

Data Collection and Preprocessing Phase

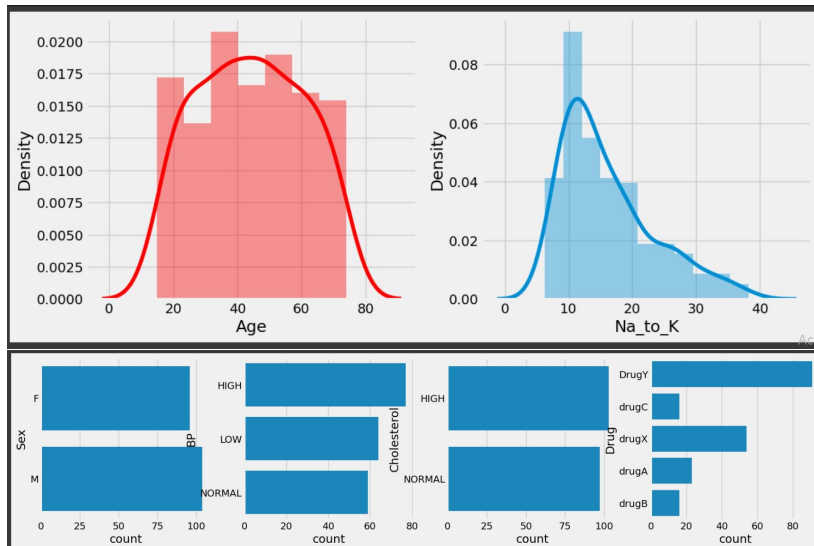
Date	July 2024
Team ID	740664
Project Title	Drug classification using machine learning
Maximum Marks	6 Marks

Data Exploration and Preprocessing Template

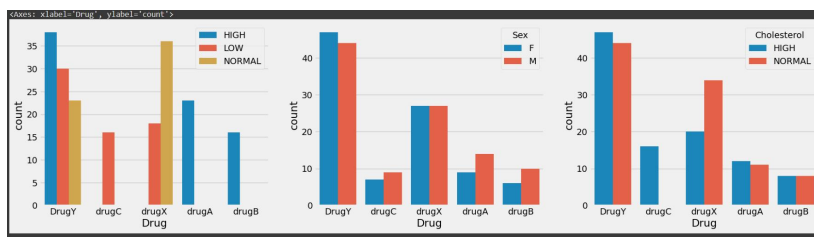
Identifies data sources, assesses quality issues like missing values and duplicates, and implements resolution plans to ensure accurate and reliable analysis.

