

day93-94-diabetes-prediction

January 28, 2024

Day93-94 Diabetes Prediction By: Loga Aswin

Import Libraries

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

from sklearn.metrics import classification_report
from sklearn.metrics import confusion_matrix
```

Load Datasets

```
[96]: df = pd.read_csv("/content/diabetes.csv")
```

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

```
[97]:
```

	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

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

```
[98]:
```

	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

```
[99]: 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
```

```
[100]: df.shape
```

```
[100]: (768, 9)
```

```
[101]: df.value_counts()
```

```
[101]: Pregnancies  Glucose  BloodPressure  SkinThickness  Insulin  BMI
DiabetesPedigreeFunction  Age  Outcome
0          57          60           0           0      21.7  0.735
67   0          1
67          67          76           0           0      45.3  0.194
46   0          1
5          103         108          37           0      39.2  0.305
```

```

65  0      1
      104      74      0      0      28.8  0.153
48  0      1
      105      72      29      325      36.9  0.159
28  0      1
      ..
2      84      50      23      76      30.4  0.968
21  0      1
      85      65      0      0      39.6  0.930
27  0      1
      87      0      23      0      28.9  0.773
25  0      1
      58      16      52      32.7  0.166
25  0      1
17      163      72      41      114      40.9  0.817
47  1      1
Length: 768, dtype: int64

```

```
[102]: df.columns
```

```
[102]: Index(['Pregnancies', 'Glucose', 'BloodPressure', 'SkinThickness', 'Insulin',
          'BMI', 'DiabetesPedigreeFunction', 'Age', 'Outcome'],
          dtype='object')
```

Checking Null Values

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

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

Exploratory Data Analysis

```
[104]: df.corr()
```

```
[104]:
```

	Pregnancies	Glucose	BloodPressure	SkinThickness	\
Pregnancies	1.000000	0.129459	0.141282	-0.081672	
Glucose	0.129459	1.000000	0.152590	0.057328	
BloodPressure	0.141282	0.152590	1.000000	0.207371	
SkinThickness	-0.081672	0.057328	0.207371	1.000000	

Insulin	-0.073535	0.331357	0.088933	0.436783
BMI	0.017683	0.221071	0.281805	0.392573
DiabetesPedigreeFunction	-0.033523	0.137337	0.041265	0.183928
Age	0.544341	0.263514	0.239528	-0.113970
Outcome	0.221898	0.466581	0.065068	0.074752

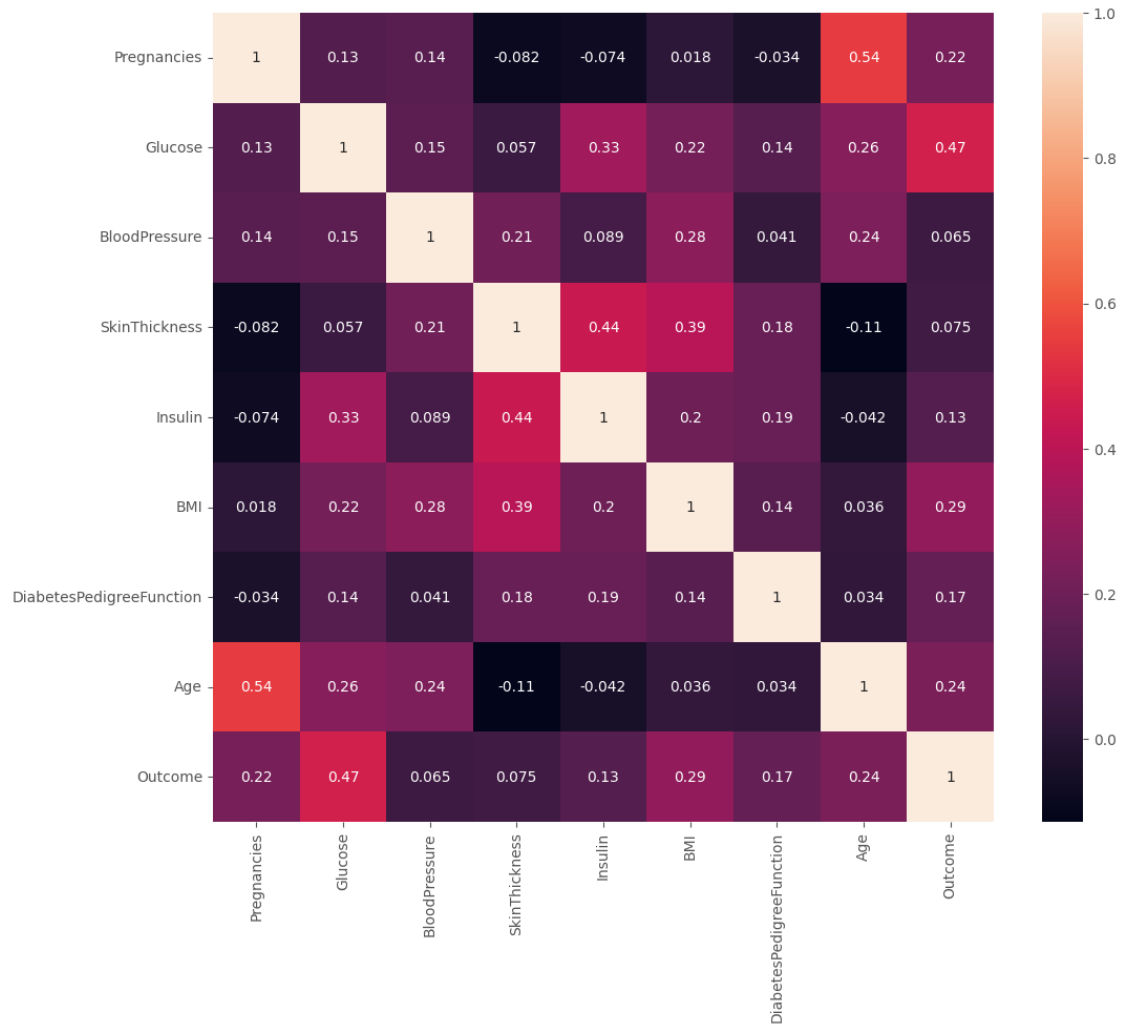
	Insulin	BMI	DiabetesPedigreeFunction	\
Pregnancies	-0.073535	0.017683	-0.033523	
Glucose	0.331357	0.221071	0.137337	
BloodPressure	0.088933	0.281805	0.041265	
SkinThickness	0.436783	0.392573	0.183928	
Insulin	1.000000	0.197859	0.185071	
BMI	0.197859	1.000000	0.140647	
DiabetesPedigreeFunction	0.185071	0.140647	1.000000	
Age	-0.042163	0.036242	0.033561	
Outcome	0.130548	0.292695	0.173844	

	Age	Outcome
Pregnancies	0.544341	0.221898
Glucose	0.263514	0.466581
BloodPressure	0.239528	0.065068
SkinThickness	-0.113970	0.074752
Insulin	-0.042163	0.130548
BMI	0.036242	0.292695
DiabetesPedigreeFunction	0.033561	0.173844
Age	1.000000	0.238356
Outcome	0.238356	1.000000

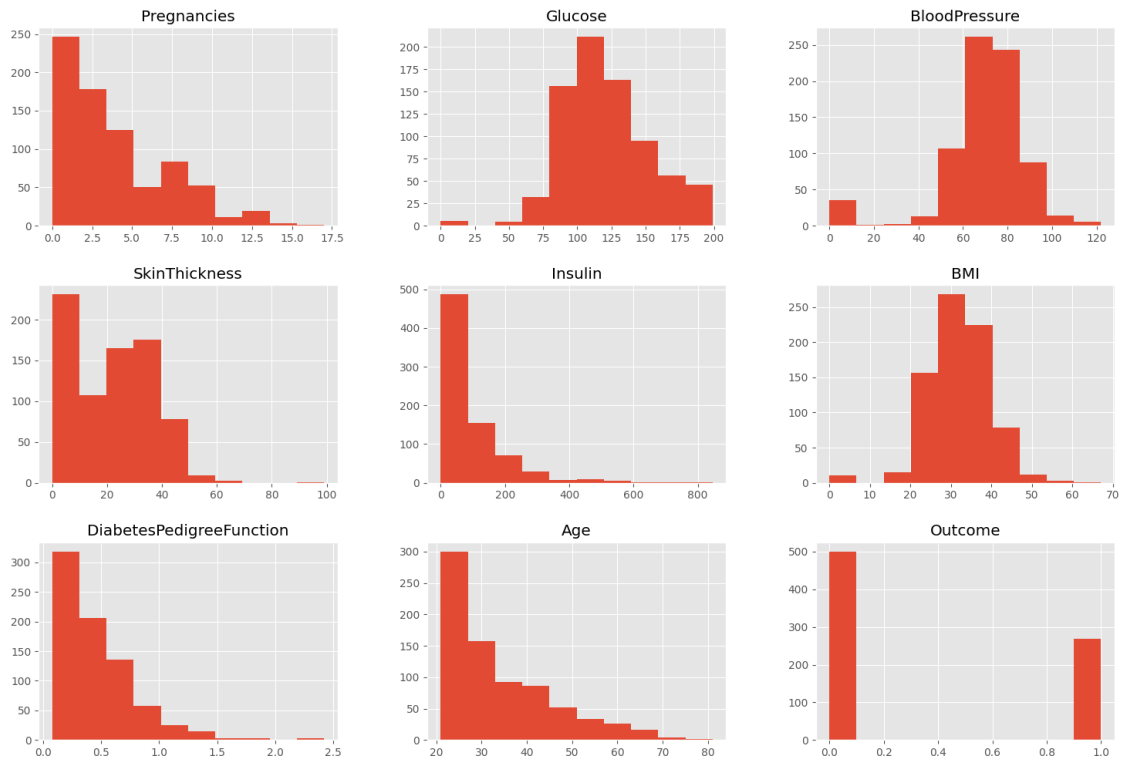
```
[105]: plt.figure(figsize = (12,10))

