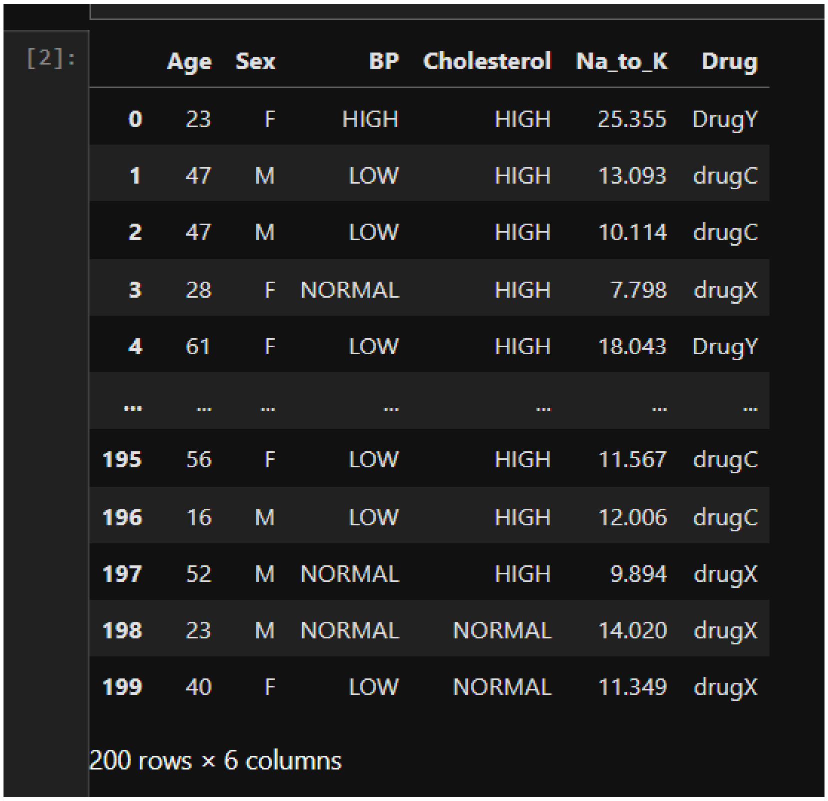
Data Collection and Preprocessing Phase

Date	15 March 2024
Team ID	SWTID1720447482
Project Title	Thyroid Classification
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	Dataset contains 200 rows and 6 columns and each column has values of Age,Sex,BP,Cholesterol,Na_to_K,Drug.

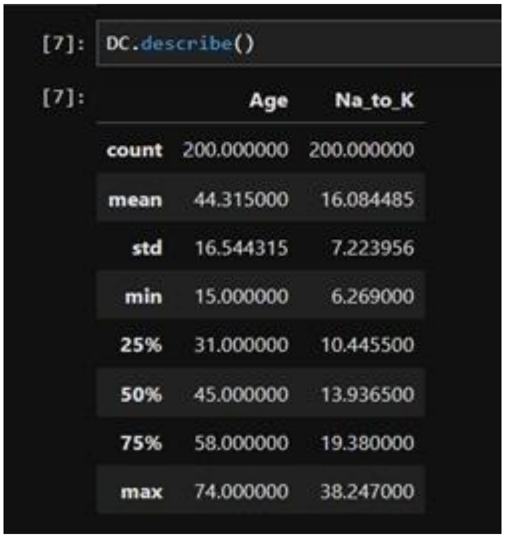
<table>
<tr>
<td></td>
<td>


[2]:

<table>
<tr><th></th><th>Age</th><th>Sex</th><th>BP</th><th>Cholesterol</th><th>Na_to_K</th><th>Drug</th></tr>
<tr><td>0</td><td>23</td><td>F</td><td>HIGH</td><td>HIGH</td><td>25.355</td><td>DrugY</td></tr>
<tr><td>1</td><td>47</td><td>M</td><td>LOW</td><td>HIGH</td><td>13.093</td><td>drugC</td></tr>
<tr><td>2</td><td>47</td><td>M</td><td>LOW</td><td>HIGH</td><td>10.114</td><td>drugC</td></tr>
<tr><td>3</td><td>28</td><td>F</td><td>NORMAL</td><td>HIGH</td><td>7.798</td><td>drugX</td></tr>
<tr><td>4</td><td>61</td><td>F</td><td>LOW</td><td>HIGH</td><td>18.043</td><td>DrugY</td></tr>
<tr><td>...</td><td>...</td><td>...</td><td>...</td><td>...</td><td>...</td><td>...</td></tr>
<tr><td>195</td><td>56</td><td>F</td><td>LOW</td><td>HIGH</td><td>11.567</td><td>drugC</td></tr>
<tr><td>196</td><td>16</td><td>M</td><td>LOW</td><td>HIGH</td><td>12.006</td><td>drugC</td></tr>
<tr><td>197</td><td>52</td><td>M</td><td>NORMAL</td><td>HIGH</td><td>9.894</td><td>drugX</td></tr>
<tr><td>198</td><td>23</td><td>M</td><td>NORMAL</td><td>NORMAL</td><td>14.020</td><td>drugX</td></tr>
<tr><td>199</td><td>40</td><td>F</td><td>LOW</td><td>NORMAL</td><td>11.349</td><td>drugX</td></tr>
</table>

