KNNProj1

September 27, 2024

[1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import matplotlib as mpl
import seaborn as sns
```

1.Exploratory Data Analysis (EDA)

[4]:
```
df = pd.read_csv(r"C:\Users\User\Downloads\archive (2)\diabetes.csv")
```

[6]:
```
df.head()
```

[6]:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	\
0	6	148	72	35	0	33.6	
1	1	85	66	29	0	26.6	
2	8	183	64	0	0	23.3	
3	1	89	66	23	94	28.1	
4	0	137	40	35	168	43.1	

	DiabetesPedigreeFunction	Age	Outcome
0	0.627	50	1
1	0.351	31	0
2	0.672	32	1
3	0.167	21	0
4	2.288	33	1

[8]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 768 entries, 0 to 767
Data columns (total 9 columns):
 #   Column                    Non-Null Count  Dtype
---  ------                    --------------  -----
 0   Pregnancies               768 non-null    int64
 1   Glucose                   768 non-null    int64
 2   BloodPressure             768 non-null    int64
 3   SkinThickness             768 non-null    int64
 4   Insulin                   768 non-null    int64
 5   BMI                       768 non-null    float64
```

```
 6   DiabetesPedigreeFunction  768 non-null    float64
 7   Age                       768 non-null    int64
 8   Outcome                   768 non-null    int64
dtypes: float64(2), int64(7)
memory usage: 54.1 KB
```

[22]:
```
df.describe()
```

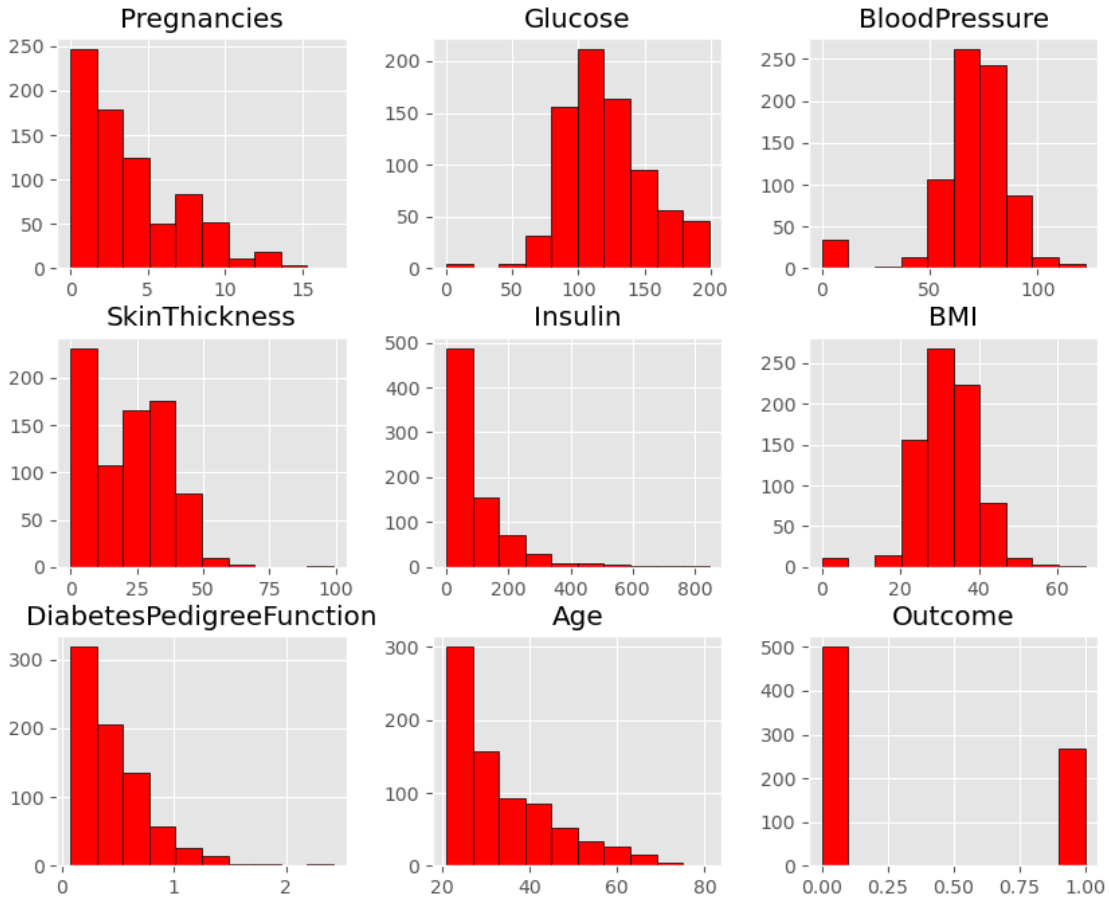
[22]:
```
       Pregnancies     Glucose  BloodPressure  SkinThickness     Insulin  \
count   768.000000  768.000000     768.000000     768.000000  768.000000
mean      3.845052  120.894531      69.105469      20.536458   79.799479
std       3.369578   31.972618      19.355807      15.952218  115.244002
min       0.000000    0.000000       0.000000       0.000000    0.000000
25%       1.000000   99.000000      62.000000       0.000000    0.000000
50%       3.000000  117.000000      72.000000      23.000000   30.500000
75%       6.000000  140.250000      80.000000      32.000000  127.250000
max      17.000000  199.000000     122.000000      99.000000  846.000000

              BMI  DiabetesPedigreeFunction         Age     Outcome
count  768.000000                768.000000  768.000000  768.000000
mean    31.992578                  0.471876   33.240885    0.348958
std      7.884160                  0.331329   11.760232    0.476951
min      0.000000                  0.078000   21.000000    0.000000
25%     27.300000                  0.243750   24.000000    0.000000
50%     32.000000                  0.372500   29.000000    0.000000
75%     36.600000                  0.626250   41.000000    1.000000
max     67.100000                  2.420000   81.000000    1.000000
```

