**Contents**

Abstract

Introduction

Requirements

Algorithm

Coding

Testing

Conclusion

Bibliography

Step-by-Step Instructions

**Abstract**

This project focuses on leveraging machine learning techniques to enhance HR analytics by predicting employee attendance. **The primary goal is to develop a predictive model using Python that can accurately determine whether an employee will be present or absent on a given day based on various metrics.** These metrics include the number of pets and children an employee has, their transportation options, travel time, and historical absence data.

To achieve this, we collected and analysed a dataset containing detailed records of employee attendance and relevant features. The data pre-processing steps included handling missing values, dealing with outliers, and normalizing the data to ensure optimal performance of the machine learning models. Several machine learning algorithms were employed and compared, including logistic regression, decision trees, and random forests, to identify the most effective model for predicting absenteeism.

The analysis revealed that factors such as travel time and the number of children significantly influence employee attendance patterns. The most effective model demonstrated a high level of accuracy in predicting absenteeism, enabling HR departments to anticipate workforce availability with greater precision.

**Uses of the Project**

1. **Proactive Workforce Management:**

By predicting employee absenteeism, HR departments can proactively manage staffing levels, ensuring that there are always enough employees available to meet operational needs.

**2. Resource Planning:**

Accurate predictions of employee attendance allow for better planning of resources. HR can allocate tasks and projects more effectively, minimizing the impact of unexpected absences on productivity.

**3. Targeted Interventions:**

Identifying patterns in absenteeism can help HR departments implement targeted interventions. For example, if travel time is a significant factor, HR could explore flexible working hours or remote work options for employees with long commutes.

**4. Improved Employee Support:**

By understanding the factors that contribute to absenteeism, organizations can offer better support to their employees. This could include providing childcare support or transportation options to those who need it.

**5. Enhanced Decision-Making:**

Integrating machine learning models into HR analytics provides data-driven insights that enhance overall decision-making. This allows HR to move from reactive to proactive strategies in managing employee attendance.

**6. Cost Savings:**

Reducing unexpected absenteeism through accurate predictions can lead to significant cost savings for the organization. Fewer disruptions in workflow mean less reliance on temporary staffing or overtime, which can be costly.

**Conclusion**

This project demonstrates the significant potential of machine learning in improving HR analytics and workforce management. By accurately predicting employee attendance, organizations can enhance their operational efficiency, better support their employees, and implement effective resource planning strategies. This not only optimizes productivity but also contributes to a more supportive and responsive workplace environment.

**Current System vs. Project Benefits**

**Current System:**

* Relies on historical attendance data and manual adjustments.
* Often reactive, leading to last-minute staffing issues.
* Limited ability to identify and address specific absenteeism patterns.
* Decisions based on intuition and incomplete data.
* Potentially higher costs due to unplanned absences and emergency staffing.

**Advantages of the project:**

* Proactive management of workforce availability, reducing operational disruptions.
* Better resource planning and allocation, enhancing productivity and project management.
* Ability to implement targeted interventions to address root causes of absenteeism.
* Improved employee support and satisfaction, leading to lower absenteeism rates.
* Data-driven decision-making for more effective HR management.
* Significant cost savings by reducing the reliance on temporary staffing and overtime.

**Requirements**

**Software Requirements**:

1. Jupyter Notebook:

- Description: Jupyter Notebook is a web-based interactive computing platform that allows users to create and share documents containing live code, equations, visualizations, and narrative text.

- Installation:

- It is recommended to utilize Anaconda Distribution which is an open-source distribution that simplifies package management and deployment. The distribution includes Jupyter Notebook, Python, and other commonly used packages for data science.

- Alternatively, Jupyter Notebook can be installed using pip:

```

pip install notebook

```

- Usage: Jupyter Notebook can be accessed through a web browser by running `jupyter notebook` in the command line.

2. Python:

- Description: Python is a versatile and powerful programming language widely used in data science, machine learning, and various other domains.

- Installation:

- Install Python using Anaconda Distribution or download the official Python release from the Python Software Foundation website.

- Verify the installation by running `python --version` in the command line.

### Hardware Requirements:

1. Operating System:

- Jupyter Notebook and Python are compatible with multiple operating systems including Windows, macOS, and Linux.

2. Processor:

- A multi-core processor (e.g., Intel Core i5 or higher) is recommended for faster computation.

3. RAM:

- Minimum 8GB RAM is recommended for handling data-intensive tasks efficiently.

