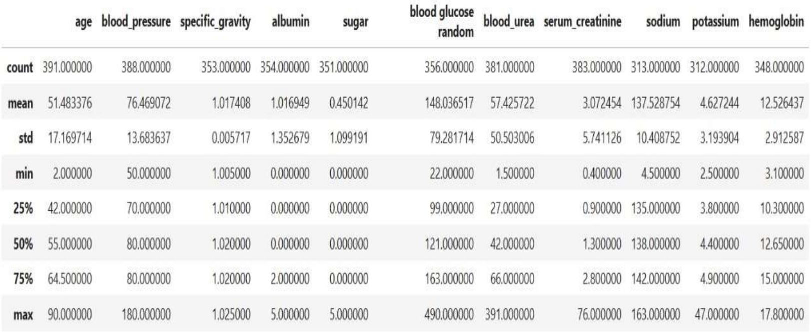
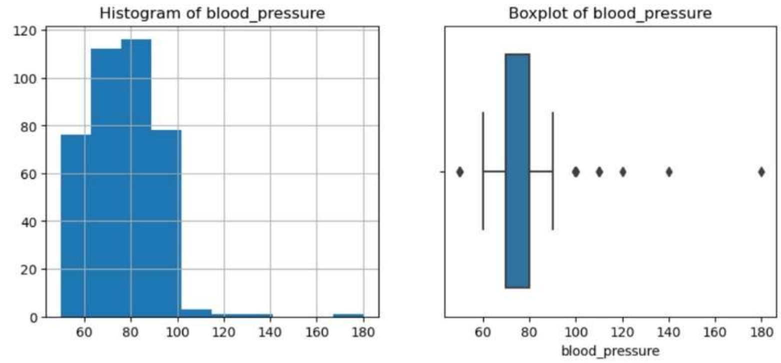
Data Collection and Preprocessing Phase