[24]:
```
df.isnull().sum()
```

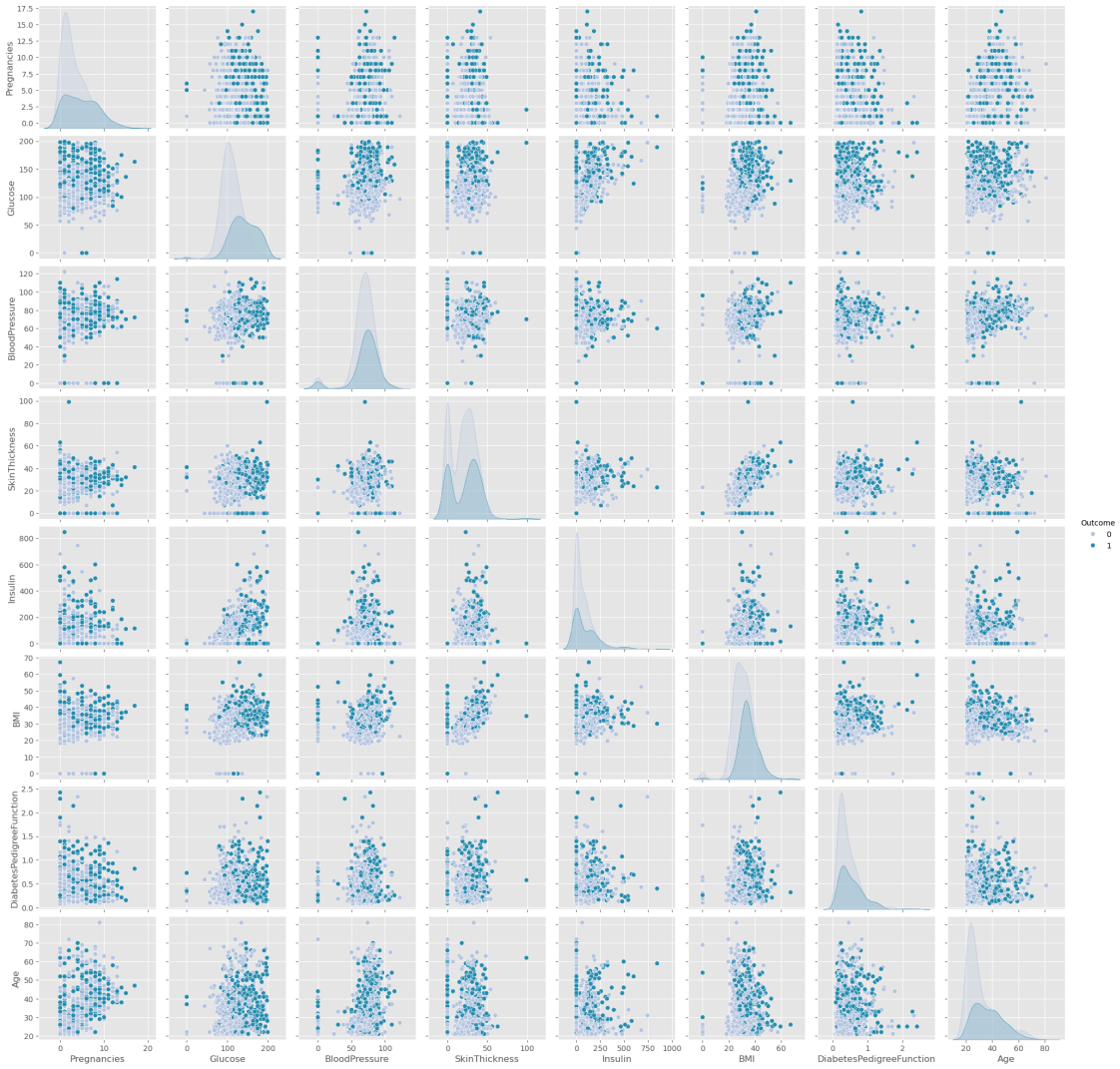
[24]:
```
Pregnancies                 0
Glucose                     0
BloodPressure               0
SkinThickness               0
Insulin                     0
BMI                         0
DiabetesPedigreeFunction    0
Age                         0
Outcome                     0
dtype: int64
```

[30]:
```
plt.style.use('ggplot')
```

[81]:
```
df.hist(figsize=(10, 8), bins=10, color='Red', edgecolor='black')
plt.show()
```



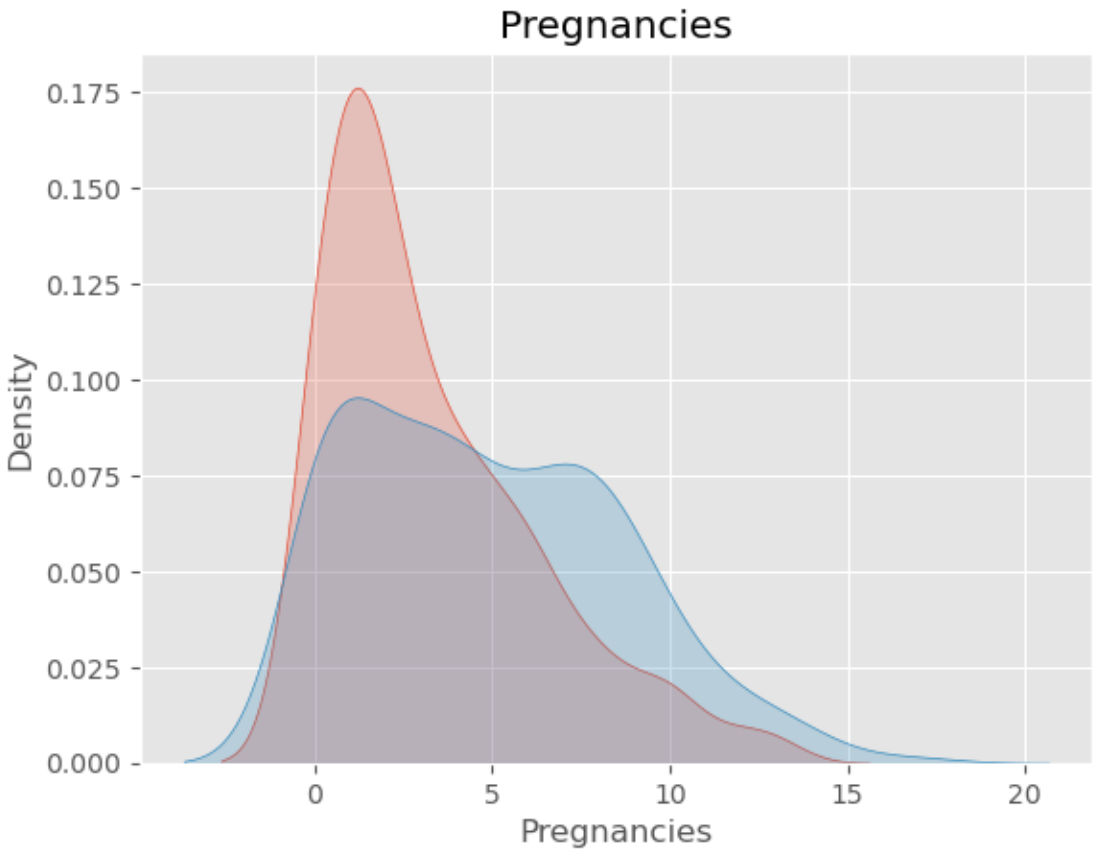
[64]:
```
sns.pairplot(df, hue='Outcome', palette='PuBuGn')
plt.show()
```