4. Storage:

- Adequate storage space to store datasets and output files generated during the data analysis process.

5. Graphics Card (Optional):

- A dedicated graphics card is beneficial for running complex visualizations efficiently.

6. Internet Connection:

- A stable internet connection is required for installing packages, accessing online resources, and collaborating on Jupyter Notebook.

By meeting these software and hardware requirements, you can establish a robust environment for data analysis, model building, and experimentation using Jupyter Notebook and Python. This setup will enable you to effectively work on following data science project and explore various machine learning algorithms.

**Definitions:**

**One-Hot Encoding:**

Definition: One-hot encoding transforms categorical variables into a binary format, creating dummy variables for each category.

Example: In the "Color" feature, each color category like Red, Green, and Blue is represented by a binary vector. This method prevents ordinal assumptions.

**Binary Classifier:**

Definition: A binary classifier is a machine learning model that categorizes input instances into one of two classes.

Example: In email spam detection, the classifier labels emails as spam (1) or not spam (0) using features like keywords and sender information.

**Machine Learning:**

Definition: Machine learning empowers systems to learn patterns from data without explicit programming, improving with experience.

Example: A movie recommendation system learns user preferences to recommend movies based on similarities with other users' viewing habits.

**Machine Learning Model:**

Definition: A machine learning model is a mathematical representation trained on data to predict outcomes or classify instances.

Example: A Decision Tree model predicts customer churn by considering factors like tenure, usage patterns, and customer complaints.

**Feature Engineering:**

Definition: Feature engineering involves creating, transforming, or selecting features to enhance a model's predictive performance.

Example: Creating a new feature "Total Income" by combining "Salary" and "Bonus" features can provide more predictive power for income-related predictions.

**Feature Selection:**

Definition: Feature selection aims to choose relevant features while excluding irrelevant ones to optimize model performance.

Example: Employing Recursive Feature Elimination, a feature selection technique, to identify and retain the most impactful features for model training.

**Data Cleaning:**

Definition: Data cleaning focuses on detecting and rectifying errors in data to ensure its accuracy and reliability for analysis.

Example: Removing duplicate entries in a customer dataset is a vital data cleaning procedure to prevent inaccuracies and misleading analysis results.

**Standardization:**

Definition: Standardization rescales numerical features to have a mean of 0 and a standard deviation of 1, ensuring a consistent scale across features.

Example: By standardizing the "Height" feature, discrepancies in numerical ranges are eliminated, leading to more uniform comparisons.

**Data Pre-processing:**

Definition: Data pre-processing involves transforming raw data into a suitable format for analysis by handling missing values, outliers, and standardizing features.

Example: Imputing missing values and normalizing numeric features are common pre-processing steps to prepare data for model training.

**Model Building:**

Definition: Model building is the process of training a machine learning algorithm on data to develop a predictive or classification model.

Example: Training a Linear Regression model on housing data, using features like size and location to predict house prices, exemplifies the model-building process.

**Python:**

Definition: Python is a versatile programming language known for its readability and simplicity, widely used in data analysis, machine learning, and web development.

Example: Writing a Python script to calculate the sum of numbers in a list showcases Python's ease of use and expressiveness in handling tasks efficiently.

**IDE (Integrated Development Environment):**

Definition: An IDE offers a comprehensive interface for software development, combining tools for editing, debugging, and managing projects.

Example: Using PyCharm, a popular Python IDE, developers can write code, debug errors, and run Python programs efficiently within an integrated environment.

**Function in Python:**

Definition: Functions in Python are reusable blocks of code that perform specific tasks, enhancing code modularity and reusability.

Example: Defining a function calculate\_area to compute the area of a rectangle with given length and width helps streamline and organize code for easy reuse and maintenance.

**Pre-processing**

If have a look at our data we can identify the following columns/features as follows:

Describing the available data, we have for our Analysis.

1. ID: Unique identification numbers assigned to employees.
2. Reason for absence: Codes representing the reasons for employee absences.
3. Month of absence: Months in which the absences occurred.
4. Day of the week: Days of the week when the absences took place.
5. Seasons: Categorization of absences based on the seasons.
6. Transportation expense: Cost of commuting to work for employees.
7. Distance from Residence to Work: Distance between residence and workplace.
8. Service time: Duration of employment with the company.
9. Age: Age of the employees.
10. Work load Average/day: Average daily workload of the employees.
11. Hit target: Achievement of work targets within a specific timeframe.
12. Disciplinary failure: Indication of any previous disciplinary issues.
13. Education: Educational level or qualifications of employees.
14. Son: Number of sons the employees have.
15. Social drinker: Whether the employee consumes alcohol socially.
16. Social smoker: Identification of employees who smoke socially.
17. Pet: Presence and number of pets owned by employees.
18. Weight: The weight of the employees.
19. Height: The height of the employees.
20. Body mass index (BMI): Calculated body mass index of the employees.
21. Absenteeism time in hours: Duration of absence in hours for each employee.