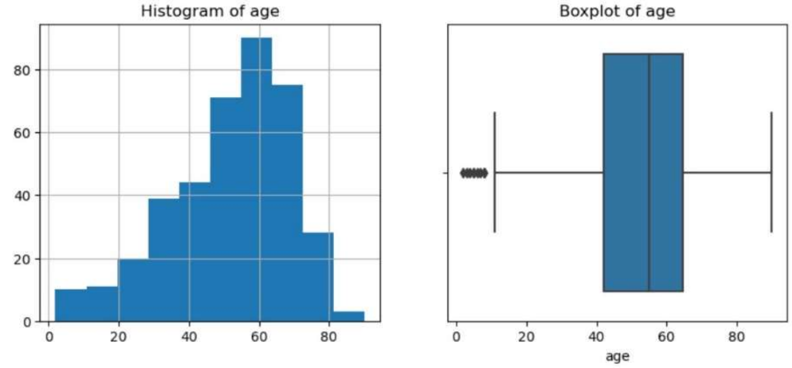
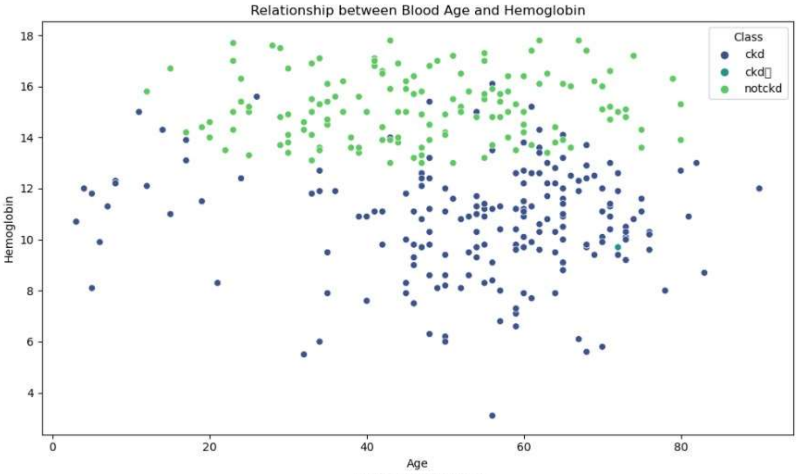
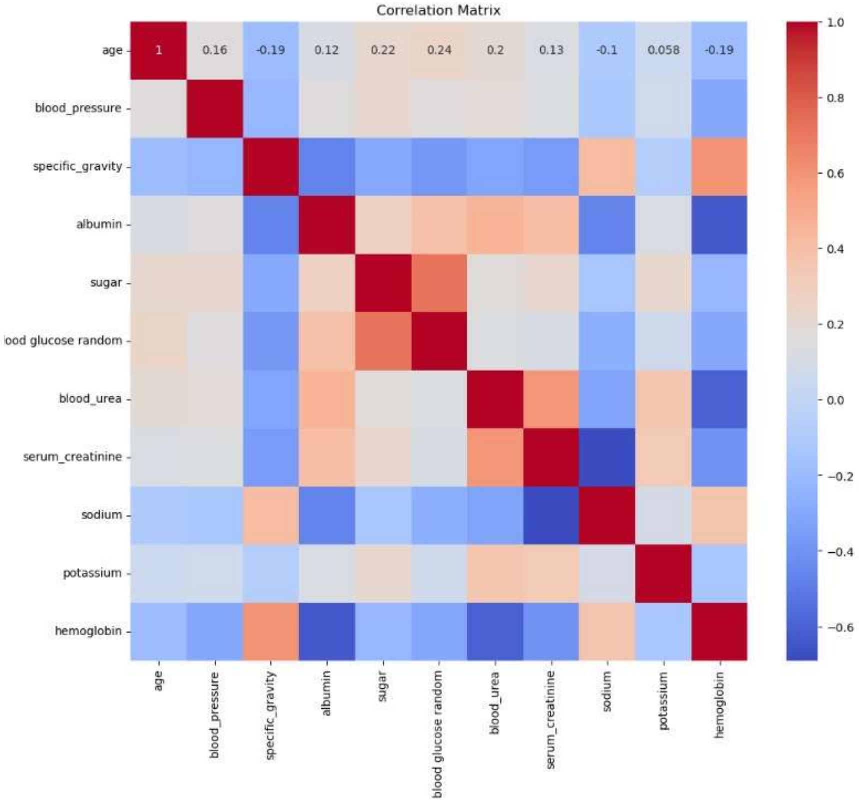
Date	06 July 2024
Team ID	SWTID1720082372
Project Title	Early Prediction of Chronic Kidney Disease Using Machine Learning
Maximum Marks	6 Marks

Data Exploration and Preprocessing Template

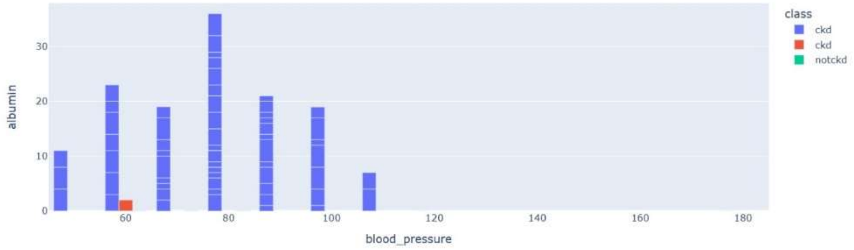
Identifies data sources, assesses quality issues like missing values and duplicates, and implements resolution plans to ensure accurate and reliable analysis.