[38]:
```
for column in df.columns[:-1]:  # Assuming 'Outcome' is the last column
    sns.kdeplot(df[df['Outcome'] == 0][column], label='Healthy', shade=True)
    sns.kdeplot(df[df['Outcome'] == 1][column], label='Diabetes', shade=True)
    plt.title(column)
    plt.show()
```

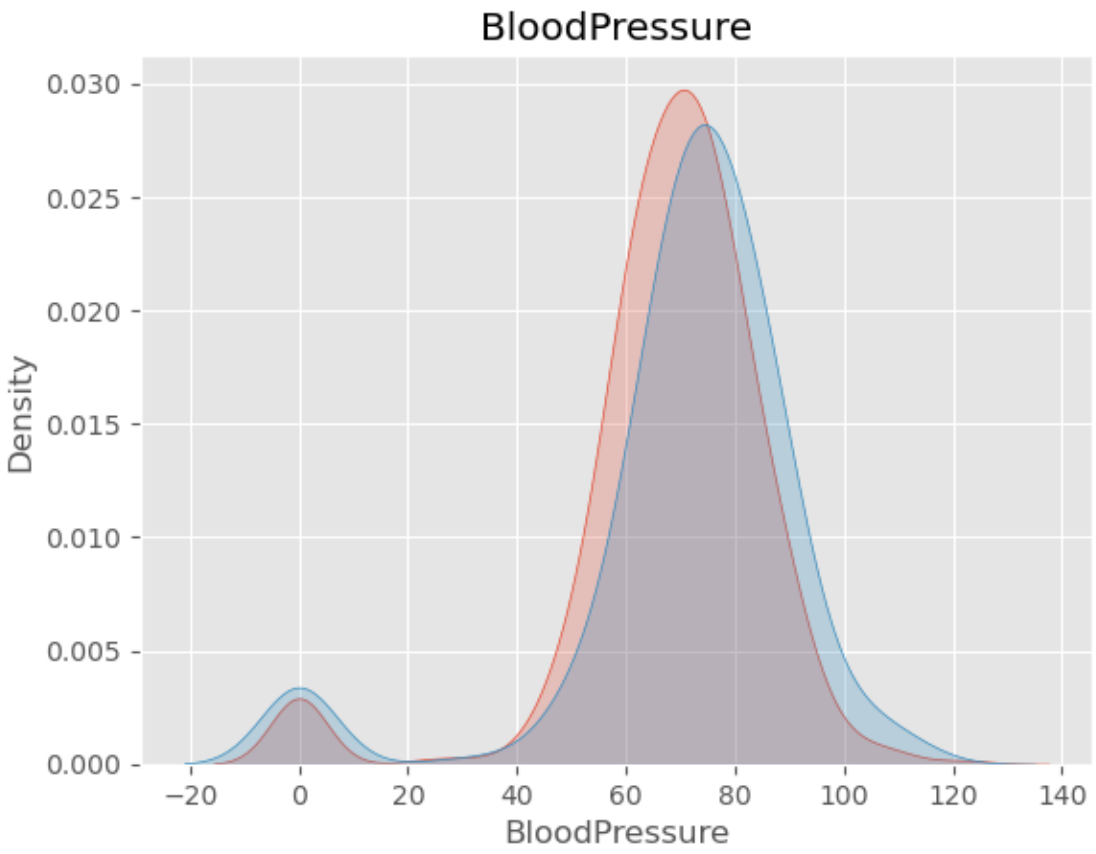
```
C:\Users\User\AppData\Local\Temp\ipykernel_21032\3555084767.py:2: FutureWarning:

`shade` is now deprecated in favor of `fill`; setting `fill=True`.
This will become an error in seaborn v0.14.0; please update your code.

  sns.kdeplot(df[df['Outcome'] == 0][column], label='Healthy', shade=True)
C:\Users\User\AppData\Local\Temp\ipykernel_21032\3555084767.py:3: FutureWarning:

`shade` is now deprecated in favor of `fill`; setting `fill=True`.
```

```
This will become an error in seaborn v0.14.0; please update your code.

  sns.kdeplot(df[df['Outcome'] == 1][column], label='Diabetes', shade=True)
```