sns.heatmap(df.corr(), annot = True)
```

```
[105]: <Axes: >
```



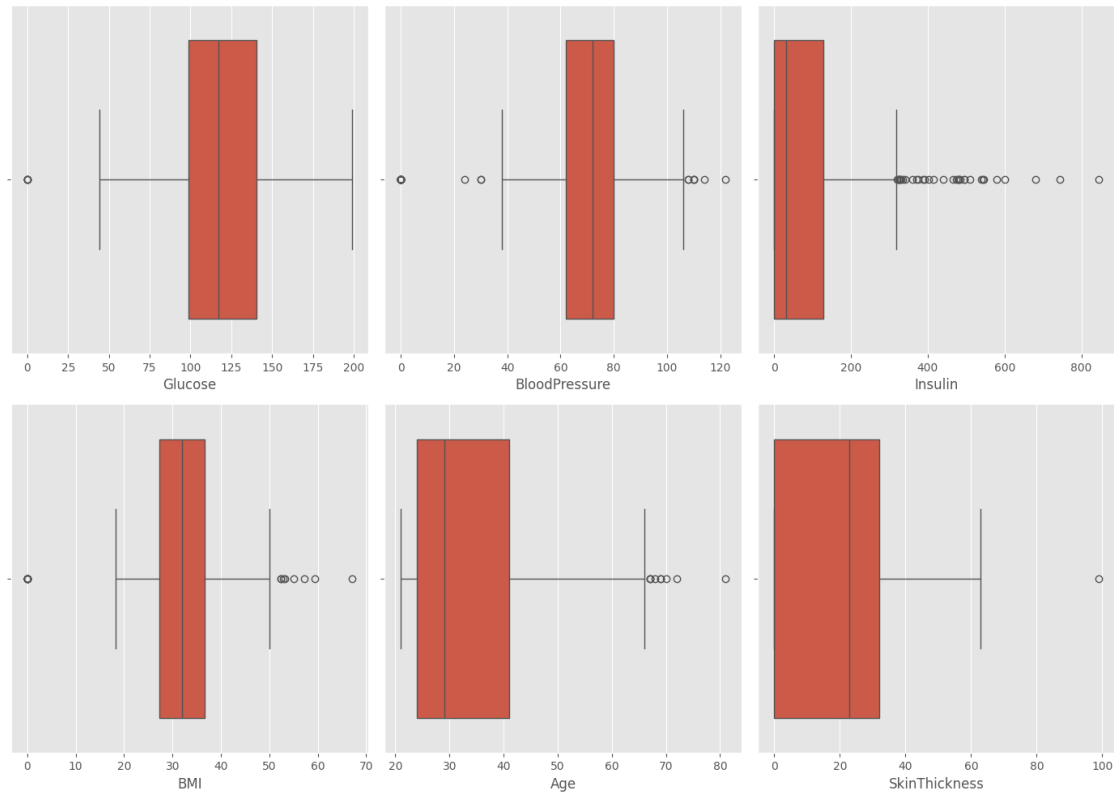
```
[106]: df.hist(figsize=(18,12))
plt.show()
```



```
[107]: features = ['Glucose', 'BloodPressure', 'Insulin', 'BMI', 'Age',
↳ 'SkinThickness']
plt.figure(figsize=(14, 10))

for i, feature in enumerate(features, start=1):
    plt.subplot(2, 3, i)
    sns.boxplot(x=feature, data=df)

plt.tight_layout()
plt.show()
```



```
[108]: mean_col = ['Glucose', 'BloodPressure', 'Insulin', 'Age', 'Outcome', 'BMI']

sns.pairplot(df[mean_col], palette='dark')
```

```
/usr/local/lib/python3.10/dist-packages/seaborn/axisgrid.py:1513: UserWarning:
Ignoring `palette` because no `hue` variable has been assigned.
  func(x=vector, **plot_kwargs)
/usr/local/lib/python3.10/dist-packages/seaborn/axisgrid.py:1513: UserWarning:
Ignoring `palette` because no `hue` variable has been assigned.
  func(x=vector, **plot_kwargs)
/usr/local/lib/python3.10/dist-packages/seaborn/axisgrid.py:1513: UserWarning:
Ignoring `palette` because no `hue` variable has been assigned.
  func(x=vector, **plot_kwargs)
/usr/local/lib/python3.10/dist-packages/seaborn/axisgrid.py:1513: UserWarning:
Ignoring `palette` because no `hue` variable has been assigned.
  func(x=vector, **plot_kwargs)
/usr/local/lib/python3.10/dist-packages/seaborn/axisgrid.py:1513: UserWarning:
Ignoring `palette` because no `hue` variable has been assigned.
  func(x=vector, **plot_kwargs)
```

```

/usr/local/lib/python3.10/dist-packages/seaborn/axisgrid.py:1615: UserWarning:
Ignoring `palette` because no `hue` variable has been assigned.
    func(x=x, y=y, **kwargs)
/usr/local/lib/python3.10/dist-packages/seaborn/axisgrid.py:1615: UserWarning:
Ignoring `palette` because no `hue` variable has been assigned.
    func(x=x, y=y, **kwargs)
/usr/local/lib/python3.10/dist-packages/seaborn/axisgrid.py:1615: UserWarning:
Ignoring `palette` because no `hue` variable has been assigned.
    func(x=x, y=y, **kwargs)
/usr/local/lib/python3.10/dist-packages/seaborn/axisgrid.py:1615: UserWarning:
Ignoring `palette` because no `hue` variable has been assigned.
    func(x=x, y=y, **kwargs)
/usr/local/lib/python3.10/dist-packages/seaborn/axisgrid.py:1615: UserWarning:
Ignoring `palette` because no `hue` variable has been assigned.
    func(x=x, y=y, **kwargs)
/usr/local/lib/python3.10/dist-packages/seaborn/axisgrid.py:1615: UserWarning:
Ignoring `palette` because no `hue` variable has been assigned.
    func(x=x, y=y, **kwargs)
/usr/local/lib/python3.10/dist-packages/seaborn/axisgrid.py:1615: UserWarning:
Ignoring `palette` because no `hue` variable has been assigned.
    func(x=x, y=y, **kwargs)
/usr/local/lib/python3.10/dist-packages/seaborn/axisgrid.py:1615: UserWarning:
Ignoring `palette` because no `hue` variable has been assigned.
    func(x=x, y=y, **kwargs)
/usr/local/lib/python3.10/dist-packages/seaborn/axisgrid.py:1615: UserWarning:
Ignoring `palette` because no `hue` variable has been assigned.
    func(x=x, y=y, **kwargs)
/usr/local/lib/python3.10/dist-packages/seaborn/axisgrid.py:1615: UserWarning:
Ignoring `palette` because no `hue` variable has been assigned.
    func(x=x, y=y, **kwargs)
/usr/local/lib/python3.10/dist-packages/seaborn/axisgrid.py:1615: UserWarning:
Ignoring `palette` because no `hue` variable has been assigned.
    func(x=x, y=y, **kwargs)
/usr/local/lib/python3.10/dist-packages/seaborn/axisgrid.py:1615: UserWarning:
Ignoring `palette` because no `hue` variable has been assigned.
    func(x=x, y=y, **kwargs)
/usr/local/lib/python3.10/dist-packages/seaborn/axisgrid.py:1615: UserWarning:
Ignoring `palette` because no `hue` variable has been assigned.
    func(x=x, y=y, **kwargs)

```



