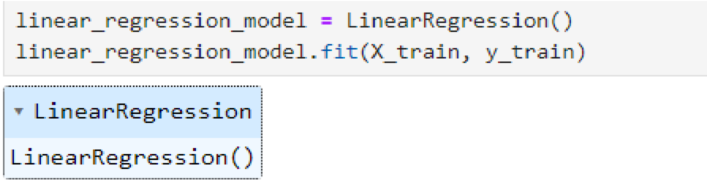


```
y=data_cleaned['Burn Rate']
X=data_cleaned.drop('Burn Rate',axis=1)
```

```
# Train-test split
X_train, X_test, y_train, y_test = train_test_split(X, y, train_size=0.8, shuffle=True, random_state=1)

# Scale X
from sklearn.preprocessing import StandardScaler
scaler = StandardScaler()
scaler.fit(X_train)
X_train = pd.DataFrame(scaler.transform(X_train), index=X_train.index, columns=X_train.columns)
X_test = pd.DataFrame(scaler.transform(X_test), index=X_test.index, columns=X_test.columns )
```

Figure 13:Split the data and used standard scaler to scale the data.

Fitting into linear regression model: Here we fit the training data set into linear regression model. We now have 5 independent variables (i.e. Designation, Resource Allocation, MFS, Company Type, WFH setup) and one dependent variable that is to be predicted (i.e. burn rate).

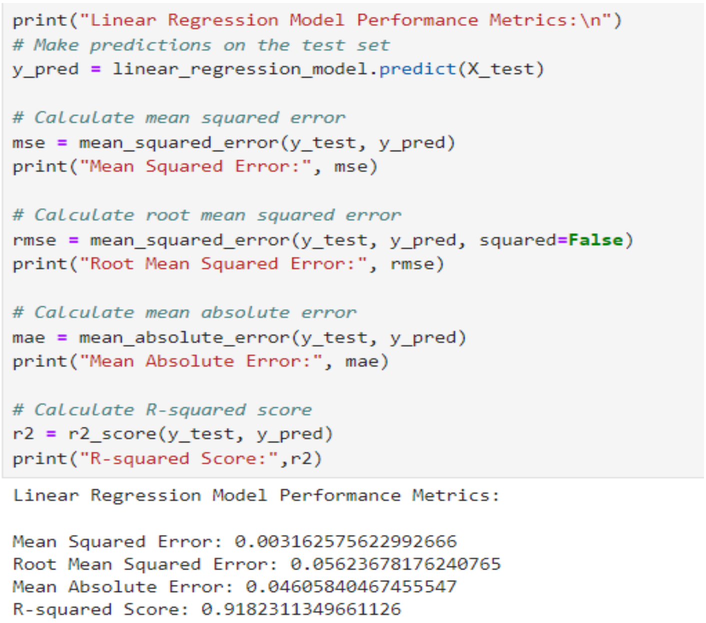


```
linear_regression_model = LinearRegression()
linear_regression_model.fit(X_train, y_train)
```

```
▾ LinearRegression
LinearRegression()
```

Figure 14:LinearRegression model

Results: Here is the result that how accurate our model can predict the dependent variable using testing set (By comparing the y_test and y_predicted). We here predict the error and accuracy of our model with these performance metrics.



```
print("Linear Regression Model Performance Metrics:\n")
# Make predictions on the test set
y_pred = linear_regression_model.predict(X_test)

# Calculate mean squared error
mse = mean_squared_error(y_test, y_pred)
print("Mean Squared Error:", mse)

# Calculate root mean squared error
rmse = mean_squared_error(y_test, y_pred, squared=False)
print("Root Mean Squared Error:", rmse)

# Calculate mean absolute error
mae = mean_absolute_error(y_test, y_pred)
print("Mean Absolute Error:", mae)

# Calculate R-squared score
r2 = r2_score(y_test, y_pred)
print("R-squared Score:",r2)
```

```
Linear Regression Model Performance Metrics:

Mean Squared Error: 0.003162575622992666
Root Mean Squared Error: 0.05623678176240765
Mean Absolute Error: 0.04605840467455547
R-squared Score: 0.9182311349661126
```

Figure 15:linear regression model performance metrics (after dropping all null valued rows)

Findings:

- **Model Performance:**

The linear regression model achieved an R² value of 0.9188, indicating that 91.88% of the variance in burnout levels was explained by the model.

