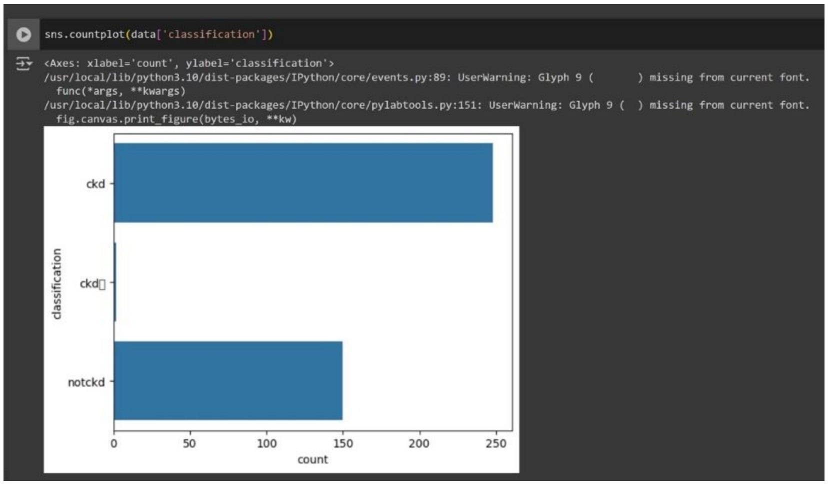
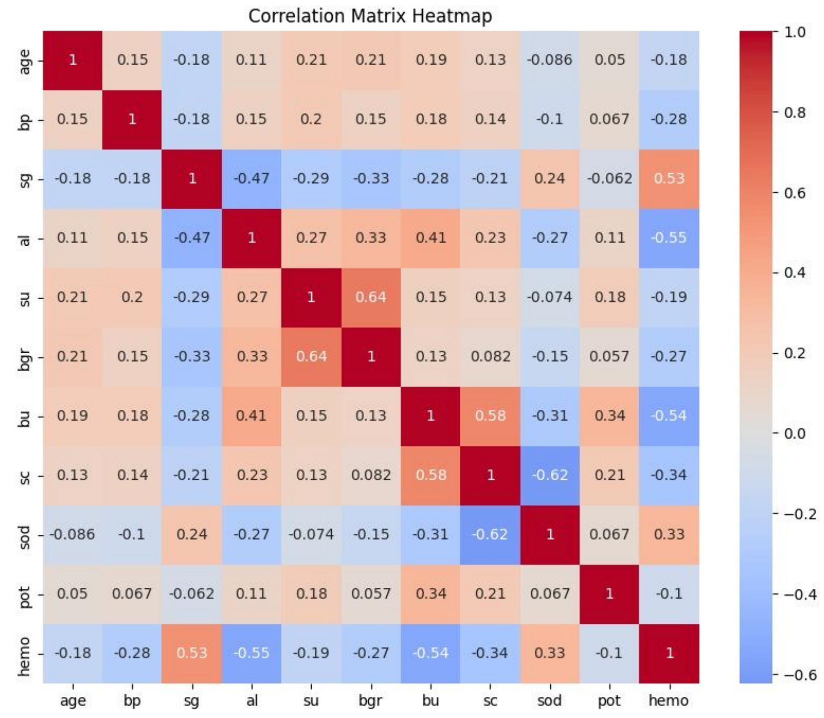
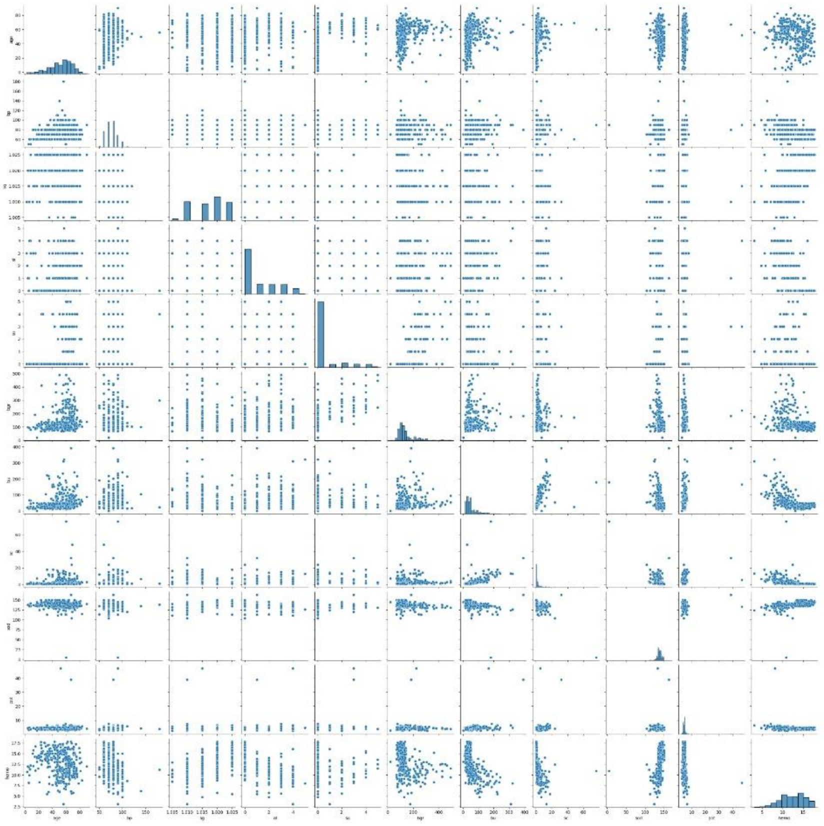
Data Collection and Preprocessing Phase