```

/usr/local/lib/python3.10/dist-packages/seaborn/axisgrid.py:1615: UserWarning:
Ignoring `palette` because no `hue` variable has been assigned.
    func(x=x, y=y, **kwargs)
/usr/local/lib/python3.10/dist-packages/seaborn/axisgrid.py:1615: UserWarning:
Ignoring `palette` because no `hue` variable has been assigned.
    func(x=x, y=y, **kwargs)
/usr/local/lib/python3.10/dist-packages/seaborn/axisgrid.py:1615: UserWarning:
Ignoring `palette` because no `hue` variable has been assigned.
    func(x=x, y=y, **kwargs)
/usr/local/lib/python3.10/dist-packages/seaborn/axisgrid.py:1615: UserWarning:
Ignoring `palette` because no `hue` variable has been assigned.
    func(x=x, y=y, **kwargs)
/usr/local/lib/python3.10/dist-packages/seaborn/axisgrid.py:1615: UserWarning:
Ignoring `palette` because no `hue` variable has been assigned.
    func(x=x, y=y, **kwargs)
/usr/local/lib/python3.10/dist-packages/seaborn/axisgrid.py:1615: UserWarning:
Ignoring `palette` because no `hue` variable has been assigned.
    func(x=x, y=y, **kwargs)
/usr/local/lib/python3.10/dist-packages/seaborn/axisgrid.py:1615: UserWarning:
Ignoring `palette` because no `hue` variable has been assigned.
    func(x=x, y=y, **kwargs)
/usr/local/lib/python3.10/dist-packages/seaborn/axisgrid.py:1615: UserWarning:
Ignoring `palette` because no `hue` variable has been assigned.
    func(x=x, y=y, **kwargs)
/usr/local/lib/python3.10/dist-packages/seaborn/axisgrid.py:1615: UserWarning:
Ignoring `palette` because no `hue` variable has been assigned.
    func(x=x, y=y, **kwargs)
/usr/local/lib/python3.10/dist-packages/seaborn/axisgrid.py:1615: UserWarning:
Ignoring `palette` because no `hue` variable has been assigned.
    func(x=x, y=y, **kwargs)
/usr/local/lib/python3.10/dist-packages/seaborn/axisgrid.py:1615: UserWarning:
Ignoring `palette` because no `hue` variable has been assigned.
    func(x=x, y=y, **kwargs)

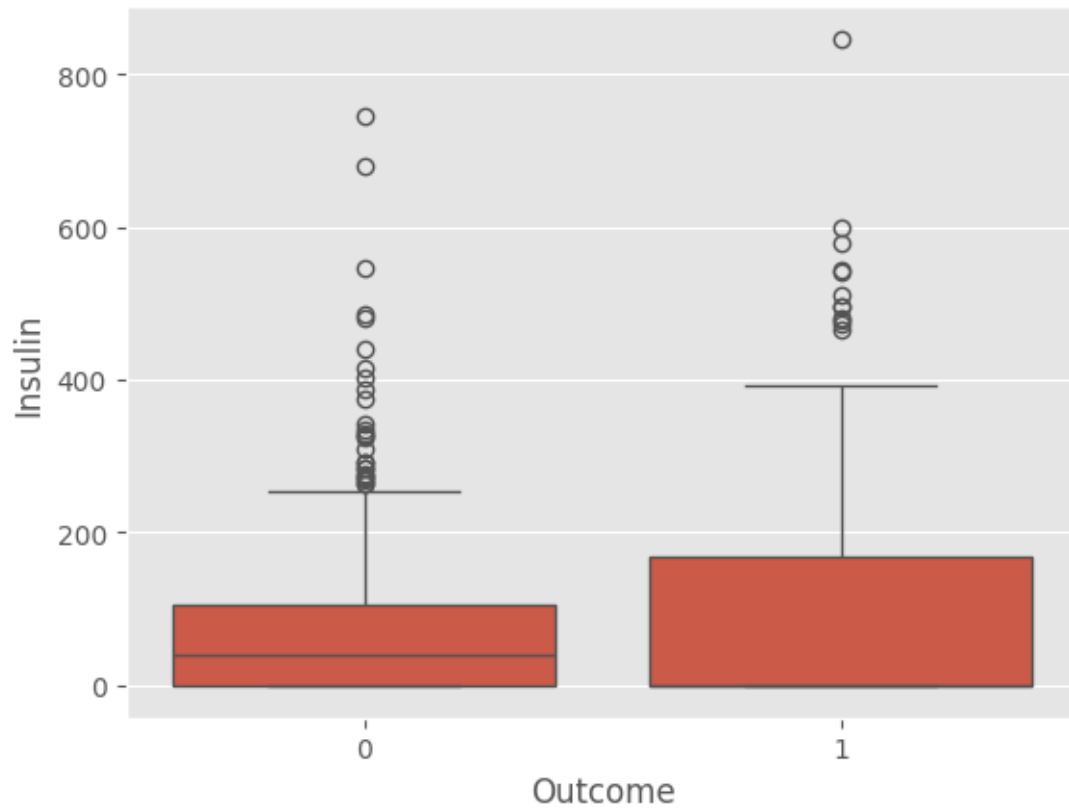
```

[108]: <seaborn.axisgrid.PairGrid at 0x7deae4c86650>



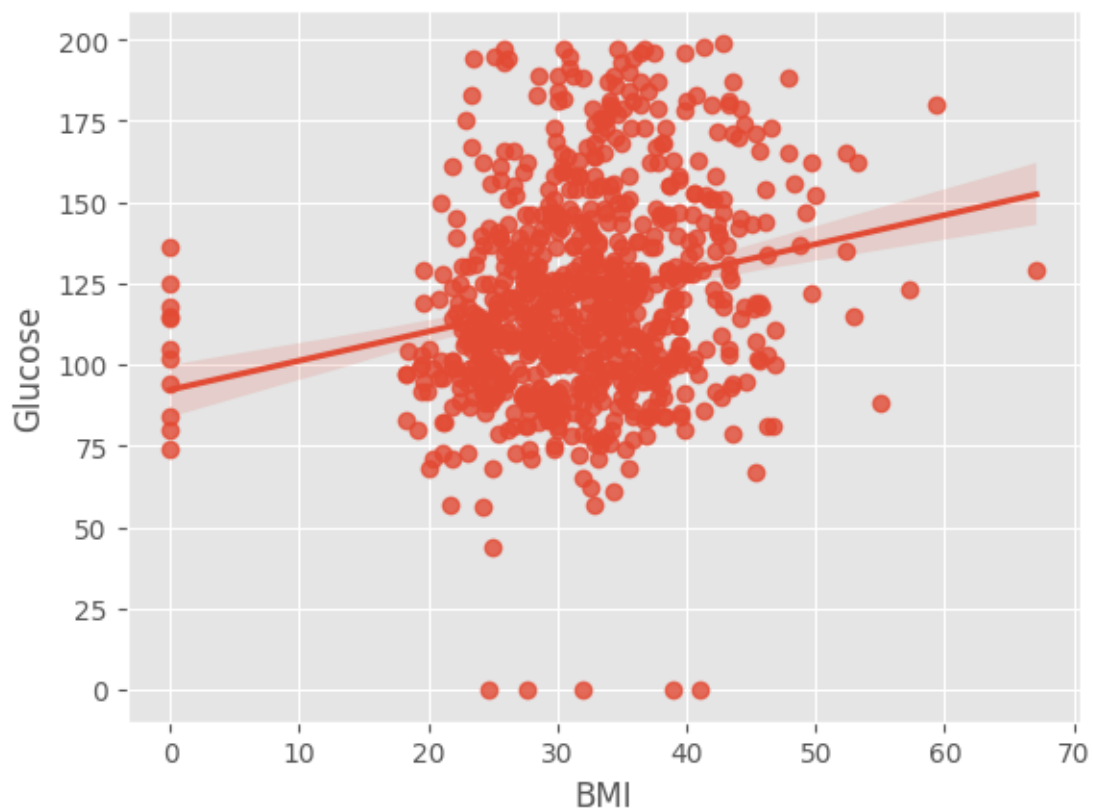
```
[109]: sns.boxplot(x='Outcome',y='Insulin',data=df)
```