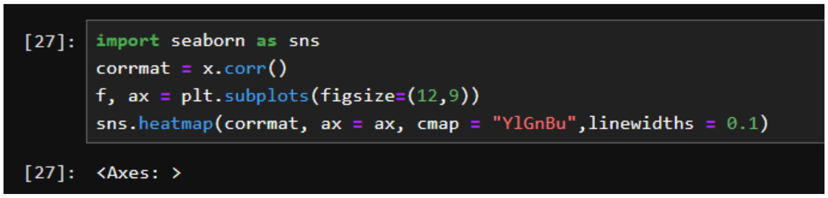
200 rows × 6 columns
</td>
</tr>
<tr>
<td>Univariate Analysis</td>
<td>
The table summarizes two numeric variables, likely columns from a dataset, labeled “Age” and “Na_to_K”.

- The table shows the two variables' counts, means, standard deviations, minimums, maximums, and quartiles (25th and 75th percentiles).
- There are 200 data points according to the count for both variables.
- The average age is 44.3 years old with a standard deviation of 16.5 years. The minimum age in the dataset is 15 years old and the maximum is 74 years old.
- The average Na_to_K value is 16.08 with a standard deviation of 7.22. The minimum value is 6.27 and the maximum value is 38.25.
</td>
</tr>
</table>



```
[7]:  DC.describe()

[7]:             Age     Na_to_K
      count  200.000000  200.000000
       mean   44.315000   16.084485
        std   16.544315    7.223956
        min   15.000000    6.269000
        25%   31.000000   10.445500
        50%   45.000000   13.936500
        75%   58.000000   19.380000
        max   74.000000   38.247000
```

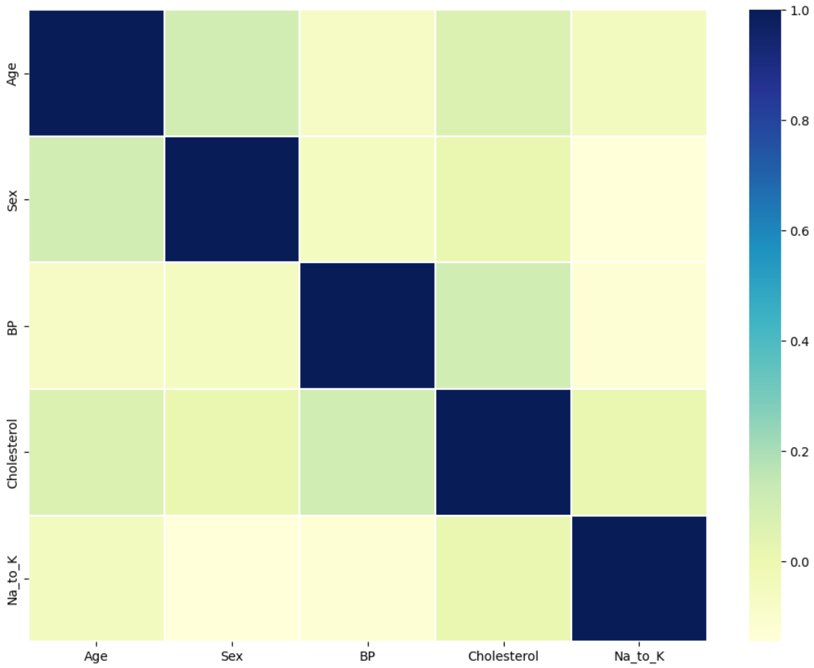
Bivariate Analysis



```
[27]:  import seaborn as sns
       cormat = x.corr()
       f, ax = plt.subplots(figsize=(12,9))
       sns.heatmap(cormat, ax = ax, cmap = "YlGnBu",linewidths = 0.1)

[27]:  <Axes: >
```