**Feature Selection Justification Report**

In our data analysis project focused on studying employee absenteeism, we have carefully selected specific features that are deemed crucial in understanding and predicting absenteeism patterns within the organization. Below is a detailed rationale behind the selection of the chosen features:

Reason for Absence: The reason behind an employee's absence from work is a fundamental factor influencing absenteeism trends. By including this feature, we aim to analyze and categorize the diverse reasons that lead to employee absences, providing valuable insights into the underlying causes.

Month of Absence: Considering the month in which absences occur is vital for recognizing any seasonal patterns in absenteeism. Understanding if there are specific months with higher or lower absenteeism rates can aid in implementing effective management strategies.

Day of the Week: The day of the week of an absence can reveal if there are certain days more prone to absenteeism. This information helps in scheduling tasks efficiently and addressing any issues related to specific weekdays.

Transportation Expense: The cost incurred by employees for transportation to work impacts their daily routines and can influence absenteeism. Monitoring this feature enables us to assess the relationship between transportation costs and absenteeism behavior.

Distance from Residence to Work: The distance employees travel from their residence to the workplace plays a role in their commute stress and fatigue levels. Analyzing this feature assists in understanding how distance affects absenteeism rates.

Workload Average/Day: The average daily workload of employees is a key determinant of their work-related stress and productivity levels. Including this feature allows us to examine the impact of workload on absenteeism tendencies.

Education: The educational background of employees can influence their approach towards work and may impact absenteeism patterns. This feature helps in evaluating if there are correlations between education level and absenteeism.

Son: The number of sons an employee has can factor into their personal responsibilities and potential reasons for absence. This feature provides insights into how family dynamics can relate to absenteeism occurrences.

Pet: Pets owned by employees can influence their work-life balance and stress levels, affecting absenteeism. By incorporating this feature, we aim to explore the interplay between pet ownership and absenteeism rates.

Body Mass Index (BMI): Employee BMI is indicative of their overall health status, which can be linked to absenteeism trends. Monitoring this feature allows us to investigate the impact of health on absenteeism occurrences.

Excessive Absenteeism: This derived feature serves as the target variable, indicating instances of excessive absenteeism that warrant attention. By including this feature, we aim to build predictive models to identify and address excessive absenteeism cases proactively.

**Data Pre-processing Report**

In the data pre-processing phase of our project, the following operations have been performed to prepare the dataset for model building and analysis:

1. **Removal of Null Values:**

Null values have been removed from the dataset to ensure data integrity and eliminate missing information that could adversely affect model performance and results.

1. **Creation of Excessive Absenteeism Flag:**

The 'Absenteeism time in hours' column has been used to determine excessive absenteeism based on a predefined threshold (median value of 3 hours).

If the absenteeism time is greater than the median (3 hours), the 'Excessive Absenteeism' flag is set to 1; otherwise, it is set to 0. This binary classification simplifies the prediction of excessive absenteeism.

1. **Dropping 'Absenteeism time in hours' Column:**

The 'Absenteeism time in hours' column has been dropped from the dataset as it was used to derive the 'Excessive Absenteeism' flag and is no longer needed for model training.

1. **Categorization of Reasons for Absence:**

The 'Reason for absence' column, containing multiple reasons, has been transformed into four distinct buckets using one-hot encoding to facilitate analysis and model training.

Dummies have been created for each reason category, and four consolidated reasons ('Reason\_1' to 'Reason\_4') have been formed based on logical grouping.

1. **Concatenation of Data**:

The newly derived reason categories have been concatenated with the original dataset along with other relevant columns for a comprehensive feature set.

Column names have been defined for clarity, aligning with the transformed data structure.

1. **Feature Adjustment:**

The 'Education' feature has been modified to a binary representation for simplification, mapping different education levels to 0 (basic education) or 1 (higher education).

Data columns have been reordered for better organization and to align with the defined order of features for model training.