```
[109]: <Axes: xlabel='Outcome', ylabel='Insulin'>
```



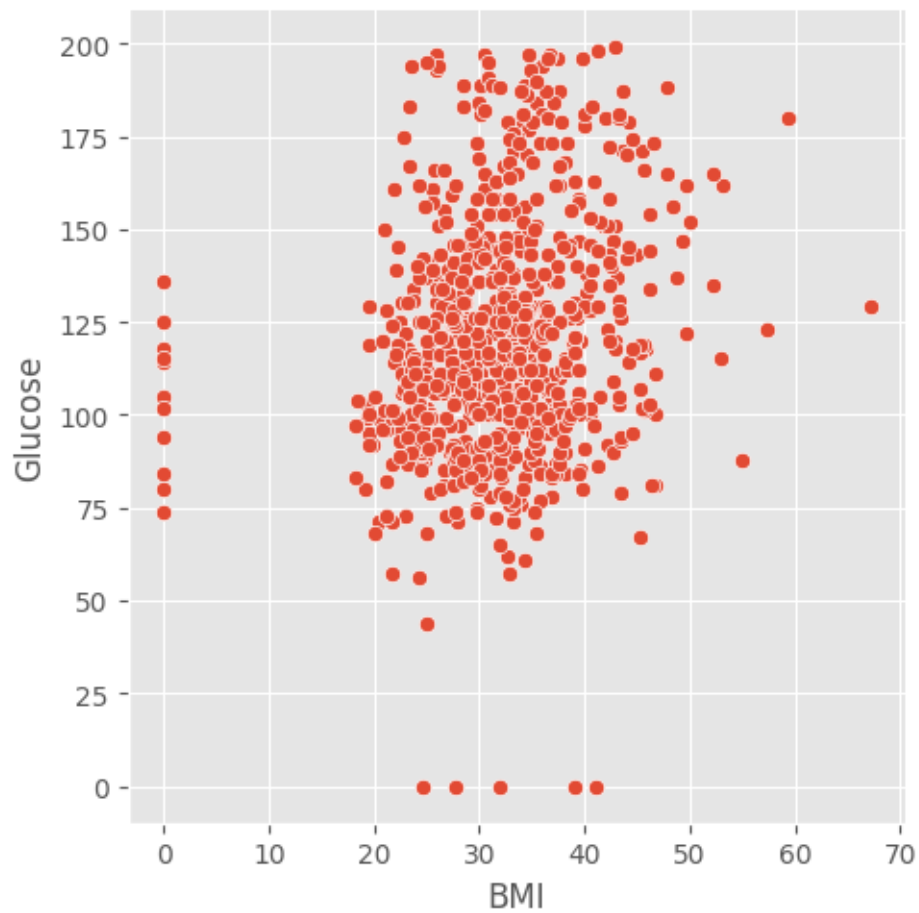
```
[110]: sns.regplot(x='BMI', y= 'Glucose', data=df)
```

```
[110]: <Axes: xlabel='BMI', ylabel='Glucose'>
```



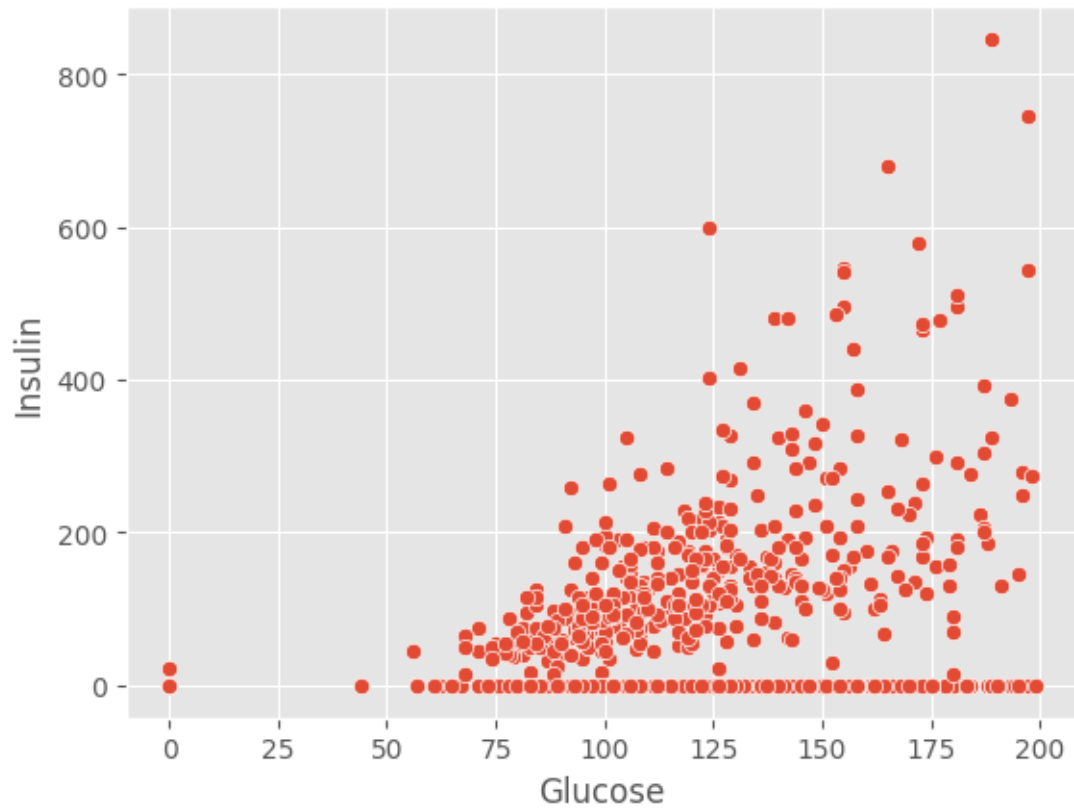
```
[111]: sns.relplot(x='BMI', y= 'Glucose', data=df)
```

```
[111]: <seaborn.axisgrid.FacetGrid at 0x7deadfba97b0>
```



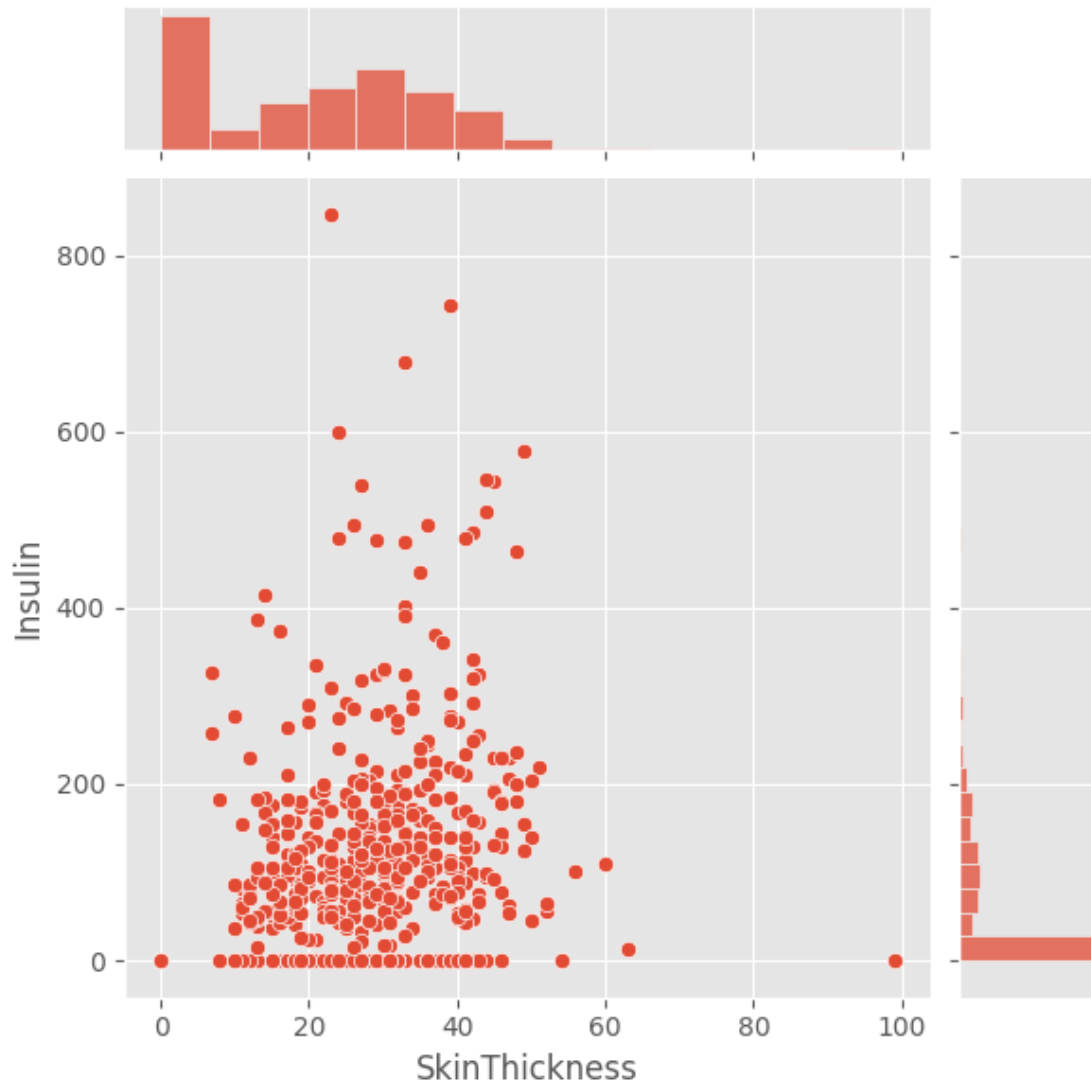
```
[112]: sns.scatterplot(x='Glucose', y= 'Insulin', data=df)
```