Date	08 July 2024
Team ID	SWTID1720174640
Project Title	Early Prediction of Chronic Kidney Disease
Maximum Marks	6 Marks

Data Exploration and Preprocessing Template

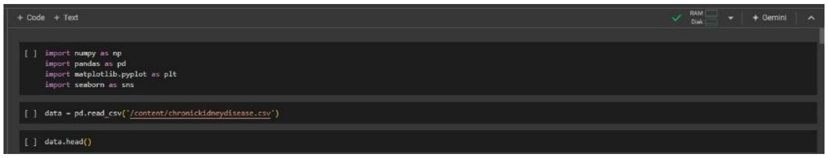
Identifies data sources, assesses quality issues like missing values and duplicates, and implements resolution plans to ensure accurate and reliable analysis.

Section	Description
Data Overview	The dataset `chronickidneydisease.csv` has 400 rows and 26 columns, including both numerical and categorical data. It contains various patient attributes like age, blood pressure, specific gravity, albumin, sugar, red blood cells, hemoglobin, and more, along with a classification label indicating chronic kidney disease status. There are missing values in several columns. Basic statistics show that attributes like age, blood pressure, and others have varying ranges and distributions.
Univariate Analysis	 sns.countplot(data['classification']) <Axes: xlabel='count', ylabel='classification'> /usr/local/lib/python3.10/dist-packages/IPython/core/events.py:89: UserWarning: Glyph 9 ( ) missing from current font. func(*args, **kwargs) /usr/local/lib/python3.10/dist-packages/IPython/core/pylabtools.py:151: UserWarning: Glyph 9 ( ) missing from current font. fig.canvas.print_figure(bytes_io, **kw) ckd ckd□ notckd classification 0 50 100 150 200 250 count

<table>
<tr><td>Bivariate Analysis</td><td>
</td></tr>
<tr><td>Multivariate Analysis</td><td></td></tr>
</table>

