

## Data Collection and Preprocessing Phase

Date	20 June 2024
Team ID	739849
Project Title	Doctors Annual Salary Prediction
Maximum Marks	6 Marks

### Data Exploration and Preprocessing Template

Identifies data sources, assesses quality issues like missing values and duplicates, and implements resolution plans to ensure accurate and reliable analysis.

Section	Description
Basic statistics, dimensions, and structure of the data.	<pre>import pandas as pd  # Load dataset url = "https://example.com/doctors_salary_data.csv" df = pd.read_csv(url)  # Display basic statistics print(df.describe())  # Display data structure print(df.info())</pre>
Exploration of individual variables (mean, median, mode, etc.)	<pre># Calculate mean, median, and mode for the 'salary' column mean_salary = df['salary'].mean() median_salary = df['salary'].median() mode_salary = df['salary'].mode()  print(f"Mean Salary: {mean_salary}") print(f"Median Salary: {median_salary}") print(f"Mode Salary: {mode_salary}")</pre>
Relationships between two variables (correlation, scatter plots)	<pre>import seaborn as sns import matplotlib.pyplot as plt  # Scatter plot between 'experience' and 'salary'</pre>

	<pre>sns.scatterplot(x='experience', y='salary', data=df) plt.title('Experience vs Salary') plt.show()  # Correlation between 'experience' and 'salary' correlation = df['experience'].corr(df['salary']) print(f"Correlation between experience and salary: {correlation}")</pre>
Patterns and relationships involving multiple variables.	<pre># Pairplot to show relationships between multiple variables sns.pairplot(df) plt.show()</pre>
Identification and treatment of outliers.	<pre># Box plot to identify outliers in 'salary' sns.boxplot(df['salary']) plt.title('Box plot of Salary') plt.show()  # Remove outliers (example: values above the 95th percentile) upper_limit = df['salary'].quantile(0.95) df = df[df['salary'] &lt;= upper_limit]</pre>
<b>Data Preprocessing Code Screenshots</b>	
Code to load the dataset into the preferred environment (e.g., Python, R)	<pre># Load dataset url = "https://example.com/doctors_salary_data.csv" df = pd.read_csv(url)</pre>
Code for identifying and handling missing values	<pre># Check for missing values missing_values = df.isnull().sum() print(missing_values)  # Fill missing values with median df.fillna(df.median(), inplace=True)</pre>
Code for transforming variables (scaling, normalization)	<pre>from sklearn.preprocessing import StandardScaler  # Standardize the 'salary' column scaler = StandardScaler() df['salary_scaled'] = scaler.fit_transform(df[['salary']])</pre>
Code for creating new features or modifying existing ones.	<pre># Create a new feature 'experience_squared' df['experience_squared'] = df['experience'] ** 2</pre>

Code to save the cleaned and processed data for future use.

```
# Save the processed data to a new CSV file
df.to_csv('processed_doctors_salary_data.csv', index=False)
```