- **Significant Predictors:**

Job satisfaction and hours worked were the most significant predictors of burnout.

- **Illustrations:**

Visualizing the relationship between the predicted and actual values means creating a plot where you can compare the predictions made by your model to the actual values from your dataset. This is typically done using a scatter plot where:

- The **x-axis** represents the actual values from your dataset.
- The **y-axis** represents the predicted values from your model.

In this plot, each point represents an observation from your dataset. If your model were perfect, all points would lie on a 45-degree line (y = x), because the predicted values would exactly match the actual values. The best fit line in this context is the 45-degree line which indicates perfect prediction.

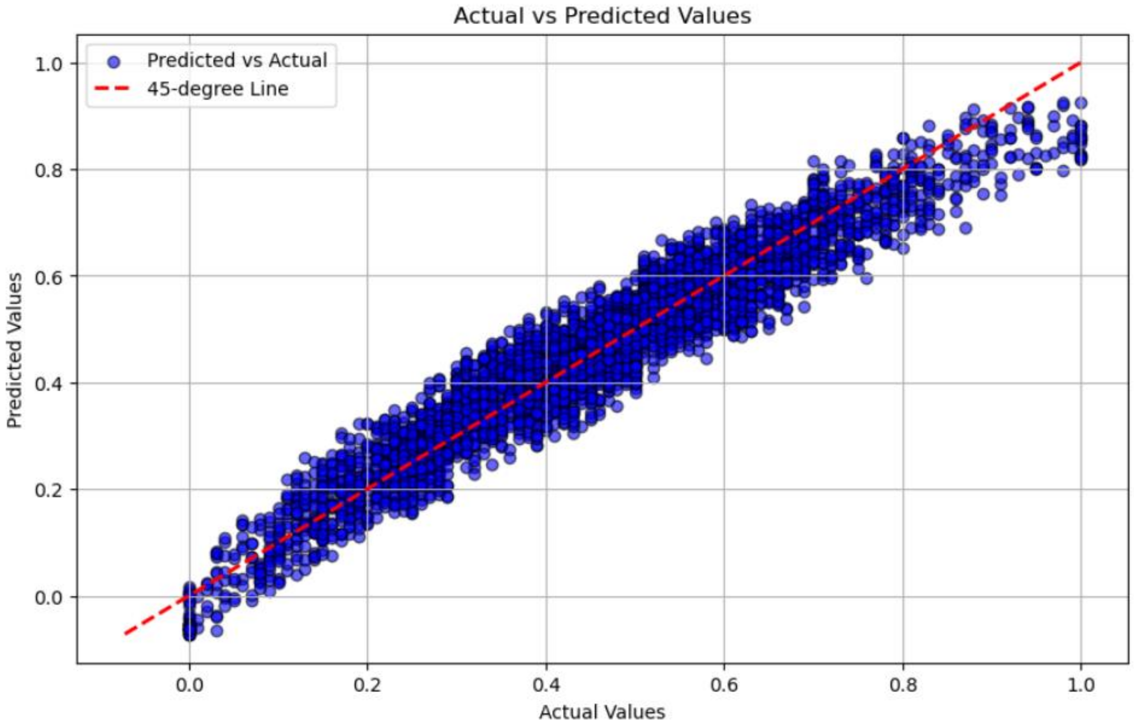


Figure 16:graph showing how accurate the model is.