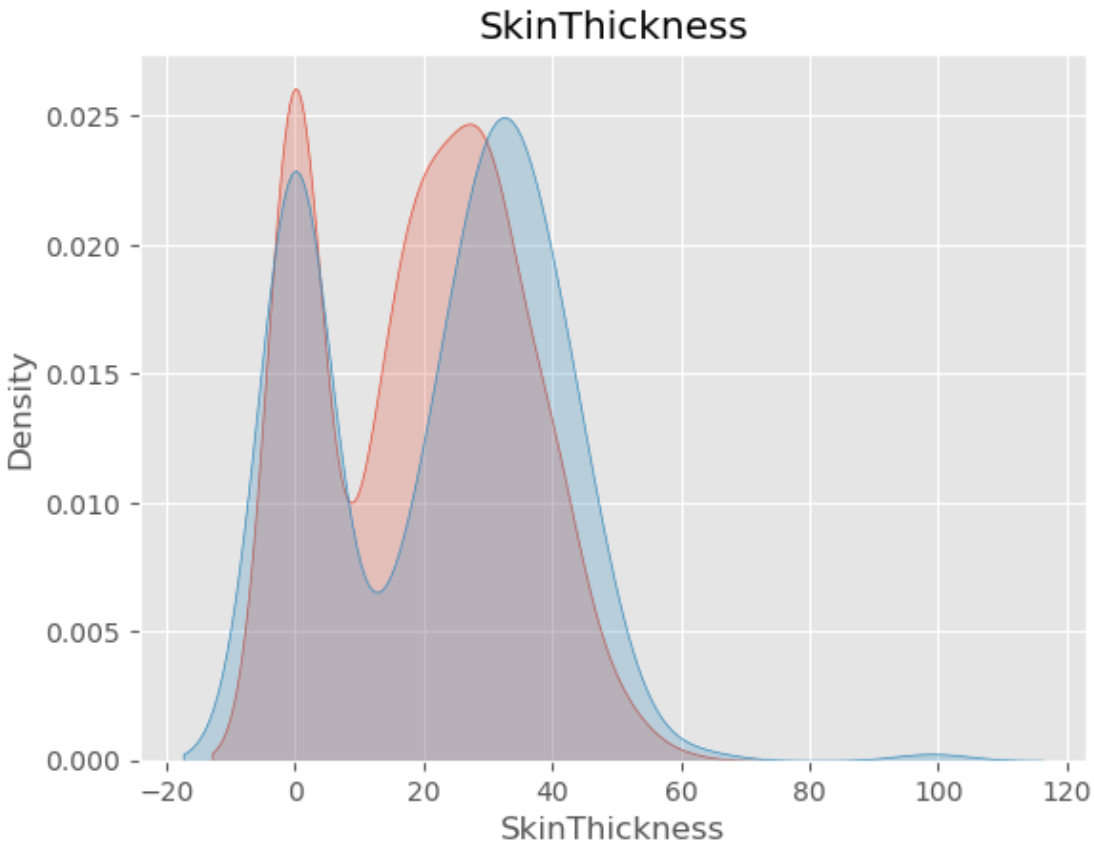



```
C:\Users\User\AppData\Local\Temp\ipykernel_21032\3555084767.py:2: FutureWarning:

`shade` is now deprecated in favor of `fill`; setting `fill=True`.
This will become an error in seaborn v0.14.0; please update your code.

  sns.kdeplot(df[df['Outcome'] == 0][column], label='Healthy', shade=True)
C:\Users\User\AppData\Local\Temp\ipykernel_21032\3555084767.py:3: FutureWarning:

`shade` is now deprecated in favor of `fill`; setting `fill=True`.
This will become an error in seaborn v0.14.0; please update your code.

  sns.kdeplot(df[df['Outcome'] == 1][column], label='Diabetes', shade=True)
```

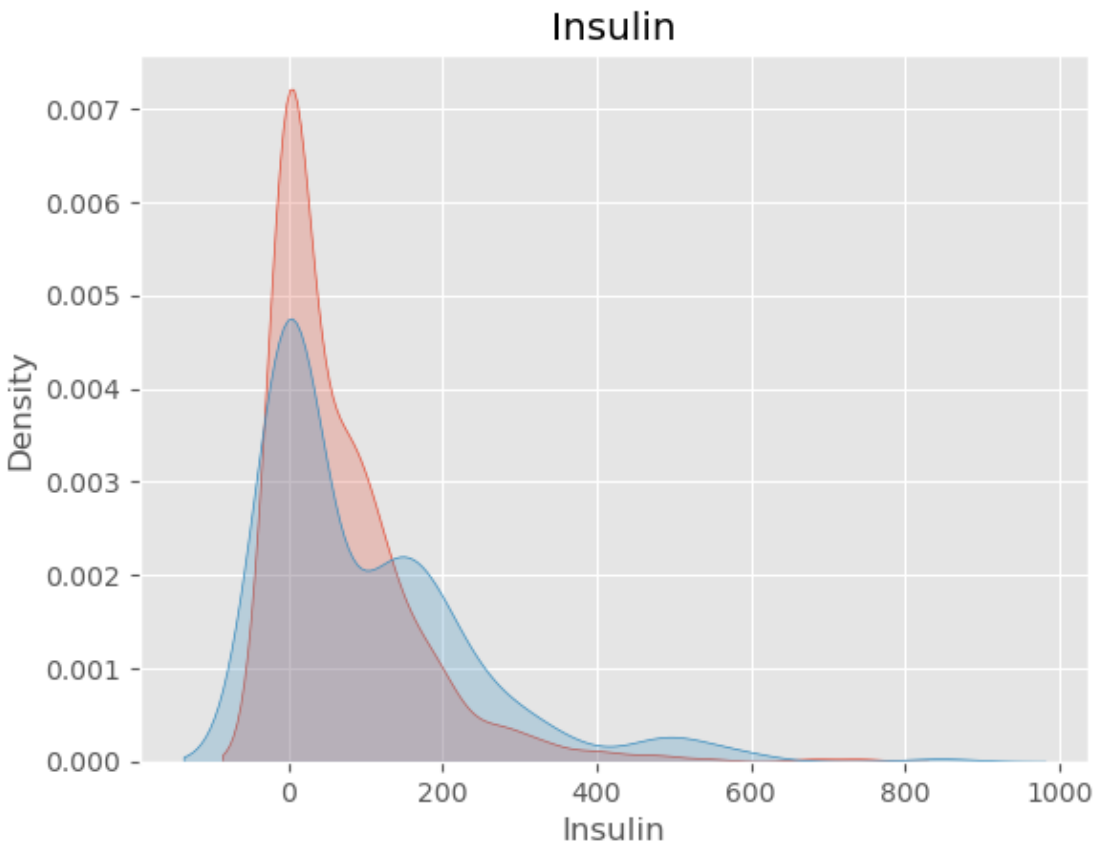

```
C:\Users\User\AppData\Local\Temp\ipykernel_21032\3555084767.py:2: FutureWarning:

`shade` is now deprecated in favor of `fill`; setting `fill=True`.
This will become an error in seaborn v0.14.0; please update your code.

  sns.kdeplot(df[df['Outcome'] == 0][column], label='Healthy', shade=True)
C:\Users\User\AppData\Local\Temp\ipykernel_21032\3555084767.py:3: FutureWarning:

`shade` is now deprecated in favor of `fill`; setting `fill=True`.
This will become an error in seaborn v0.14.0; please update your code.

  sns.kdeplot(df[df['Outcome'] == 1][column], label='Diabetes', shade=True)
```