Section	Description																																																																																				
Data Overview	<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>count</td><td>200.000000</td><td>200</td><td>200</td><td>200</td><td>200.000000</td><td>200</td></tr><tr><td>unique</td><td>NaN</td><td>2</td><td>3</td><td>2</td><td>NaN</td><td>5</td></tr><tr><td>top</td><td>NaN</td><td>M</td><td>HIGH</td><td>HIGH</td><td>NaN</td><td>DrugY</td></tr><tr><td>freq</td><td>NaN</td><td>104</td><td>77</td><td>103</td><td>NaN</td><td>91</td></tr><tr><td>mean</td><td>44.315000</td><td>NaN</td><td>NaN</td><td>NaN</td><td>16.084485</td><td>NaN</td></tr><tr><td>std</td><td>16.544315</td><td>NaN</td><td>NaN</td><td>NaN</td><td>7.223956</td><td>NaN</td></tr><tr><td>min</td><td>15.000000</td><td>NaN</td><td>NaN</td><td>NaN</td><td>6.269000</td><td>NaN</td></tr><tr><td>25%</td><td>31.000000</td><td>NaN</td><td>NaN</td><td>NaN</td><td>10.445500</td><td>NaN</td></tr><tr><td>50%</td><td>45.000000</td><td>NaN</td><td>NaN</td><td>NaN</td><td>13.936500</td><td>NaN</td></tr><tr><td>75%</td><td>58.000000</td><td>NaN</td><td>NaN</td><td>NaN</td><td>19.380000</td><td>NaN</td></tr><tr><td>max</td><td>74.000000</td><td>NaN</td><td>NaN</td><td>NaN</td><td>38.247000</td><td>NaN</td></tr></table>		Age	Sex	BP	Cholesterol	Na_to_K	Drug	count	200.000000	200	200	200	200.000000	200	unique	NaN	2	3	2	NaN	5	top	NaN	M	HIGH	HIGH	NaN	DrugY	freq	NaN	104	77	103	NaN	91	mean	44.315000	NaN	NaN	NaN	16.084485	NaN	std	16.544315	NaN	NaN	NaN	7.223956	NaN	min	15.000000	NaN	NaN	NaN	6.269000	NaN	25%	31.000000	NaN	NaN	NaN	10.445500	NaN	50%	45.000000	NaN	NaN	NaN	13.936500	NaN	75%	58.000000	NaN	NaN	NaN	19.380000	NaN	max	74.000000	NaN	NaN	NaN	38.247000	NaN
		Age	Sex	BP	Cholesterol	Na_to_K	Drug																																																																														
	count	200.000000	200	200	200	200.000000	200																																																																														
	unique	NaN	2	3	2	NaN	5																																																																														
	top	NaN	M	HIGH	HIGH	NaN	DrugY																																																																														
	freq	NaN	104	77	103	NaN	91																																																																														
	mean	44.315000	NaN	NaN	NaN	16.084485	NaN																																																																														
	std	16.544315	NaN	NaN	NaN	7.223956	NaN																																																																														
	min	15.000000	NaN	NaN	NaN	6.269000	NaN																																																																														
	25%	31.000000	NaN	NaN	NaN	10.445500	NaN																																																																														
	50%	45.000000	NaN	NaN	NaN	13.936500	NaN																																																																														
	75%	58.000000	NaN	NaN	NaN	19.380000	NaN																																																																														
max	74.000000	NaN	NaN	NaN	38.247000	NaN																																																																															