<table>
<tr><th>Section</th><th>Description</th></tr>
<tr><td>Data Overview</td><td>
Dimension:
400 rows X 26 columns
Descriptive analysis:

<table>
<tr><th></th><th>age</th><th>blood_pressure</th><th>specific_gravity</th><th>albumin</th><th>sugar</th><th>blood glucose random</th><th>blood_urea</th><th>serum_creatinine</th><th>sodium</th><th>potassium</th><th>hemoglobin</th></tr>
<tr><td>count</td><td>391.000000</td><td>388.000000</td><td>353.000000</td><td>354.000000</td><td>351.000000</td><td>356.000000</td><td>381.000000</td><td>383.000000</td><td>313.000000</td><td>312.000000</td><td>348.000000</td></tr>
<tr><td>mean</td><td>51.483376</td><td>76.469072</td><td>1.017408</td><td>1.016949</td><td>0.450142</td><td>148.036517</td><td>57.425722</td><td>3.072454</td><td>137.528754</td><td>4.627244</td><td>12.526437</td></tr>
<tr><td>std</td><td>17.169714</td><td>13.683637</td><td>0.005717</td><td>1.352679</td><td>1.099191</td><td>79.281714</td><td>50.503006</td><td>5.741126</td><td>10.408752</td><td>3.193904</td><td>2.912587</td></tr>
<tr><td>min</td><td>2.000000</td><td>50.000000</td><td>1.005000</td><td>0.000000</td><td>0.000000</td><td>22.000000</td><td>1.500000</td><td>0.400000</td><td>4.500000</td><td>2.500000</td><td>3.100000</td></tr>
<tr><td>25%</td><td>42.000000</td><td>70.000000</td><td>1.010000</td><td>0.000000</td><td>0.000000</td><td>99.000000</td><td>27.000000</td><td>0.900000</td><td>135.000000</td><td>3.800000</td><td>10.300000</td></tr>
<tr><td>50%</td><td>55.000000</td><td>80.000000</td><td>1.020000</td><td>0.000000</td><td>0.000000</td><td>121.000000</td><td>42.000000</td><td>1.300000</td><td>138.000000</td><td>4.400000</td><td>12.650000</td></tr>
<tr><td>75%</td><td>64.500000</td><td>80.000000</td><td>1.020000</td><td>2.000000</td><td>0.000000</td><td>163.000000</td><td>66.000000</td><td>2.800000</td><td>142.000000</td><td>4.900000</td><td>15.000000</td></tr>
<tr><td>max</td><td>90.000000</td><td>180.000000</td><td>1.025000</td><td>5.000000</td><td>5.000000</td><td>490.000000</td><td>391.000000</td><td>76.000000</td><td>163.000000</td><td>47.000000</td><td>17.800000</td></tr>
</table>
</td></tr>
<tr><td>Univariate Analysis</td><td>