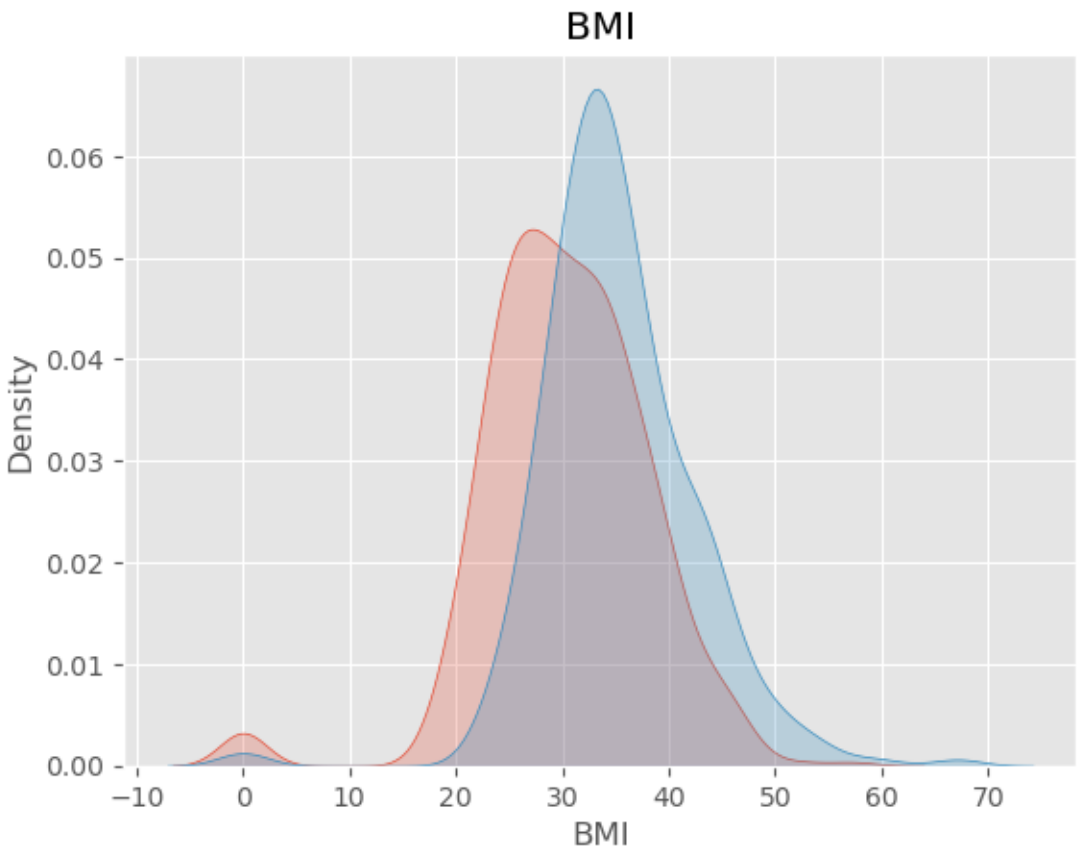

```
C:\Users\User\AppData\Local\Temp\ipykernel_21032\3555084767.py:2: FutureWarning:

`shade` is now deprecated in favor of `fill`; setting `fill=True`.
This will become an error in seaborn v0.14.0; please update your code.

  sns.kdeplot(df[df['Outcome'] == 0][column], label='Healthy', shade=True)
C:\Users\User\AppData\Local\Temp\ipykernel_21032\3555084767.py:3: FutureWarning:

`shade` is now deprecated in favor of `fill`; setting `fill=True`.
This will become an error in seaborn v0.14.0; please update your code.

  sns.kdeplot(df[df['Outcome'] == 1][column], label='Diabetes', shade=True)
```



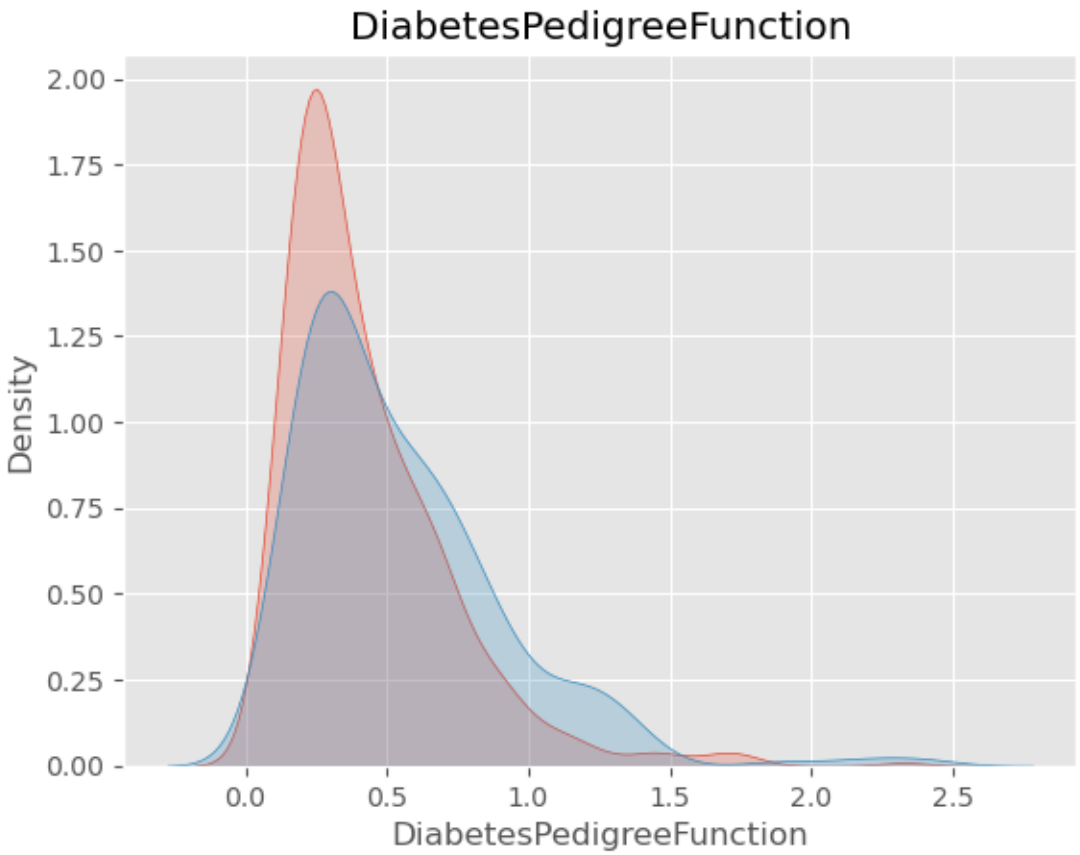

```
C:\Users\User\AppData\Local\Temp\ipykernel_21032\3555084767.py:2: FutureWarning:

`shade` is now deprecated in favor of `fill`; setting `fill=True`.
This will become an error in seaborn v0.14.0; please update your code.

  sns.kdeplot(df[df['Outcome'] == 0][column], label='Healthy', shade=True)
C:\Users\User\AppData\Local\Temp\ipykernel_21032\3555084767.py:3: FutureWarning:

`shade` is now deprecated in favor of `fill`; setting `fill=True`.
This will become an error in seaborn v0.14.0; please update your code.

  sns.kdeplot(df[df['Outcome'] == 1][column], label='Diabetes', shade=True)
```