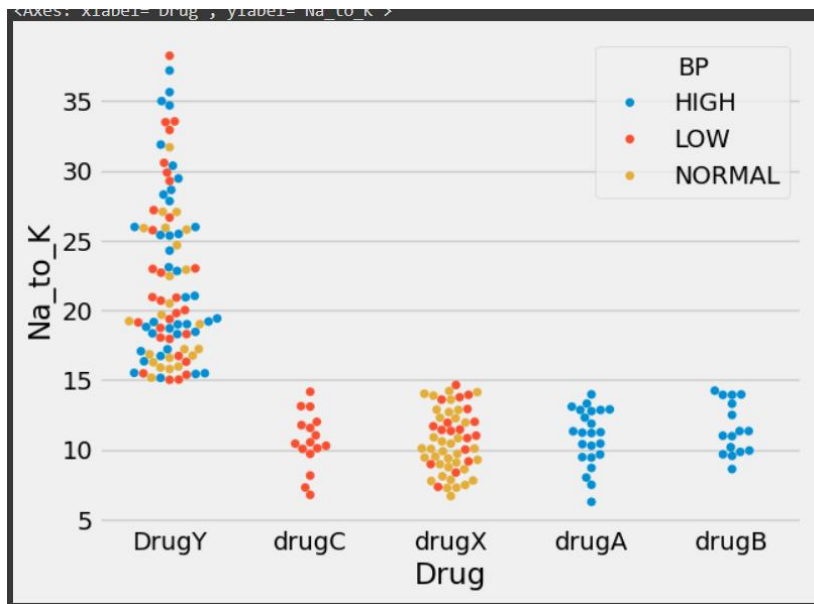
Univariate Analysis



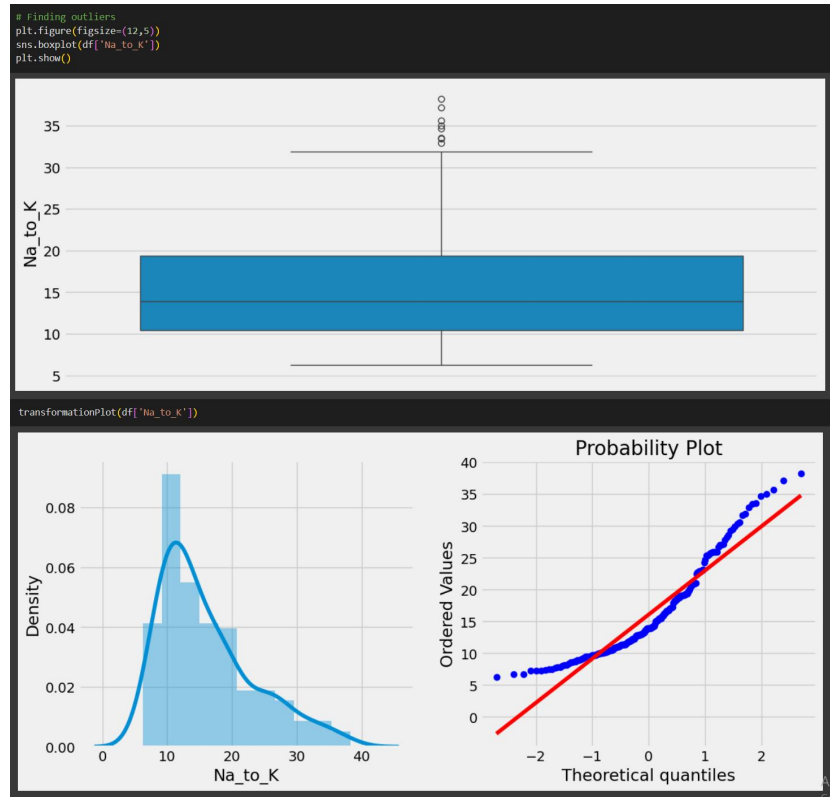
Bivariate Analysis



Multivariate Analysis



Outliers and Anomalies



Data Preprocessing Code Screenshots

Loading Data	<pre># Read the Csv data df = pd.read_csv('/content/sample_data/drug200.csv') df.head()</pre> <table><thead><tr><th></th><th>Age</th><th>Sex</th><th>BP</th><th>Cholesterol</th><th>Na_to_K</th><th>Drug</th></tr></thead><tbody><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></tbody></table>		Age	Sex	BP	Cholesterol	Na_to_K	Drug	0	23	F	HIGH	HIGH	25.355	DrugY	1	47	M	LOW	HIGH	13.093	drugC	2	47	M	LOW	HIGH	10.114	drugC	3	28	F	NORMAL	HIGH	7.798	drugX	4	61	F	LOW	HIGH	18.043	DrugY
	Age	Sex	BP	Cholesterol	Na_to_K	Drug																																					
0	23	F	HIGH	HIGH	25.355	DrugY																																					
1	47	M	LOW	HIGH	13.093	drugC																																					
2	47	M	LOW	HIGH	10.114	drugC																																					
3	28	F	NORMAL	HIGH	7.798	drugX																																					
4	61	F	LOW	HIGH	18.043	DrugY																																					
Handling Missing Data	<pre># Finding null values df.isnull().sum()</pre> <table><tbody><tr><td>Age</td><td>0</td></tr><tr><td>Sex</td><td>0</td></tr><tr><td>BP</td><td>0</td></tr><tr><td>Cholesterol</td><td>0</td></tr><tr><td>Na_to_K</td><td>0</td></tr><tr><td>Drug</td><td>0</td></tr><tr><td colspan="2">dtype: int64</td></tr></tbody></table>	Age	0	Sex	0	BP	0	Cholesterol	0	Na_to_K	0	Drug	0	dtype: int64																													
Age	0																																										
Sex	0																																										
BP	0																																										
Cholesterol	0																																										
Na_to_K	0																																										
Drug	0																																										
dtype: int64																																											
Splitting data into train and test	<pre>x = df.drop('Drug', axis=1) y = df['Drug'] from sklearn.model_selection import train_test_split x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.3, random_state=10) print('Shape of x_train {}'.format(x_train.shape)) print('Shape of y_train {}'.format(y_train.shape)) print('Shape of x_test {}'.format(x_test.shape)) print('Shape of y_test {}'.format(y_test.shape)) Shape of x_train (140, 5) Shape of y_train (140,) Shape of x_test (60, 5) Shape of y_test (60,)</pre>																																										