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

	
Bivariate Analysis	 

Multivariate Analysis

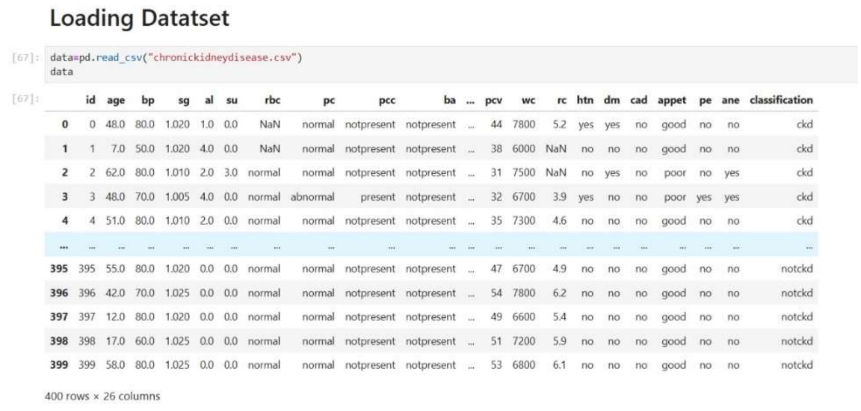


Outliers and Anomalies

-

Data Preprocessing Code Screenshots

Loading Data



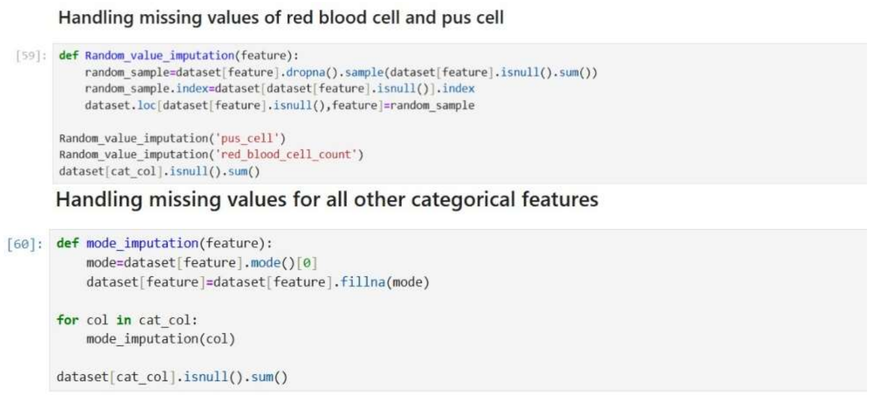
Loading Datatset

```
[67]: data=pd.read_csv("chronickidneydisease.csv")
      data
```

```
[67]:
       id   age    bp     sg   al   su     rbc        pc         pcc          ba  ...  pcv    wc   rc  htn   dm  cad  appet   pe  ane  classification
  0     0  48.0  80.0  1.020  1.0  0.0     NaN    normal  notpresent  notpresent  ...   44  7800  5.2  yes  yes   no   good   no   no             ckd
  1     1   7.0  50.0  1.020  4.0  0.0     NaN    normal  notpresent  notpresent  ...   38  6000  NaN   no   no   no   good   no   no             ckd
  2     2  62.0  80.0  1.010  2.0  3.0  normal    normal  notpresent  notpresent  ...   31  7500  NaN   no  yes   no   poor   no  yes             ckd
  3     3  48.0  70.0  1.005  4.0  0.0  normal  abnormal     present  notpresent  ...   32  6700  3.9  yes   no   no   poor  yes  yes             ckd
  4     4  51.0  80.0  1.010  2.0  0.0  normal    normal  notpresent  notpresent  ...   35  7300  4.6   no   no   no   good   no   no             ckd
...   ...   ...   ...    ...  ...  ...     ...       ...         ...         ...  ...  ...   ...  ...  ...  ...  ...    ...  ...  ...             ...
395   395  55.0  80.0  1.020  0.0  0.0  normal    normal  notpresent  notpresent  ...   47  6700  4.9   no   no   no   good   no   no          notckd
396   396  42.0  70.0  1.025  0.0  0.0  normal    normal  notpresent  notpresent  ...   54  7800  6.2   no   no   no   good   no   no          notckd
397   397  12.0  80.0  1.020  0.0  0.0  normal    normal  notpresent  notpresent  ...   49  6600  5.4   no   no   no   good   no   no          notckd
398   398  17.0  60.0  1.025  0.0  0.0  normal    normal  notpresent  notpresent  ...   51  7200  5.9   no   no   no   good   no   no          notckd
399   399  58.0  80.0  1.025  0.0  0.0  normal    normal  notpresent  notpresent  ...   53  6800  6.1   no   no   no   good   no   no          notckd

400 rows × 26 columns
```

Handling Missing Data



Handling missing values of red blood cell and pus cell

```
[59]: def Random_value_imputation(feature):
          random_sample=dataset[feature].dropna().sample(dataset[feature].isnull().sum())
          random_sample.index=dataset[dataset[feature].isnull()].index
          dataset.loc[dataset[feature].isnull(),feature]=random_sample

      Random_value_imputation('pus_cell')
      Random_value_imputation('red_blood_cell_count')
      dataset[cat_col].isnull().sum()
```