Data Preprocessing Code Screenshots

Loading Data

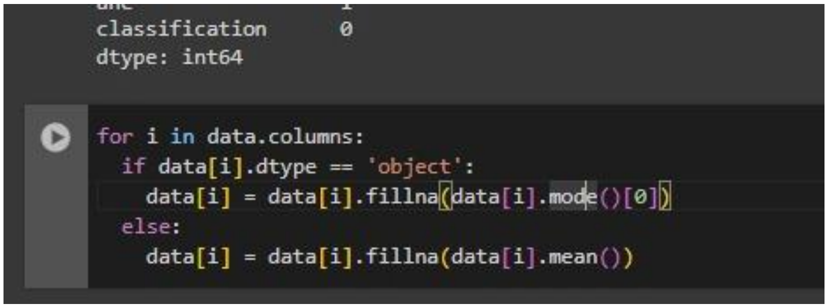


```
[ ] import numpy as np
    import pandas as pd
    import matplotlib.pyplot as plt
    import seaborn as sns

[ ] data = pd.read_csv('/content/chronickidneydisease.csv')

[ ] data.head()
```

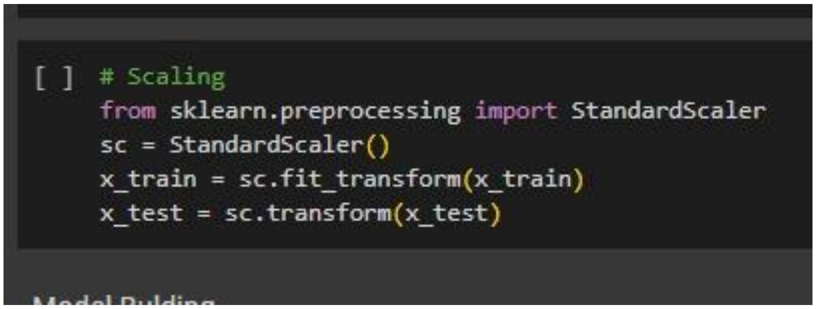
Handling Missing Data



```
classification     0
dtype: int64

for i in data.columns:
  if data[i].dtype == 'object':
    data[i] = data[i].fillna(data[i].mode()[0])
  else:
    data[i] = data[i].fillna(data[i].mean())
```

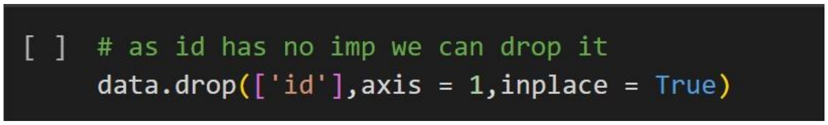
Data Transformation



```
[ ] # Scaling
    from sklearn.preprocessing import StandardScaler
    sc = StandardScaler()
    x_train = sc.fit_transform(x_train)
    x_test = sc.transform(x_test)
```

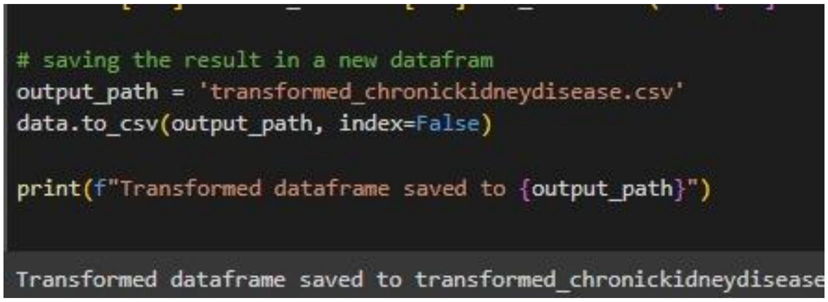
Model Bulding

Feature Engineering



```
[ ] # as id has no imp we can drop it
    data.drop(['id'],axis = 1,inplace = True)
```

Save Processed Data



```
# saving the result in a new datafram
output_path = 'transformed_chronickidneydisease.csv'
data.to_csv(output_path, index=False)

print(f"Transformed dataframe saved to {output_path}")

Transformed dataframe saved to transformed_chronickidneydisease
```