SMARTBRIDGE
Let's Bridge the Gap
a Veranda Enterprise

Smart Internz

Data Collection and Preprocessing Phase

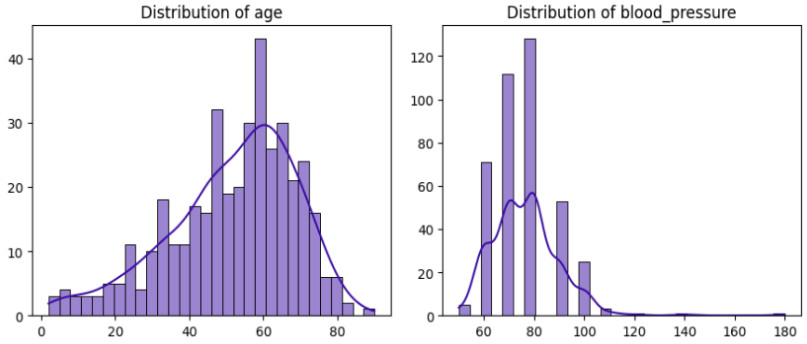
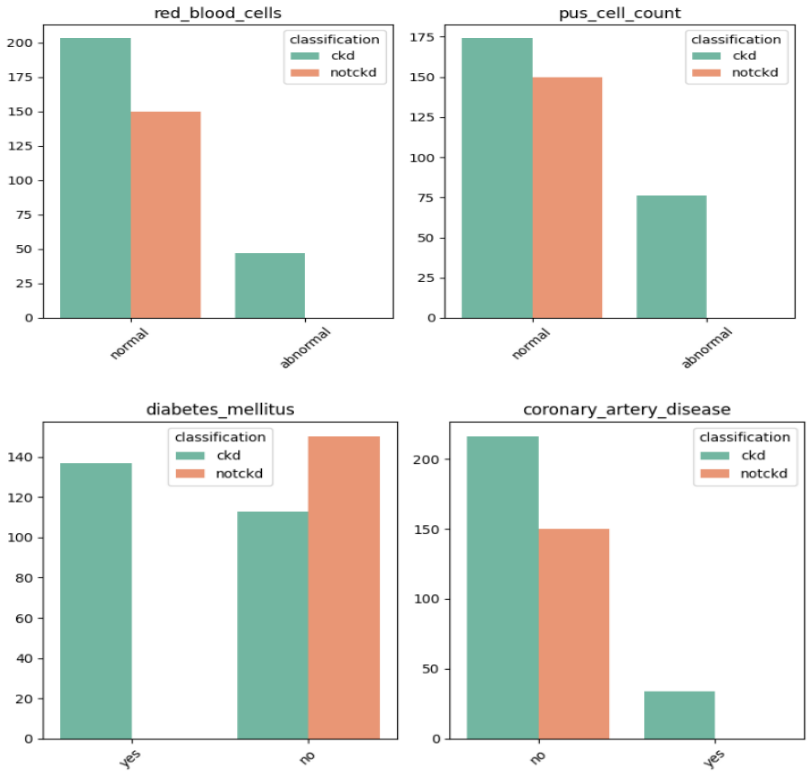
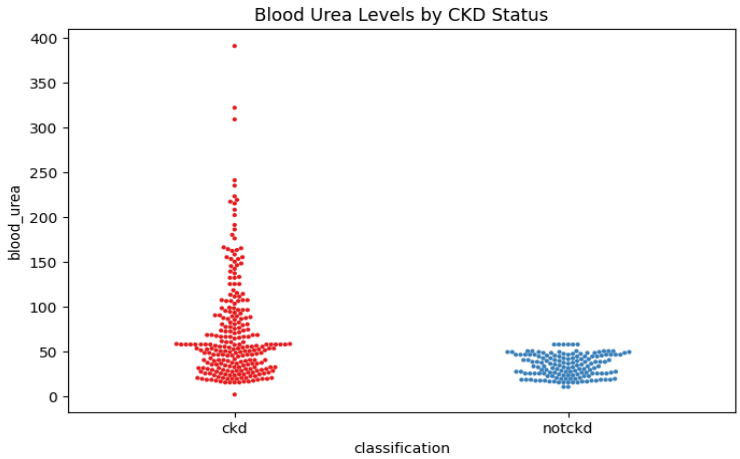
Date	22 June 2025
Team ID	SWTID1749634408
Project Title	Early Prediction for Chronic Kidney Disease Detection: A Progressive Approach to Health Management
Maximum Marks	6 Marks

Data Exploration and Preprocessing Report

Dataset variables will be statistically analyzed to identify patterns and outliers, with Python employed for preprocessing tasks like normalization and feature engineering. Data cleaning will address missing values and outliers, ensuring quality for subsequent analysis and modeling, and forming a strong foundation for insights and predictions.