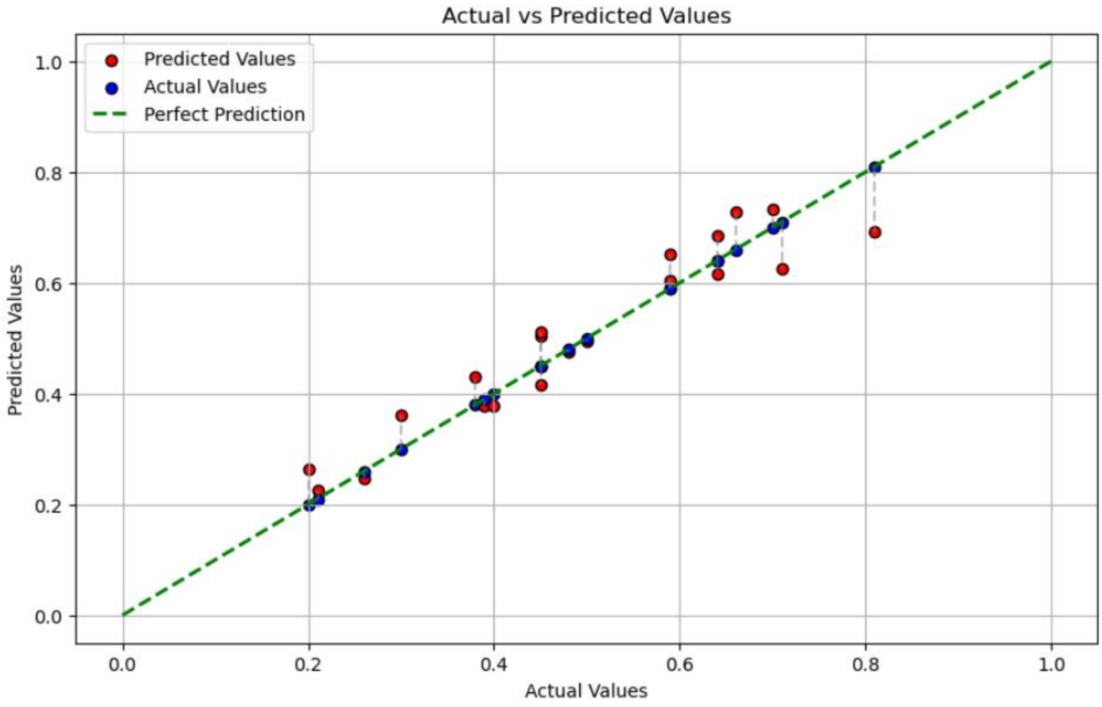
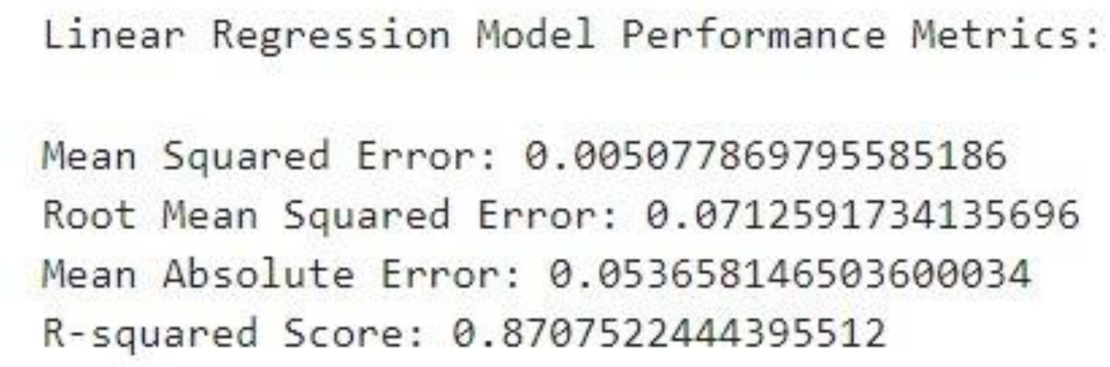


Figure 17:Here we took 20 random samples and plotted their actual and predicted values to show accuracy visually.