```
[112]: <Axes: xlabel='Glucose', ylabel='Insulin'>
```



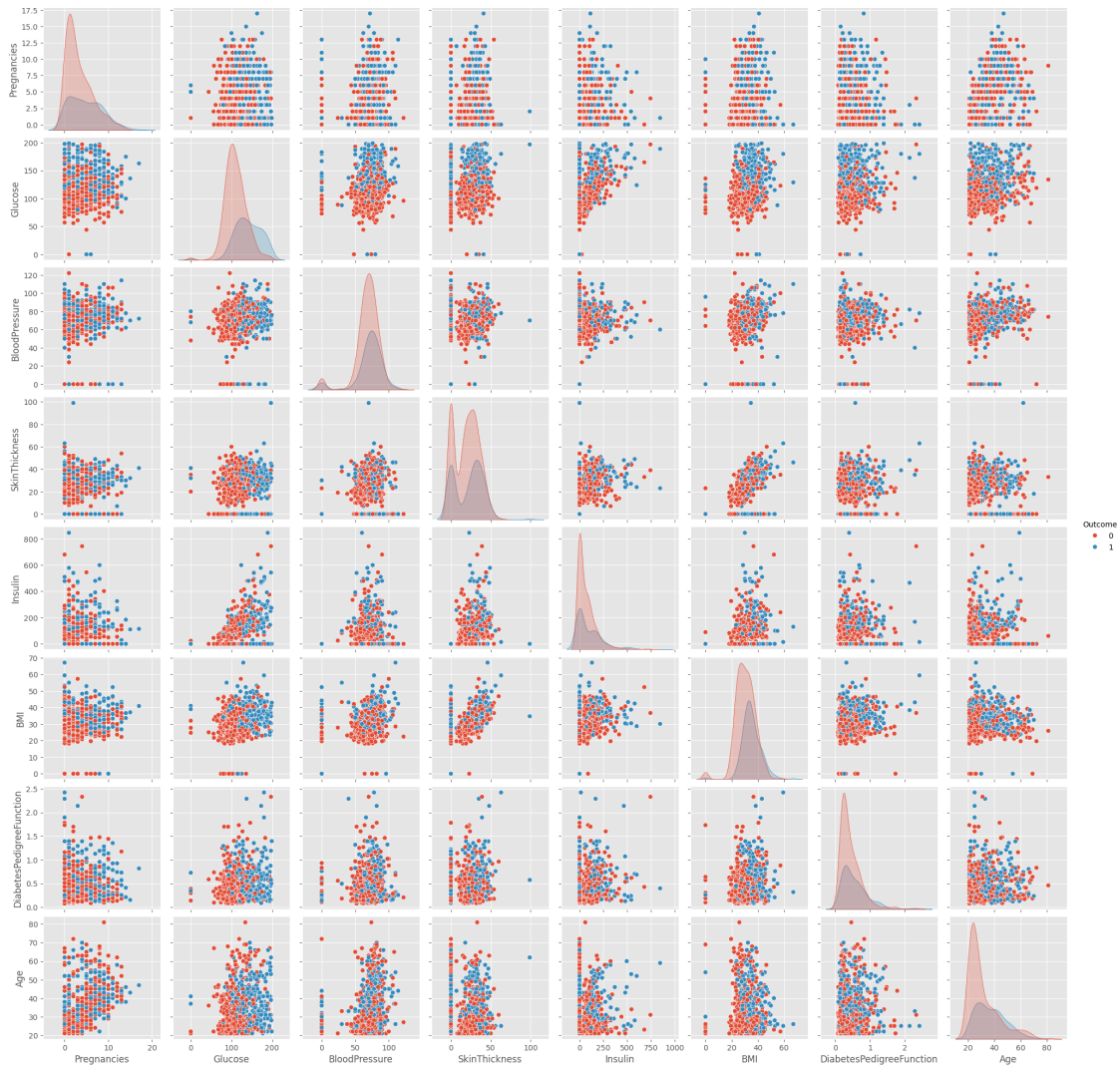
```
[113]: sns.jointplot(x='SkinThickness', y= 'Insulin', data=df)
```

```
[113]: <seaborn.axisgrid.JointGrid at 0x7deadfa2cbe0>
```



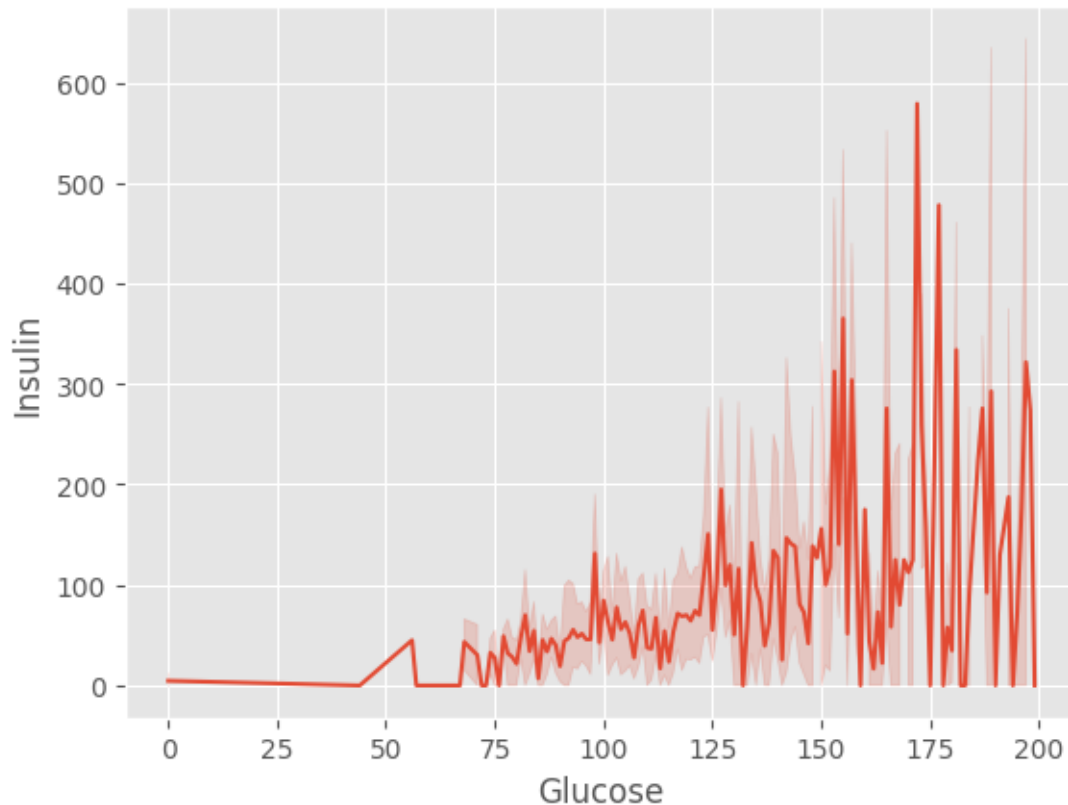
```
[114]: sns.pairplot(df,hue='Outcome')
```

```
[114]: <seaborn.axisgrid.PairGrid at 0x7deadf9c3d60>
```



```
[115]: sns.lineplot(x='Glucose', y= 'Insulin', data=df)
```

```
[115]: <Axes: xlabel='Glucose', ylabel='Insulin'>
```

```
[116]: sns.swarmplot(x='Glucose', y= 'Insulin', data=df)
```

```
/usr/local/lib/python3.10/dist-packages/seaborn/categorical.py:3398:
UserWarning: 60.0% of the points cannot be placed; you may want to decrease the
size of the markers or use stripplot.
  warnings.warn(msg, UserWarning)
/usr/local/lib/python3.10/dist-packages/seaborn/categorical.py:3398:
UserWarning: 50.0% of the points cannot be placed; you may want to decrease the
size of the markers or use stripplot.
  warnings.warn(msg, UserWarning)
/usr/local/lib/python3.10/dist-packages/seaborn/categorical.py:3398:
UserWarning: 33.3% of the points cannot be placed; you may want to decrease the
size of the markers or use stripplot.
  warnings.warn(msg, UserWarning)
/usr/local/lib/python3.10/dist-packages/seaborn/categorical.py:3398:
UserWarning: 25.0% of the points cannot be placed; you may want to decrease the
size of the markers or use stripplot.
  warnings.warn(msg, UserWarning)
/usr/local/lib/python3.10/dist-packages/seaborn/categorical.py:3398:
UserWarning: 66.7% of the points cannot be placed; you may want to decrease the
size of the markers or use stripplot.
```