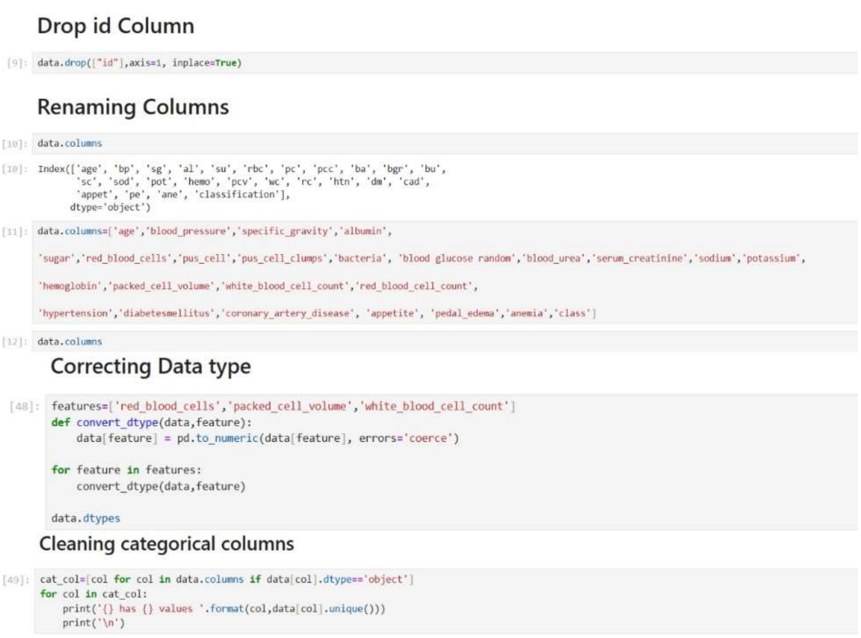
Handling missing values for all other categorical features

```
[60]: def mode_imputation(feature):
          mode=dataset[feature].mode()[0]
          dataset[feature]=dataset[feature].fillna(mode)

      for col in cat_col:
          mode_imputation(col)

      dataset[cat_col].isnull().sum()
```

<table>
<tr>
<td>Data Transformation</td>
<td>



Drop id Column

```
[9]: data.drop(["id"],axis=1, inplace=True)
```

Renaming Columns

```
[10]: data.columns

[10]: Index(['age', 'bp', 'sg', 'al', 'su', 'rbc', 'pc', 'pcc', 'ba', 'bgr', 'bu',
              'sc', 'sod', 'pot', 'hemo', 'pcv', 'wc', 'rc', 'htn', 'dm', 'cad',
              'appet', 'pe', 'ane', 'classification'],
             dtype='object')

[11]: data.columns=['age','blood_pressure','specific_gravity','albumin',

      'sugar','red_blood_cells','pus_cell','pus_cell_clumps','bacteria', 'blood glucose random','blood_urea','serum_creatinine','sodium','potassium',

      'hemoglobin','packed_cell_volume','white_blood_cell_count','red_blood_cell_count',

      'hypertension','diabetesmellitus','coronary_artery_disease', 'appetite', 'pedal_edema','anemia','class']

[12]: data.columns
```

Correcting Data type

```
[48]: features=['red_blood_cells','packed_cell_volume','white_blood_cell_count']
      def convert_dtype(data,feature):
          data[feature] = pd.to_numeric(data[feature], errors='coerce')

      for feature in features:
          convert_dtype(data,feature)

      data.dtypes
```

Cleaning categorical columns

```
[49]: cat_col=[col for col in data.columns if data[col].dtype=='object']
      for col in cat_col:
          print('{} has {} values '.format(col,data[col].unique()))
          print('\n')
```

</td>
</tr>
<tr>
<td>Feature Engineering</td>
<td>Attached the codes in final submission.</td>
</tr>
<tr>
<td>Save Processed Data</td>
<td>-</td>
</tr>
</table>