Performance metrics when we tried to fill the null valued data with different imputation techniques:

1)Filling with mean:

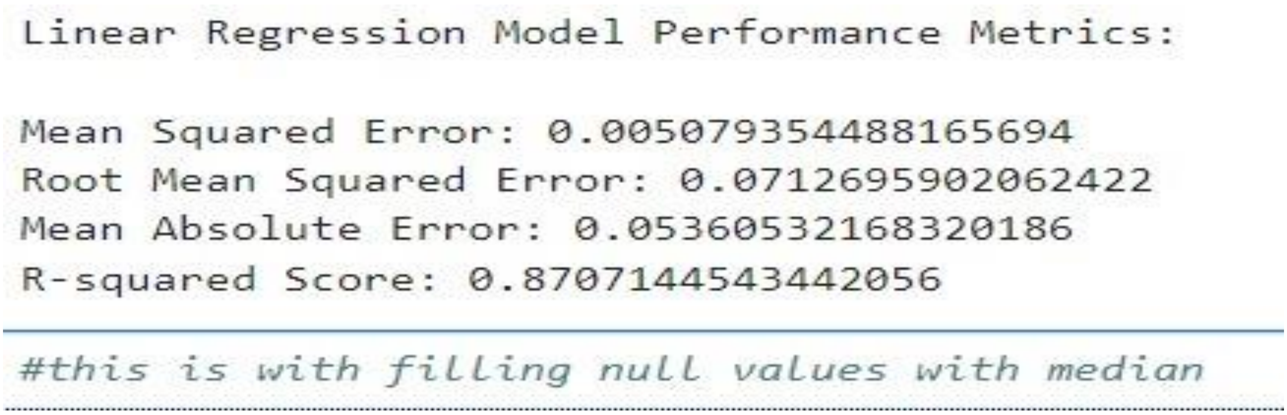


```
Linear Regression Model Performance Metrics:

Mean Squared Error: 0.005077869795585186
Root Mean Squared Error: 0.0712591734135696
Mean Absolute Error: 0.053658146503600034
R-squared Score: 0.8707522444395512
```

Figure 18:Linear Regression Model Performance Metrics (after filling null values with mean)

We filled the null values in both the rows using mean of the data in that column, after dropping the rows which consists null values in Burn rate and proceeded with the same process we have done above, and we got the above metrics for this process

2)Filling with median:

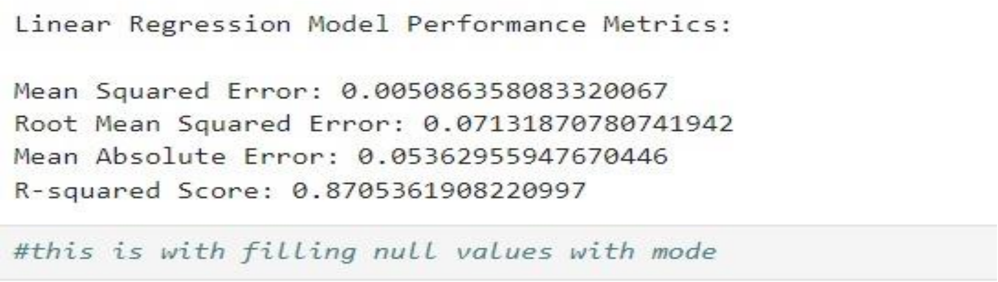


```
Linear Regression Model Performance Metrics:

Mean Squared Error: 0.005079354488165694
Root Mean Squared Error: 0.0712695902062422
Mean Absolute Error: 0.05360532168320186
R-squared Score: 0.8707144543442056

#this is with filling null values with median
```

Figure 19:Linear Regression Model Performance Metrics (after filling null values with median)

We filled the null values in both the rows using median of the data in that column, after dropping the rows which consists null values in Burn rate and proceeded with the same process we have done above, and we got the above metrics for this process

3)Filling with mode:



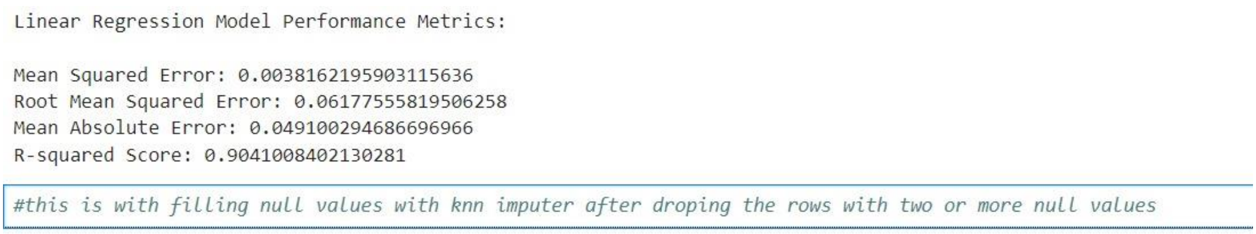
```
Linear Regression Model Performance Metrics:

Mean Squared Error: 0.005086358083320067
Root Mean Squared Error: 0.07131870780741942
Mean Absolute Error: 0.05362955947670446
R-squared Score: 0.8705361908220997
```

```
#this is with filling null values with mode
```

Figure 20:Linear Regression Model Performance Metrics (after filling null values with mode)

We filled the null values in both the rows using mode of the data in that column, after dropping the rows which consists null values in Burn rate and proceeded with the same process we have done above, and we got the above metrics for this process

3)Filling with KNN imputer:



```
Linear Regression Model Performance Metrics:

Mean Squared Error: 0.0038162195903115636
Root Mean Squared Error: 0.06177555819506258
Mean Absolute Error: 0.049100294686696966
R-squared Score: 0.9041008402130281
```

```
#this is with filling null values with knn imputer after droping the rows with two or more null values
```

Figure 21:Linear Regression Model Performance Metrics (after filling null values with KNN imputer after dropping rows with more than one null value i.e. nearly 156 rows are dropped)

We filled the null values in both the rows using KNN imputation of the data in that column, after dropping the rows which consists null values in Burn rate ,the rows which contain null values in more than 1 column excluding Burn rate column and then proceeded with the same process we have done above, and we got the above metrics for this process

But dropping the rows with null values gave us the better accuracy so we had used that method for calculating burn rate (Reasons were explained before itself)

Discussion and Interpretations:

Results Analysis:

1.The high R² value suggests that the model effectively captures the relationship between predictors and burnout.

2. Low mean square error, root mean square error, mean absolute error say that our model works with very small/minimal error.

3.The strong correlation of Designation, Resource allocation, Mental Fatigue Score on Burn rate shows that they are the major contributors of burn rate.

4.And weak correlation of Date of joining says that burn rate nearly doesn’t depend on number of days that the employee worked.

Discrepancies:

Some unexpected findings included lower-than-expected significance of management support. This could be due to subjective nature of the survey responses or sample size limitations.

Possible Explanations:

Discrepancies might arise from biases in survey responses or external factors not captured by the model.

Conclusion:

Summary:

1.The linear regression model provided valuable insights into the predictors of employee burnout.

2.Key predictors include Designation, Resource allocation, Mental Fatigue Score.

Recommendations:

1.Organizations should focus on improving job satisfaction, Resource allocation, create a healthy entertaining environment at least one day per week to decrease mental fatigue score of employees, and managing workloads to reduce burnout.

2.Further research with a larger sample size and additional variables may enhance model accuracy.

Future Work:

Incorporate more advanced models like polynomial regression or machine learning techniques for better prediction accuracy.

Final Observation:

The linear regression model is the most effective for predicting burnout, according to evaluation metrics. The model has the lowest MSE, RMSE, and MAE, indicating superior accuracy and prediction. The model has the greatest R-squared value, suggesting a strong fit to the data and explaining a greater share of the variance in the target variable.

REFERENCES:

1.Matplotlib documentation — Matplotlib 3.9.1 documentation

2. pandas documentation — pandas 2.2.2 documentation (pydata.org)

3. NumPy Documentation

4. scikit-learn: machine learning in Python — scikit-learn 1.5.1 documentation

5.https://www.kaggle.com/datasets/vijaysubhashp/employee-burnout-prediction

6. Home - Edunet Foundation

7. edunetfoundation.org

PROJECT REPOSITORY LINK:

https://github.com/phantom0345/Burn-out-rate

*Refer to this repository link for more explanations, it may find useful to solve your doubts as there is a ppt presentation of this project in that repository so please try to check it.

MAIL:

yaswanthsai_mannem@srmap.edu.in