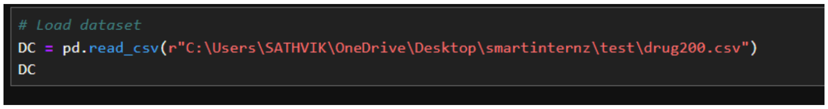
Multivariate Analysis

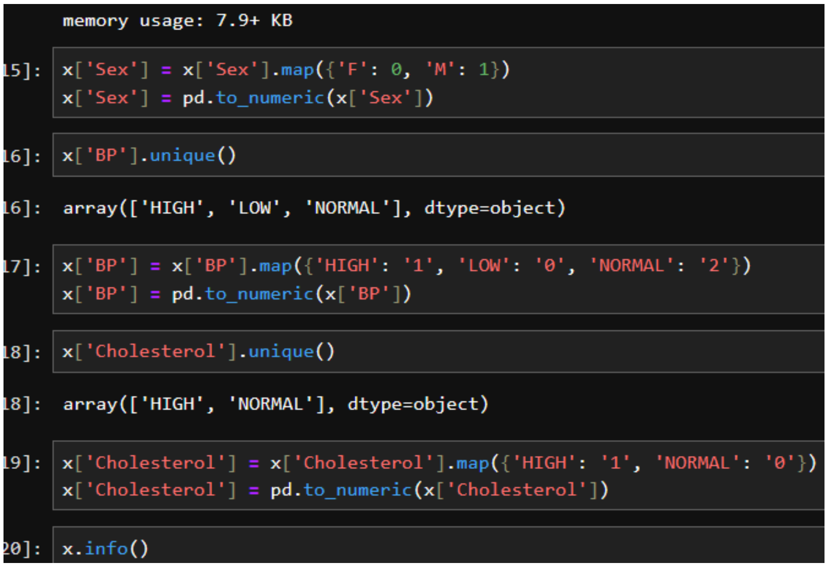
<Axes: >

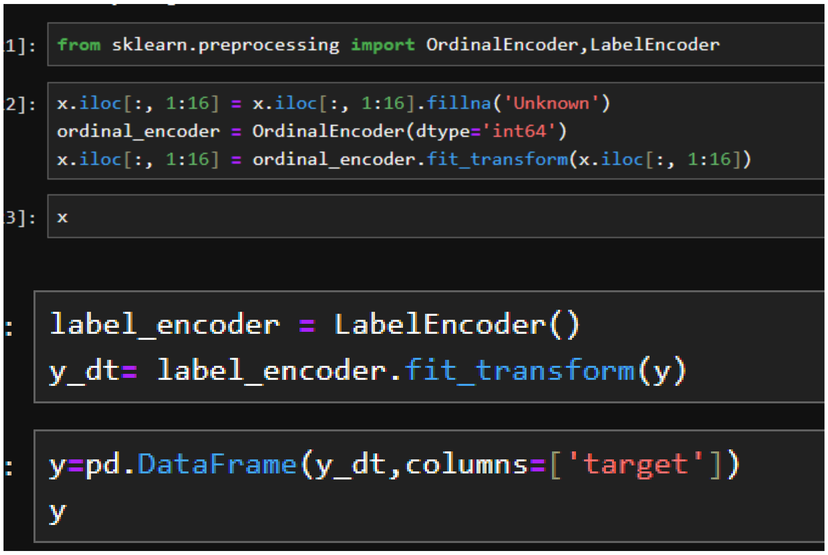


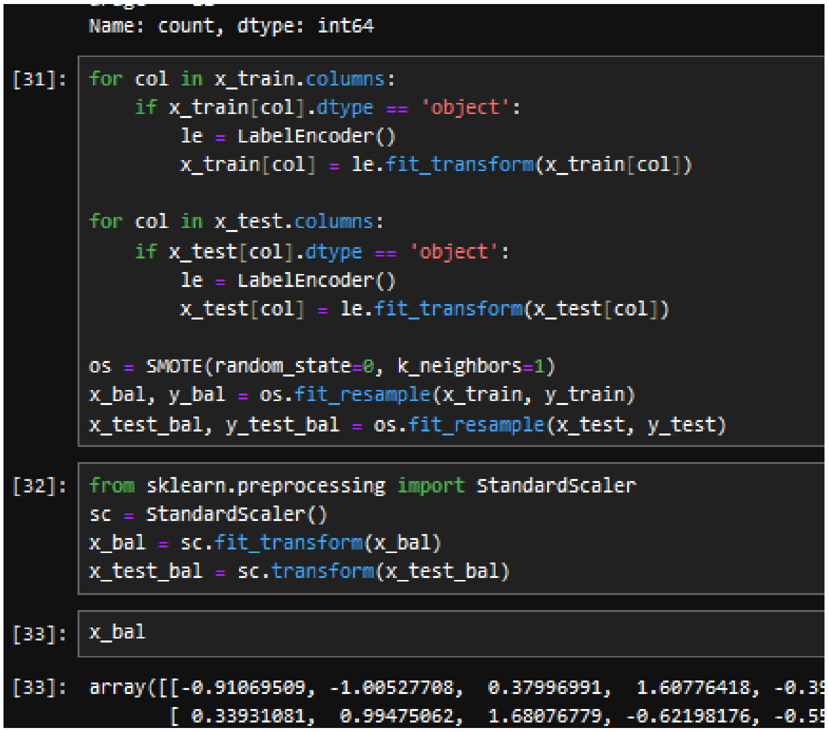
Outliers and Anomalies	Identification and treatment of outliers.

Data Preprocessing Code Screenshots	

Loading Data	
	`# Load dataset`
	`DC = pd.read_csv(r"C:\Users\SATHVIK\OneDrive\Desktop\smartinternz\test\drug200.csv")`
	`DC`

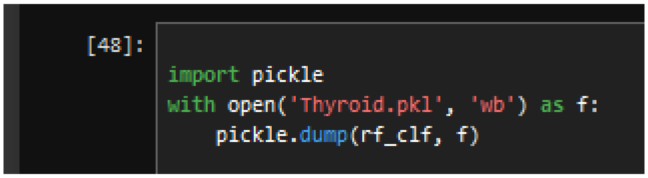
Handling Missing Data	
	`memory usage: 7.9+ KB`
	`15]: x['Sex'] = x['Sex'].map({'F': 0, 'M': 1})`
	`x['Sex'] = pd.to_numeric(x['Sex'])`
	`16]: x['BP'].unique()`
	`16]: array(['HIGH', 'LOW', 'NORMAL'], dtype=object)`
	`17]: x['BP'] = x['BP'].map({'HIGH': '1', 'LOW': '0', 'NORMAL': '2'})`
	`x['BP'] = pd.to_numeric(x['BP'])`
	`18]: x['Cholesterol'].unique()`
	`18]: array(['HIGH', 'NORMAL'], dtype=object)`
	`19]: x['Cholesterol'] = x['Cholesterol'].map({'HIGH': '1', 'NORMAL': '0'})`
	`x['Cholesterol'] = pd.to_numeric(x['Cholesterol'])`
	`20]: x.info()`

Data Transformation	
	`1]: from sklearn.preprocessing import OrdinalEncoder,LabelEncoder`
	`2]: x.iloc[:, 1:16] = x.iloc[:, 1:16].fillna('Unknown')`
	`ordinal_encoder = OrdinalEncoder(dtype='int64')`