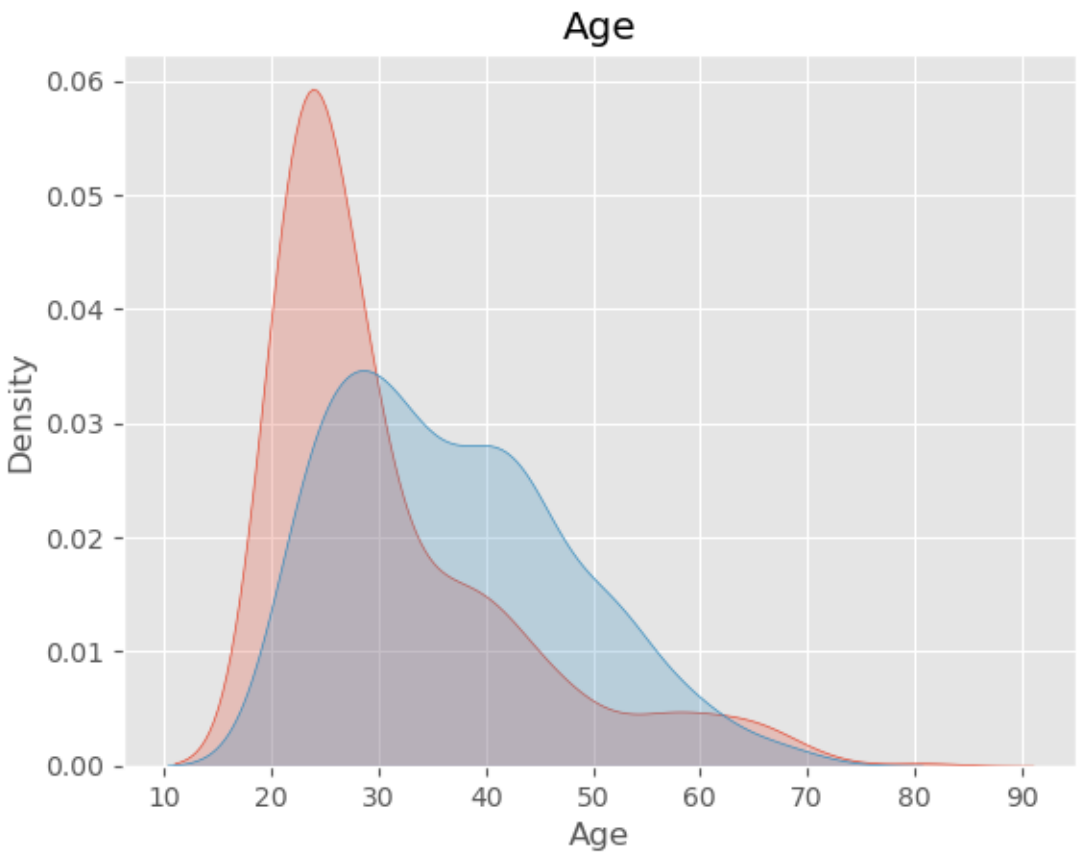

```
C:\Users\User\AppData\Local\Temp\ipykernel_21032\3555084767.py:2: FutureWarning:

`shade` is now deprecated in favor of `fill`; setting `fill=True`.
This will become an error in seaborn v0.14.0; please update your code.

  sns.kdeplot(df[df['Outcome'] == 0][column], label='Healthy', shade=True)
C:\Users\User\AppData\Local\Temp\ipykernel_21032\3555084767.py:3: FutureWarning:

`shade` is now deprecated in favor of `fill`; setting `fill=True`.
This will become an error in seaborn v0.14.0; please update your code.

  sns.kdeplot(df[df['Outcome'] == 1][column], label='Diabetes', shade=True)
```



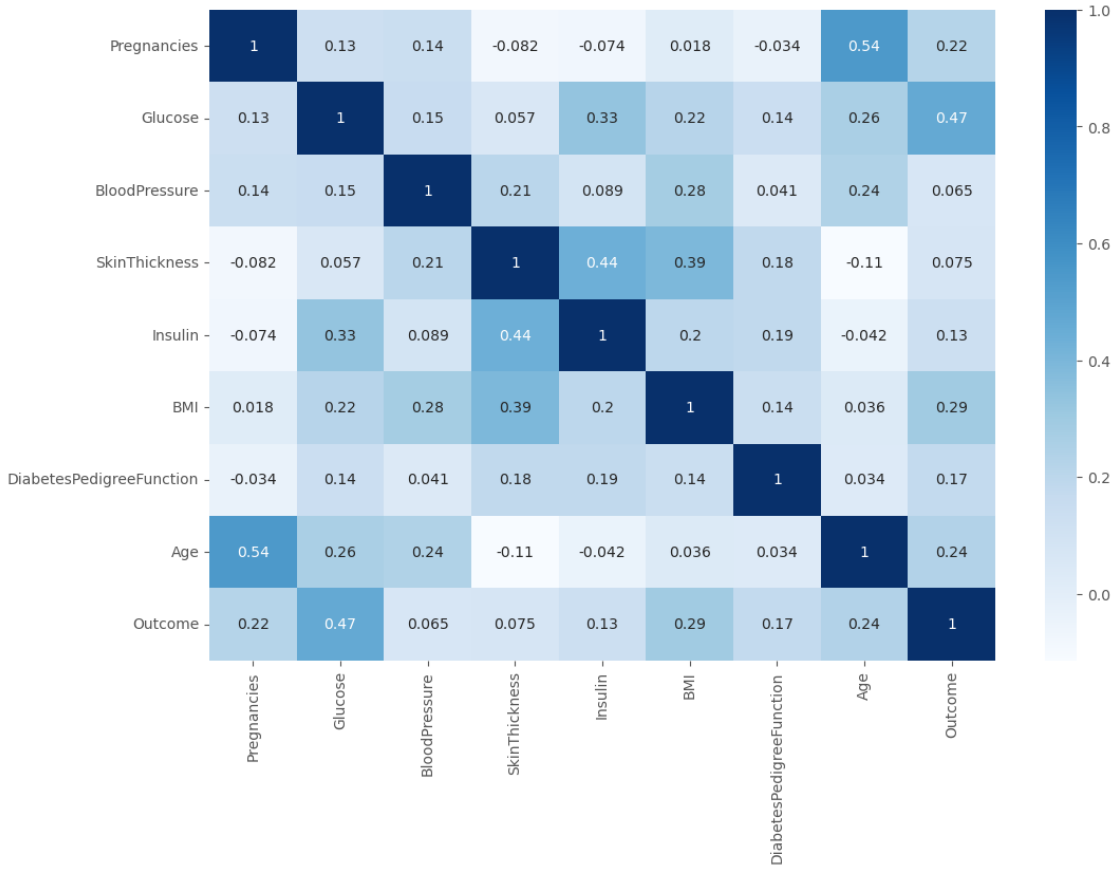

```
C:\Users\User\AppData\Local\Temp\ipykernel_21032\3555084767.py:2: FutureWarning:

`shade` is now deprecated in favor of `fill`; setting `fill=True`.
This will become an error in seaborn v0.14.0; please update your code.

  sns.kdeplot(df[df['Outcome'] == 0][column], label='Healthy', shade=True)
C:\Users\User\AppData\Local\Temp\ipykernel_21032\3555084767.py:3: FutureWarning:

`shade` is now deprecated in favor of `fill`; setting `fill=True`.
This will become an error in seaborn v0.14.0; please update your code.

  sns.kdeplot(df[df['Outcome'] == 1][column], label='Diabetes', shade=True)
```