1. **Exporting Processed Data:**

The processed dataset has been saved into a CSV file named 'processed\_data\_.csv' for future reference and model building.

The final pre-processed data is stored in the 'self.data' variable for subsequent analysis and model development.

Model Building :

1. Reading the Preprocessed Data:

```python

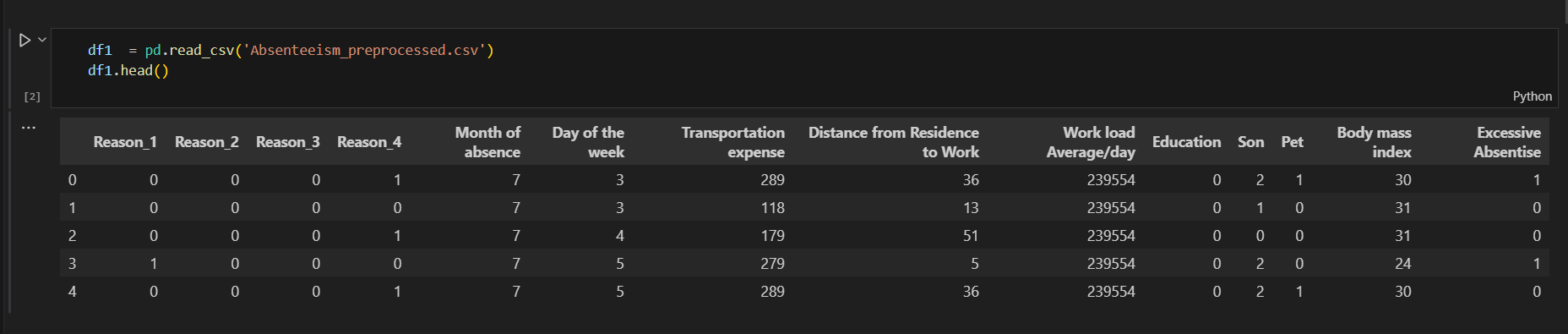
df1 = pd.read\_csv('Absenteeism\_preprocessed.csv')

df1.head()

df1.info()

```

The code reads the preprocessed data from Absenteeism\_preprocessed.csv' file and displays the first few rows of the dataset as well as its information showing the non-null counts and data types of each column.



2. Target and Features Selection:

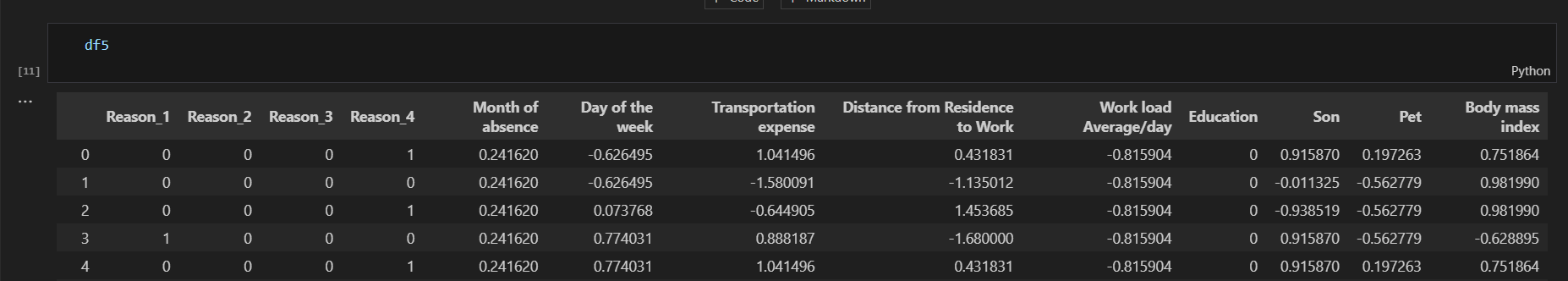
```python

target = df1.iloc[:,-1]

features = df1.iloc[:,:-1]

```

This code separates the target variable ('Excessive Absentise') and the features from the preprocessed dataset for further use in model training.



3. Scaling the Features:

```python

scaler = StandardScaler()

scaler.fit(df5[columns\_to\_scale])

df5[columns\_to\_scale] = scaler.transform(df5[columns\_to\_scale])

```

The selected numeric columns are scaled using the StandardScaler, except for the 'Reasons' and 'Education' columns. This scaling process standardizes the data to have a mean of 0 and a standard deviation of 1, which is important for many machine learning models.