	`x.iloc[:, 1:16] = ordinal_encoder.fit_transform(x.iloc[:, 1:16])`
	`3]: x`
	`: label_encoder = LabelEncoder()`
	`y_dt= label_encoder.fit_transform(y)`
	`: y=pd.DataFrame(y_dt,columns=['target'])`
	`y`



```
       Name: count, dtype: int64

[31]:  for col in x_train.columns:
           if x_train[col].dtype == 'object':
               le = LabelEncoder()
               x_train[col] = le.fit_transform(x_train[col])

       for col in x_test.columns:
           if x_test[col].dtype == 'object':
               le = LabelEncoder()
               x_test[col] = le.fit_transform(x_test[col])

       os = SMOTE(random_state=0, k_neighbors=1)
       x_bal, y_bal = os.fit_resample(x_train, y_train)
       x_test_bal, y_test_bal = os.fit_resample(x_test, y_test)

[32]:  from sklearn.preprocessing import StandardScaler
       sc = StandardScaler()
       x_bal = sc.fit_transform(x_bal)
       x_test_bal = sc.transform(x_test_bal)

[33]:  x_bal

[33]:  array([[-0.91069509, -1.00527708,  0.37996991,  1.60776418, -0.3
              [ 0.33931081,  0.99475062,  1.68076779, -0.62198176, -0.5
```

Save Processed Data



```
[48]:  import pickle
       with open('Thyroid.pkl', 'wb') as f:
           pickle.dump(rf_clf, f)
```