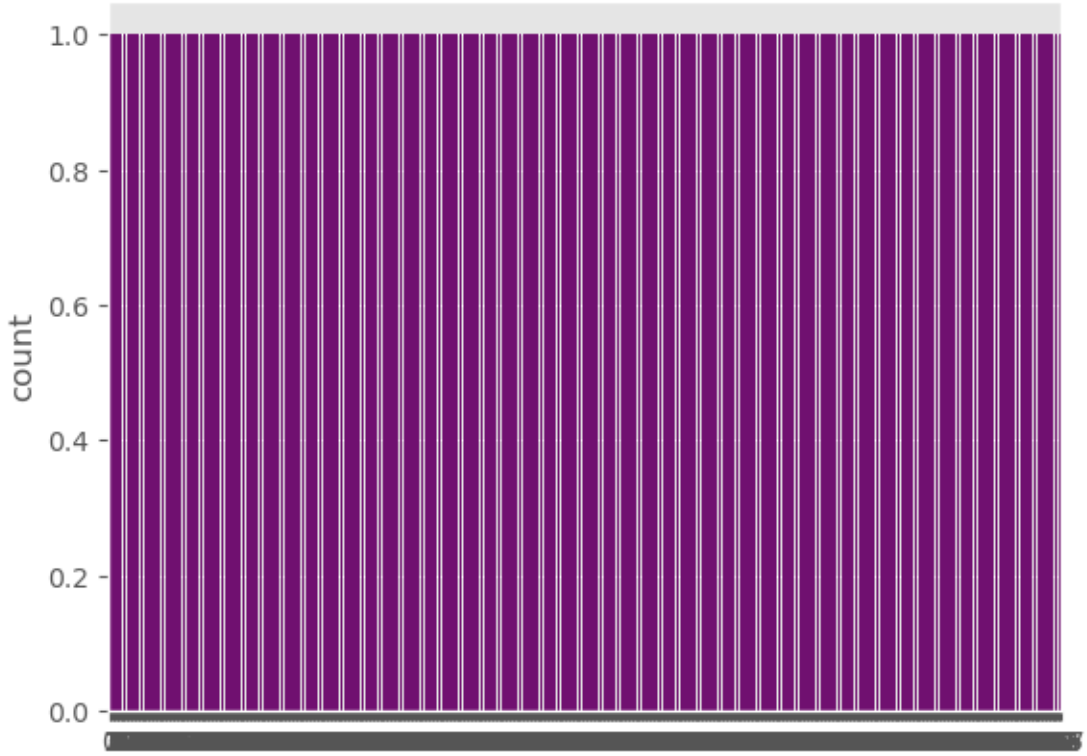

```
C:\Users\User\AppData\Local\Temp\ipykernel_21032\3555084767.py:2: FutureWarning:

`shade` is now deprecated in favor of `fill`; setting `fill=True`.
This will become an error in seaborn v0.14.0; please update your code.

  sns.kdeplot(df[df['Outcome'] == 0][column], label='Healthy', shade=True)
C:\Users\User\AppData\Local\Temp\ipykernel_21032\3555084767.py:3: FutureWarning:

`shade` is now deprecated in favor of `fill`; setting `fill=True`.
This will become an error in seaborn v0.14.0; please update your code.

  sns.kdeplot(df[df['Outcome'] == 1][column], label='Diabetes', shade=True)
```



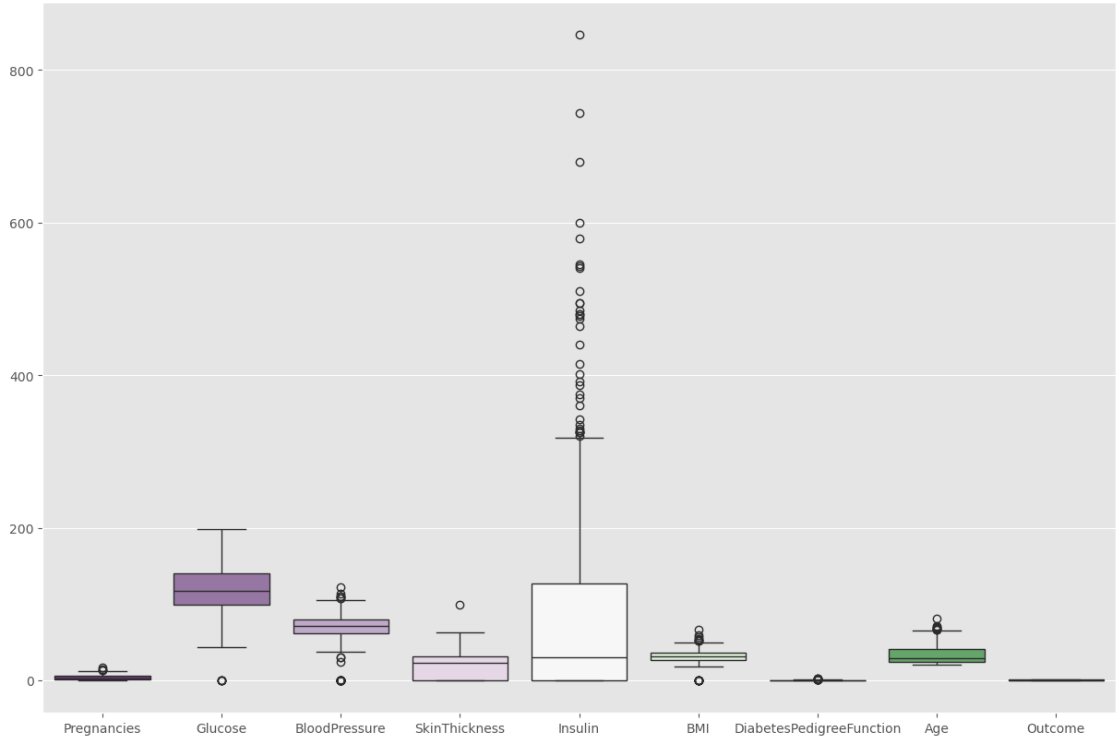
[42]:
```
plt.figure(figsize=(12, 8))
sns.heatmap(df.corr(), annot=True, cmap='Blues')
plt.show()
```