Section	Description
Data Overview	Dimension: 614 rows × 13 columns Descriptive statistics:

	age	blood_pressure	specific_gravity	albumin	sugar
count	391.000000	388.000000	353.000000	354.000000	351.000000
mean	51.483376	76.469072	1.017408	1.016949	0.450142
std	17.169714	13.683637	0.005717	1.352679	1.099191
min	2.000000	50.000000	1.005000	0.000000	0.000000
25%	42.000000	70.000000	1.010000	0.000000	0.000000
50%	55.000000	80.000000	1.020000	0.000000	0.000000
75%	64.500000	80.000000	1.020000	2.000000	0.000000
max	90.000000	180.000000	1.025000	5.000000	5.000000

Univariate Analysis	
Bivariate Analysis	
Multivariate Analysis	

Outliers and Anomalies	-
Data Preprocessing Code Screenshots	
Loading Data	`df.head()`

	id	age	bp	sg	al	su	rbc	pc	pcc	ba
0	0	48.0	80.0	1.020	1.0	0.0	NaN	normal	notpresent	notpresent
1	1	7.0	50.0	1.020	4.0	0.0	NaN	normal	notpresent	notpresent
2	2	62.0	80.0	1.010	2.0	3.0	normal	normal	notpresent	notpresent
3	3	48.0	70.0	1.005	4.0	0.0	normal	abnormal	present	notpresent
4	4	51.0	80.0	1.010	2.0	0.0	normal	normal	notpresent	notpresent

Handling Missing Data

```
df['age'] = df['age'].fillna(df['age'].mode()[0])
df['hypertension'] = df['hypertension'].fillna(df['hypertension'].mode()[0])
df['pus_cell_clumps'] = df['pus_cell_clumps'].fillna(df['pus_cell_clumps'].mode()[0])
df['appetite'] = df['appetite'].fillna(df['appetite'].mode()[0])
df['albumin'] = df['albumin'].fillna(df['albumin'].mode()[0])
df['pus_cell_count'] = df['pus_cell_count'].fillna(df['pus_cell_count'].mode()[0])
df['red_blood_cells'] = df['red_blood_cells'].fillna(df['red_blood_cells'].mode()[0])
df['coronary_artery_disease'] = df['coronary_artery_disease'].fillna(df['coronary_artery_disease'].mode()[0])
df['bacteria'] = df['bacteria'].fillna(df['bacteria'].mode()[0])
df['anemia'] = df['anemia'].fillna(df['anemia'].mode()[0])
df['sugar'] = df['sugar'].fillna(df['sugar'].mode()[0])
df['diabetes_mellitus'] = df['diabetes_mellitus'].fillna(df['diabetes_mellitus'].mode()[0])
df['pedal_edema'] = df['pedal_edema'].fillna(df['pedal_edema'].mode()[0])
df['specific_gravity'] = df['specific_gravity'].fillna(df['specific_gravity'].mode()[0])
```

Data Transformation

```
df['diabetes_mellitus'] = df['diabetes_mellitus'].replace(to_replace = {'\tno' : 'no', '\tyes' : 'yes', ' yes' : 'yes'})
df['coronary_artery_disease'] = df['coronary_artery_disease'].replace(to_replace = {'\tno' : 'no'})
df['classification'] = df['classification'].replace(to_replace = {'ckd\t' : 'ckd'})
```

```
df['packed_cell_volume'] = pd.to_numeric(df['packed_cell_volume'], errors = 'coerce')
df['white_blood_cell_count'] = pd.to_numeric(df['white_blood_cell_count'], errors = 'coerce')
df['red_blood_cell_count'] = pd.to_numeric(df['red_blood_cell_count'], errors = 'coerce')
```

Feature Engineering	Attached the codes in final submission.
Save Processed Data	-