4. Train-Test Split:

```python

x\_train, x\_test, y\_train, y\_test = train\_test\_split(df5, target, train\_size=0.8, random\_state=20)

print(f'x\_train size = {x\_train.shape}')

print(f'x\_test size = {x\_test.shape}')

print(f'y\_train size = {y\_train.shape}')

print(f'y\_test size = {y\_test.shape}')

```

The dataset is split into training and testing sets, with 80% of the data used for training the model and 20% for testing. The sizes of the resulting train and test sets are printed for verification.

5. Training the Logistic Regression Model:

```python

reg = LogisticRegression()

reg.fit(x\_train, y\_train)

reg.score(x\_train, y\_train)

```

A Logistic Regression model is instantiated and trained using the training data. The accuracy score on the training data is also computed, showing the proportion of correct classifications by the model.

6. Understanding Intercepts and Coefficients:

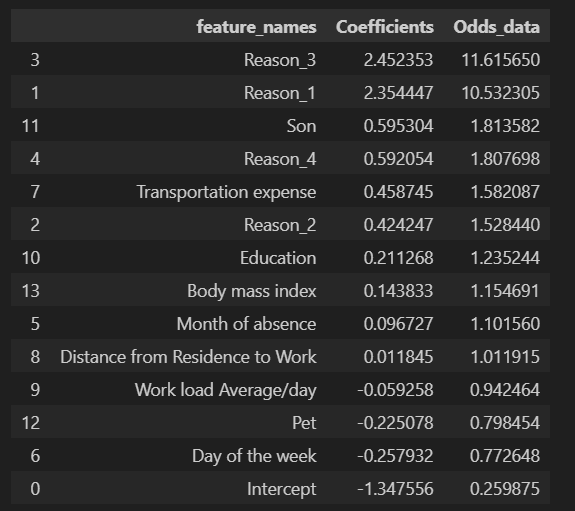
- \*\*Intercept\*\*: The intercept reflects the estimated constant term of the logistic regression model when all other features are set to zero.

- \*\*Coefficients\*\*: These represent the change in the log-odds of the target (excessive absenteeism) for a one-unit change in the corresponding feature.

The coefficient values for each feature indicate the strength and direction of their influence on the target variable, while the intercept represents the baseline log-odds of the target variable when all feature values are zero.

7. Summary Table:

The table lists the features, their corresponding coefficients, and the calculated odds based on the coefficients. The coefficients are then interpreted in terms of odds to understand their impact on excessive absenteeism. The higher the odds value, the stronger the impact on the likelihood of excessive absenteeism.



This analysis provides a comprehensive insight into the logistic regression model, its training, and the interpretability of the coefficients to assess feature importance and their impact on the prediction of excessive absenteeism.

**Logistic Regression Overview:**

Definition: Logistic Regression is a statistical method used for binary classification tasks, predicting the probability of a binary outcome based on input features.

Function: It estimates the probability that a given instance belongs to a particular class using a logistic function to map predictions between 0 and 1.

Output: Despite its name, logistic regression is commonly used for classification, where it predicts the likelihood of an instance belonging to a specific class.

Selection Criteria for Logistic Regression:

Logistic Regression can be a suitable choice under certain circumstances:

Binary Outcome: When the target variable is binary, i.e., having two classes or categories.

Interpretability: When interpretability of results is important due to its clear analysis of the impact of features on the target variable.

Linear Relationship: When the relationship between independent variables and the log-odds of the binary outcome can be assumed to be linear.

Limited Overfitting: Logistic regression tends to have limited overfitting concerns, making it suitable for datasets with limited training samples.

Feature Importance: When you want to interpret the importance of different features affecting the target variable.

Application in Excessive Absenteeism Prediction:

Predictive Context: In your case of predicting "Excessive Absenteeism," where you have a binary outcome (Excessive Absenteeism or not), Logistic Regression fits well.

Features Considered:

Categorical Features (Reasons): Reason\_1, Reason\_2, Reason\_3, Reason\_4.

Numerical Features: Month of absence, Day of the week, Transportation expense, Distance from Residence to Work, Workload Average/day, Education, Son, Pet, Body mass index.

Model Interpretation: Logistic Regression provides coefficients for each feature, enabling you to interpret the impact of reasons and other features on the likelihood of excessive absenteeism.

Predictive Power: By analyzing these selected features, Logistic Regression can estimate the probability of an individual exhibiting excessive absenteeism based on the input factors.

Conclusion:

Accuracy : 73.28 %