```

warnings.warn(msg, UserWarning)
/usr/local/lib/python3.10/dist-packages/seaborn/categorical.py:3398:
UserWarning: 71.4% of the points cannot be placed; you may want to decrease the
size of the markers or use stripplot.
warnings.warn(msg, UserWarning)
/usr/local/lib/python3.10/dist-packages/seaborn/categorical.py:3398:
UserWarning: 42.9% of the points cannot be placed; you may want to decrease the
size of the markers or use stripplot.
warnings.warn(msg, UserWarning)
/usr/local/lib/python3.10/dist-packages/seaborn/categorical.py:3398:
UserWarning: 55.6% of the points cannot be placed; you may want to decrease the
size of the markers or use stripplot.
warnings.warn(msg, UserWarning)
/usr/local/lib/python3.10/dist-packages/seaborn/categorical.py:3398:
UserWarning: 81.8% of the points cannot be placed; you may want to decrease the
size of the markers or use stripplot.
warnings.warn(msg, UserWarning)
/usr/local/lib/python3.10/dist-packages/seaborn/categorical.py:3398:
UserWarning: 57.1% of the points cannot be placed; you may want to decrease the
size of the markers or use stripplot.
warnings.warn(msg, UserWarning)
/usr/local/lib/python3.10/dist-packages/seaborn/categorical.py:3398:
UserWarning: 61.5% of the points cannot be placed; you may want to decrease the
size of the markers or use stripplot.
warnings.warn(msg, UserWarning)
/usr/local/lib/python3.10/dist-packages/seaborn/categorical.py:3398:
UserWarning: 37.5% of the points cannot be placed; you may want to decrease the
size of the markers or use stripplot.
warnings.warn(msg, UserWarning)
/usr/local/lib/python3.10/dist-packages/seaborn/categorical.py:3398:
UserWarning: 64.7% of the points cannot be placed; you may want to decrease the
size of the markers or use stripplot.
warnings.warn(msg, UserWarning)
/usr/local/lib/python3.10/dist-packages/seaborn/categorical.py:3398:
UserWarning: 44.4% of the points cannot be placed; you may want to decrease the
size of the markers or use stripplot.
warnings.warn(msg, UserWarning)
/usr/local/lib/python3.10/dist-packages/seaborn/categorical.py:3398:
UserWarning: 76.9% of the points cannot be placed; you may want to decrease the
size of the markers or use stripplot.
warnings.warn(msg, UserWarning)
/usr/local/lib/python3.10/dist-packages/seaborn/categorical.py:3398:
UserWarning: 53.8% of the points cannot be placed; you may want to decrease the
size of the markers or use stripplot.
warnings.warn(msg, UserWarning)
/usr/local/lib/python3.10/dist-packages/seaborn/categorical.py:3398:
UserWarning: 85.7% of the points cannot be placed; you may want to decrease the
size of the markers or use stripplot.

```

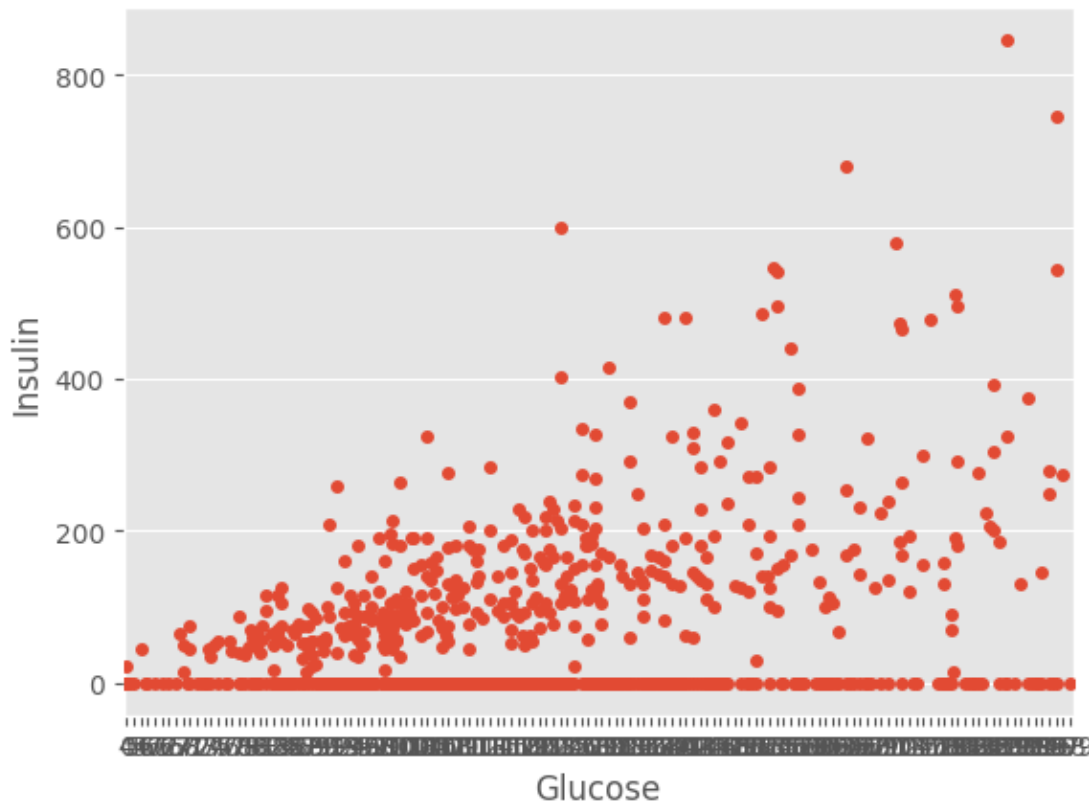
```

warnings.warn(msg, UserWarning)
/usr/local/lib/python3.10/dist-packages/seaborn/categorical.py:3398:
UserWarning: 63.6% of the points cannot be placed; you may want to decrease the
size of the markers or use stripplot.
warnings.warn(msg, UserWarning)
/usr/local/lib/python3.10/dist-packages/seaborn/categorical.py:3398:
UserWarning: 64.3% of the points cannot be placed; you may want to decrease the
size of the markers or use stripplot.
warnings.warn(msg, UserWarning)
/usr/local/lib/python3.10/dist-packages/seaborn/categorical.py:3398:
UserWarning: 69.2% of the points cannot be placed; you may want to decrease the
size of the markers or use stripplot.
warnings.warn(msg, UserWarning)
/usr/local/lib/python3.10/dist-packages/seaborn/categorical.py:3398:
UserWarning: 70.0% of the points cannot be placed; you may want to decrease the
size of the markers or use stripplot.
warnings.warn(msg, UserWarning)
/usr/local/lib/python3.10/dist-packages/seaborn/categorical.py:3398:
UserWarning: 45.5% of the points cannot be placed; you may want to decrease the
size of the markers or use stripplot.
warnings.warn(msg, UserWarning)
/usr/local/lib/python3.10/dist-packages/seaborn/categorical.py:3398:
UserWarning: 54.5% of the points cannot be placed; you may want to decrease the
size of the markers or use stripplot.
warnings.warn(msg, UserWarning)
/usr/local/lib/python3.10/dist-packages/seaborn/categorical.py:3398:
UserWarning: 58.3% of the points cannot be placed; you may want to decrease the
size of the markers or use stripplot.
warnings.warn(msg, UserWarning)
/usr/local/lib/python3.10/dist-packages/seaborn/categorical.py:3398:
UserWarning: 22.2% of the points cannot be placed; you may want to decrease the
size of the markers or use stripplot.
warnings.warn(msg, UserWarning)
/usr/local/lib/python3.10/dist-packages/seaborn/categorical.py:3398:
UserWarning: 40.0% of the points cannot be placed; you may want to decrease the
size of the markers or use stripplot.
warnings.warn(msg, UserWarning)
/usr/local/lib/python3.10/dist-packages/seaborn/categorical.py:3398:
UserWarning: 80.0% of the points cannot be placed; you may want to decrease the
size of the markers or use stripplot.
warnings.warn(msg, UserWarning)
/usr/local/lib/python3.10/dist-packages/seaborn/categorical.py:3398:
UserWarning: 16.7% of the points cannot be placed; you may want to decrease the
size of the markers or use stripplot.
warnings.warn(msg, UserWarning)
/usr/local/lib/python3.10/dist-packages/seaborn/categorical.py:3398:
UserWarning: 62.5% of the points cannot be placed; you may want to decrease the
size of the markers or use stripplot.