[60]:
```
sns.countplot(df['Outcome'],color='purple')
plt.show()
```



[62]:
```
plt.figure(figsize=(15, 10))
sns.boxplot(data=df, palette='PRGn')
plt.show()
```



2. Preprocessing for KNN

[84]:
```
from sklearn.preprocessing import StandardScaler
```

[86]:
```
scaler = StandardScaler()
X = df.drop('Outcome', axis=1)
y = df['Outcome']
X_scaled = scaler.fit_transform(X)
```

[88]:
```
from sklearn.model_selection import train_test_split
```

[90]:
```
X_train, X_test, y_train, y_test = train_test_split(X_scaled, y, test_size=0.2,␣
↪random_state=42)
```

3.KNN Classifier

[93]:
```
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import classification_report, confusion_matrix
```

[95]:
```
# Initialize the KNN model
knn = KNeighborsClassifier(n_neighbors=5)
knn.fit(X_train, y_train)
```

[95]:
```
KNeighborsClassifier()
```

[97]:
```
# Predict on the test data
y_pred = knn.predict(X_test)
```

[99]:
```
# Evaluate the model
print(classification_report(y_test, y_pred))
```

```
              precision    recall  f1-score   support

           0       0.74      0.80      0.77        99
           1       0.57      0.49      0.53        55

    accuracy                           0.69       154
   macro avg       0.66      0.64      0.65       154
weighted avg       0.68      0.69      0.68       154
```

[101]:
```
from sklearn.model_selection import cross_val_score
```

[103]:
```
k_range = range(1, 31)
scores = []
```

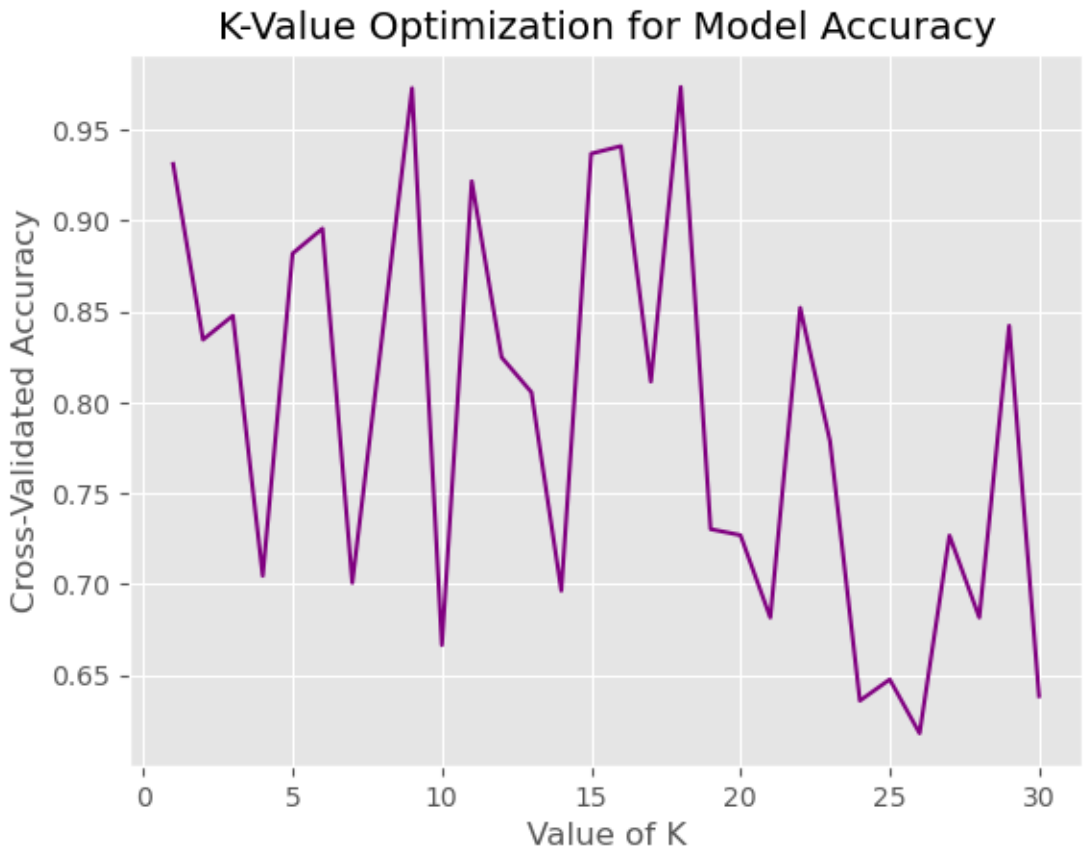
[105]:
```
for k in k_range:
    knn = KNeighborsClassifier(n_neighbors=k)
    score = cross_val_score(knn, X_train, y_train, cv=10, scoring='accuracy')
    scores.append(score.mean())
```

[109]:
```
print(f'Length of k_range: {len(k_range)}')
print(f'Length of scores: {len(scores)}')
```

```
Length of k_range: 30
Length of scores: 30
```

[111]:
```
k_range = list(range(1, 31))  # k_range with 30 elements
# Generate random scores for demonstration (between 0.6 and 1.0)
scores = np.random.uniform(0.6, 1.0, size=30).tolist()  # Generate 30 random␣
 ↪scores
```

[113]:
```
plt.plot(k_range, scores, color='Purple')

# Adding labels and title
plt.xlabel('Value of K')
plt.ylabel('Cross-Validated Accuracy')
plt.title('K-Value Optimization for Model Accuracy')

# Show the plot
plt.show()
```



[ ]:
```
df.to_csv('final_diabetes_data.csv', index=False)
```