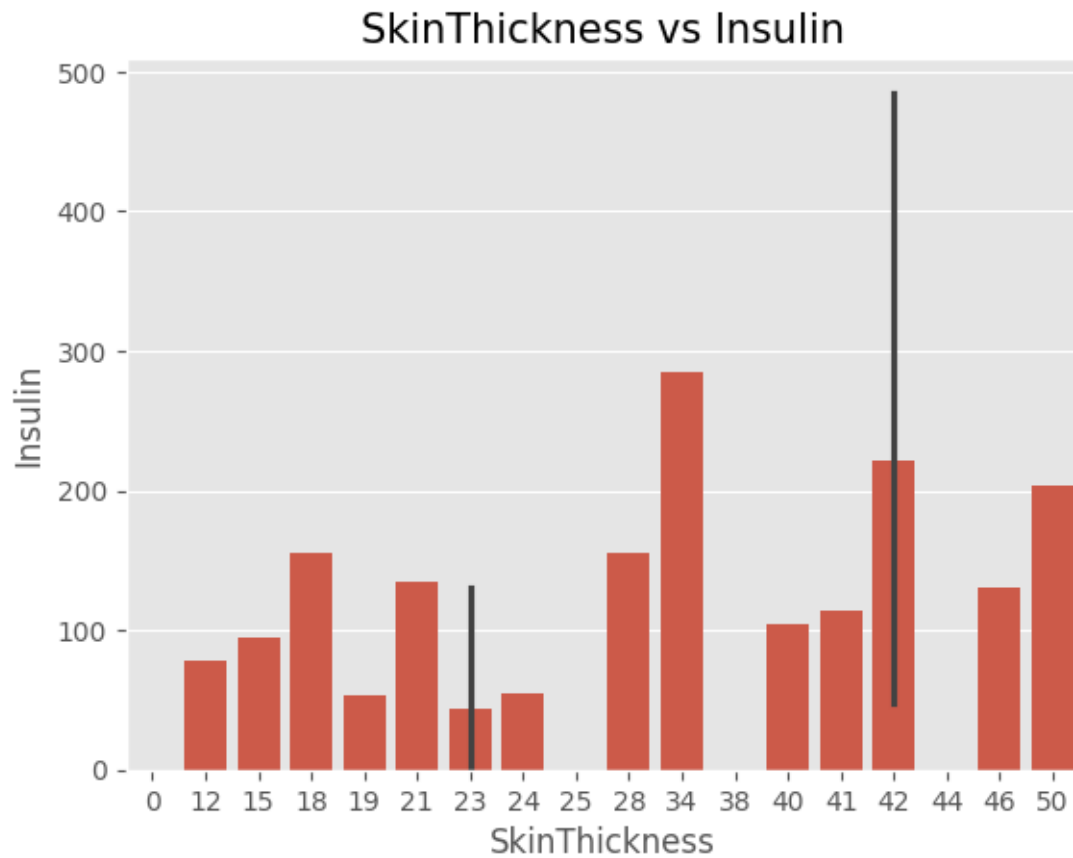
```

```
warnings.warn(msg, UserWarning)
/usr/local/lib/python3.10/dist-packages/seaborn/categorical.py:3398:
UserWarning: 20.0% of the points cannot be placed; you may want to decrease the
size of the markers or use stripplot.
warnings.warn(msg, UserWarning)
/usr/local/lib/python3.10/dist-packages/seaborn/categorical.py:3398:
UserWarning: 28.6% of the points cannot be placed; you may want to decrease the
size of the markers or use stripplot.
warnings.warn(msg, UserWarning)
```

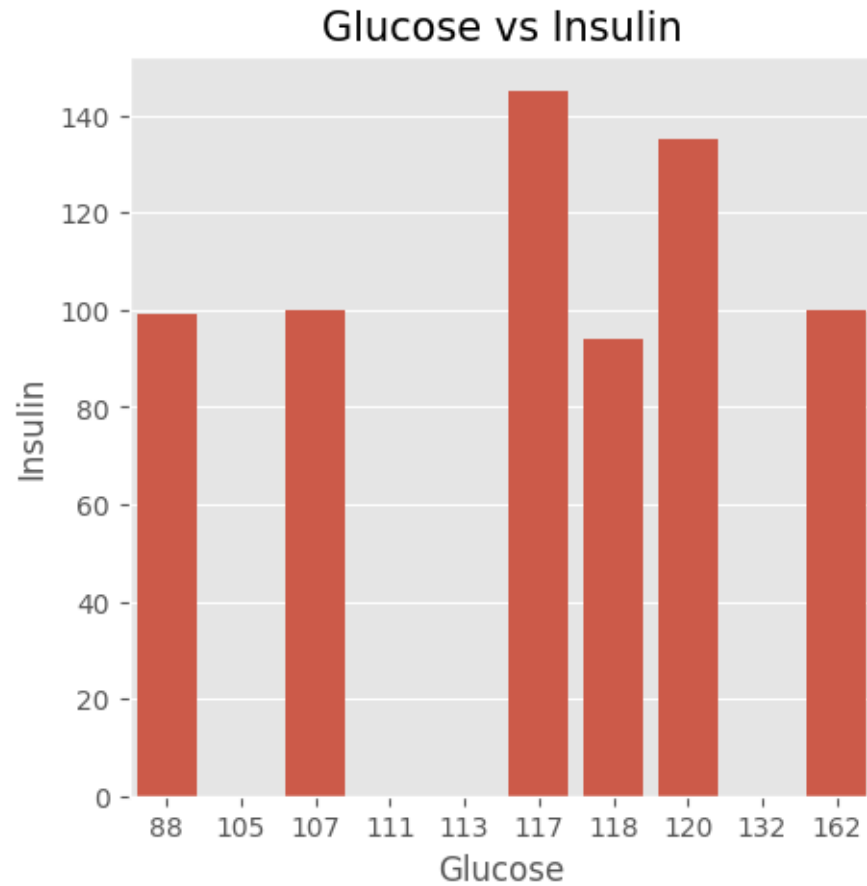
```
[116]: <Axes: xlabel='Glucose', ylabel='Insulin'>
```



```
[117]: sns.barplot(x="SkinThickness", y="Insulin", data=df[150:180])
plt.title("SkinThickness vs Insulin",fontsize=15)
plt.xlabel("SkinThickness")
plt.ylabel("Insulin")
plt.show()
plt.style.use("ggplot")
```



```
[118]: plt.figure(figsize=(5,5))
sns.barplot(x="Glucose", y="Insulin", data=df[120:130])
plt.title("Glucose vs Insulin",fontsize=15)
plt.xlabel("Glucose")
plt.ylabel("Insulin")
plt.show()
```



Training and Testing Data

```
[119]: x = df.drop(columns = 'Outcome')

y = df['Outcome']

from sklearn.model_selection import train_test_split
X_train,X_test,y_train,y_test = train_test_split(x,y,test_size=0.
↪2,random_state=0)
```

MODELS

1. Logistic Regression

```
[120]: from sklearn.linear_model import LogisticRegression
model = LogisticRegression()
model.fit(X_train, y_train)

y_pred = model.predict(X_test)
```

```

print(classification_report(y_test, y_pred))
print(confusion_matrix(y_test, y_pred))

from sklearn.metrics import accuracy_score
LRAcc = accuracy_score(y_pred, y_test)
print('Logistic Regression accuracy is: {:.2f}%'.format(LRAcc*100))

```

	precision	recall	f1-score	support
0	0.84	0.92	0.88	107
1	0.76	0.62	0.68	47
accuracy			0.82	154
macro avg	0.80	0.77	0.78	154
weighted avg	0.82	0.82	0.82	154

```

[[98  9]
 [18 29]]

```

Logistic Regression accuracy is: 82.47%

/usr/local/lib/python3.10/dist-packages/sklearn/linear_model/_logistic.py:458:

ConvergenceWarning: lbfgs failed to converge (status=1):

STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:

<https://scikit-learn.org/stable/modules/preprocessing.html>

Please also refer to the documentation for alternative solver options:

https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression

```
n_iter_i = _check_optimize_result(
```

2. KNeighborsClassifier

```

[121]: from sklearn.neighbors import KNeighborsClassifier
model = KNeighborsClassifier(n_neighbors=7)
model.fit(X_train, y_train)

y_pred = model.predict(X_test)

print(classification_report(y_test, y_pred))
print(confusion_matrix(y_test, y_pred))

from sklearn.metrics import accuracy_score
KNAcc = accuracy_score(y_pred, y_test)
print('KNeighborsClassifier accuracy is: {:.2f}%'.format(KNAcc*100))

```

	precision	recall	f1-score	support
0	0.82	0.84	0.83	107

	1	0.61	0.57	0.59	47
accuracy				0.76	154
macro avg		0.72	0.71	0.71	154
weighted avg		0.76	0.76	0.76	154

```
[[90 17]
 [20 27]]
```

KNeighborsClassifier accuracy is: 75.97%

3. SVC

```
[122]: from sklearn.svm import SVC
model = SVC()
model.fit(X_train, y_train)

y_pred = model.predict(X_test)

print(classification_report(y_test, y_pred))
print(confusion_matrix(y_test, y_pred))

from sklearn.metrics import accuracy_score
SVCAcc = accuracy_score(y_pred, y_test)
print('SVC accuracy is: {:.2f}%'.format(SVCAcc*100))
```

		precision	recall	f1-score	support
	0	0.81	0.92	0.86	107
	1	0.73	0.51	0.60	47
accuracy				0.79	154
macro avg		0.77	0.71	0.73	154
weighted avg		0.78	0.79	0.78	154

```
[[98  9]
 [23 24]]
```

SVC accuracy is: 79.22%

4. RandomForestClassifier

```
[123]: from sklearn.ensemble import RandomForestClassifier
model = RandomForestClassifier()
model.fit(X_train, y_train)

y_pred = model.predict(X_test)

print(classification_report(y_test, y_pred))
print(confusion_matrix(y_test, y_pred))
```



```

from sklearn.metrics import accuracy_score
RFAcc = accuracy_score(y_pred,y_test)
print('RFC accuracy is: {:.2f}%'.format(RFAcc*100))

```

	precision	recall	f1-score	support
0	0.85	0.88	0.87	107
1	0.70	0.66	0.68	47
accuracy			0.81	154
macro avg	0.78	0.77	0.77	154
weighted avg	0.81	0.81	0.81	154

```

[[94 13]
 [16 31]]
RFC accuracy is: 81.17%

```

5. Gradient Boosting Classifier

```

[124]: from sklearn.ensemble import GradientBoostingClassifier
model = GradientBoostingClassifier()
model.fit(X_train, y_train)

y_pred = model.predict(X_test)

print(classification_report(y_test, y_pred))
print(confusion_matrix(y_test, y_pred))

from sklearn.metrics import accuracy_score
GBCAcc = accuracy_score(y_pred,y_test)
print('GBC accuracy is: {:.2f}%'.format(GBCAcc*100))

```

	precision	recall	f1-score	support
0	0.87	0.87	0.87	107
1	0.70	0.70	0.70	47
accuracy			0.82	154
macro avg	0.79	0.79	0.79	154
weighted avg	0.82	0.82	0.82	154

```

[[93 14]
 [14 33]]
GBC accuracy is: 81.82%

```

6. Naive Bayes

```
[125]: from sklearn.naive_bayes import GaussianNB
model = GaussianNB()
model.fit(X_train, y_train)

y_pred = model.predict(X_test)

print(classification_report(y_test, y_pred))
print(confusion_matrix(y_test, y_pred))

from sklearn.metrics import accuracy_score
GNBAcc = accuracy_score(y_pred, y_test)
print('GNB accuracy is: {:.2f}%'.format(GNBAcc*100))
```

	precision	recall	f1-score	support
0	0.84	0.87	0.85	107
1	0.67	0.62	0.64	47
accuracy			0.79	154
macro avg	0.76	0.74	0.75	154
weighted avg	0.79	0.79	0.79	154

```
[[93 14]
 [18 29]]
GNB accuracy is: 79.22%
```

Compare Models

```
[126]: compare = pd.DataFrame({'Model': ['Logistic Regression', 'K Neighbors', 'SVM', 'Random Forest', 'GradientBoostingClassifier', 'GaussianNB'],
                               'Accuracy': [LRAcc*100, KNAcc*100, SVCAcc*100, RFAcc*100, GBCAcc*100, GNBAcc*100]})
compare.sort_values(by='Accuracy', ascending=False)
```

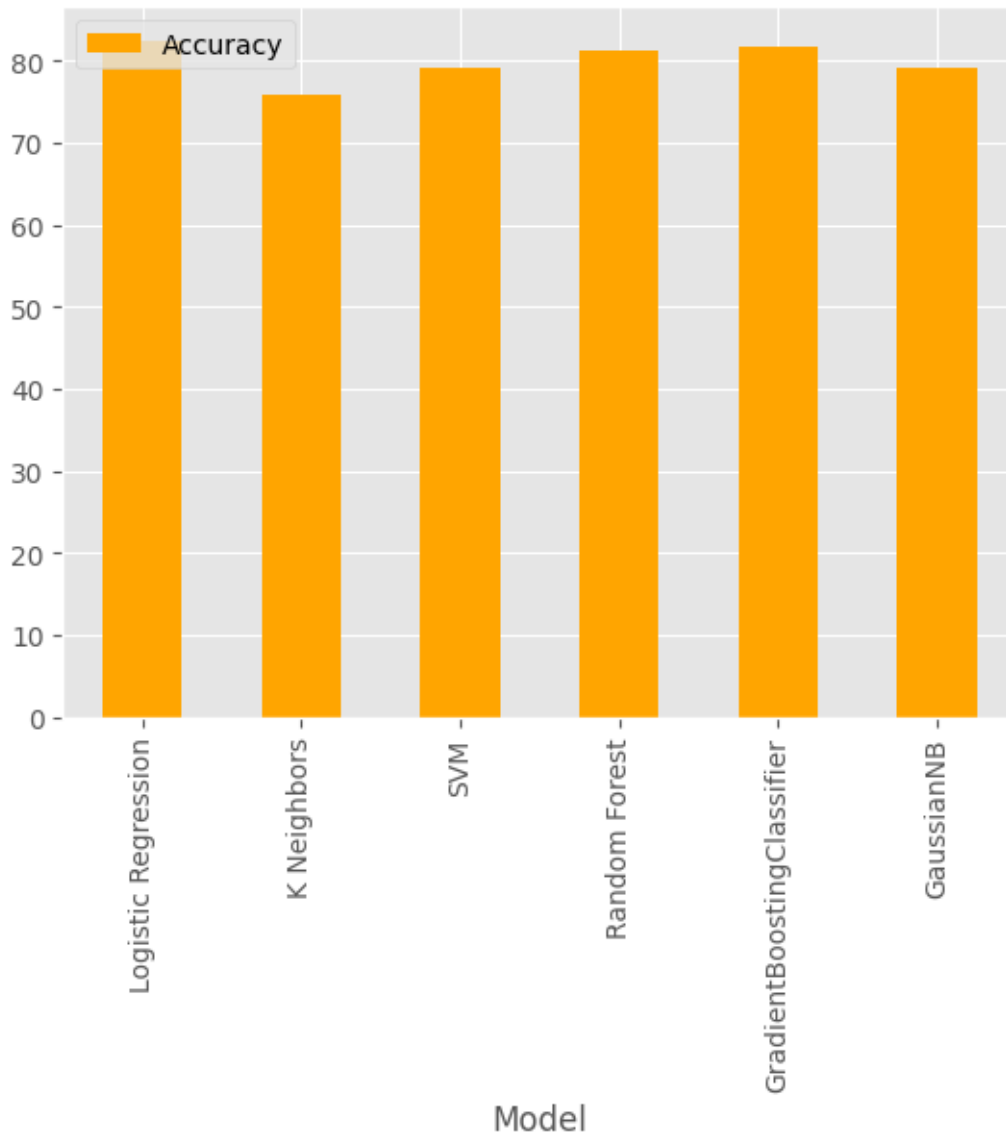
```
[126]:
```

	Model	Accuracy
0	Logistic Regression	82.467532
4	GradientBoostingClassifier	81.818182
3	Random Forest	81.168831
2	SVM	79.220779
5	GaussianNB	79.220779
1	K Neighbors	75.974026

Plotting Model Comparison

```
[127]: compare.plot(x='Model', y='Accuracy', kind='bar', color='orange')
```

```
[127]: <Axes: xlabel='Model'>
```



From the comparison plot, among the 6 ML models, Logistic Regression had achieved the highest accuracy of 82.50%.