Assignment 02: Evaluate the Diabetes Dataset

The comments/sections provided are your cues to perform the assignment. You don't need to limit yourself to the number of rows/cells provided. You can add additional rows in each section to add more lines of code.

If at any point in time you need help on solving this assignment, view our demo video to understand the different steps of the code.

Happy coding!

1: Import the dataset

In [1]:

```
#Import the required libraries
import pandas as pd
```

In [3]:

```
#Import the diabetes dataset
df=pd.read_csv("pima-indians-diabetes.data",header=None)
df
```

Out[3]:

	0	1	2	3	4	5	6	7	8
0	6	148	72	35	0	33.6	0.627	50	1
1	1	85	66	29	0	26.6	0.351	31	0
2	8	183	64	0	0	23.3	0.672	32	1
3	1	89	66	23	94	28.1	0.167	21	0
4	0	137	40	35	168	43.1	2.288	33	1
...	...	...	...	...	...	...	...	...	...
763	10	101	76	48	180	32.9	0.171	63	0
764	2	122	70	27	0	36.8	0.340	27	0
765	5	121	72	23	112	26.2	0.245	30	0
766	1	126	60	0	0	30.1	0.349	47	1
767	1	93	70	31	0	30.4	0.315	23	0

768 rows × 9 columns

2: Analyze the dataset

In [4]:

```
#View the first five observations of the dataset
df.head()
```

Out[4]:

	0	1	2	3	4	5	6	7	8
0	6	148	72	35	0	33.6	0.627	50	1
1	1	85	66	29	0	26.6	0.351	31	0

2	8	183	64	0	0	23.3	0.672	32	1
3	1	89	66	23	94	28.1	0.167	21	0
4	0	137	40	35	168	43.1	2.288	33	1

3: Find the features of the dataset

In [49]:

```
#Use the .NAMES file to view and set the features of the dataset
feature_names=["pregnancy","glucose","bp","skin","insuline","bmi","pedigree","age","label
"]
```

In [50]:

```
#Use the feature names set earlier and fix it as the column headers of the dataset
df=pd.read_csv("pima-indians-diabetes.data",header=None,names=feature_names)
df
```

Out[50]:

	pregnancy	glucose	bp	skin	insuline	bmi	pedigree	age	label
0	6	148	72	35	0	33.6	0.627	50	1
1	1	85	66	29	0	26.6	0.351	31	0
2	8	183	64	0	0	23.3	0.672	32	1
3	1	89	66	23	94	28.1	0.167	21	0
4	0	137	40	35	168	43.1	2.288	33	1
...	...	...	...	...	...	...	...	...	...
763	10	101	76	48	180	32.9	0.171	63	0
764	2	122	70	27	0	36.8	0.340	27	0
765	5	121	72	23	112	26.2	0.245	30	0
766	1	126	60	0	0	30.1	0.349	47	1
767	1	93	70	31	0	30.4	0.315	23	0

768 rows × 9 columns

In [51]:

```
#Verify if the dataset is updated with the new headers
df.head()
```

Out[51]:

	pregnancy	glucose	bp	skin	insuline	bmi	pedigree	age	label
0	6	148	72	35	0	33.6	0.627	50	1
1	1	85	66	29	0	26.6	0.351	31	0
2	8	183	64	0	0	23.3	0.672	32	1
3	1	89	66	23	94	28.1	0.167	21	0
4	0	137	40	35	168	43.1	2.288	33	1

In [52]:

```
#View the number of observations and features of the dataset
df.shape
```

Out[52]:

```
(768, 9)
```

4: Find the response of the dataset

In [53]:

```
#Select features from the dataset to create the model
feature_select_cols=df[["pregnancy","insuline","bmi","age"]]
feature_select_cols
```

Out[53]:

	pregnancy	insuline	bmi	age
0	6	0	33.6	50
1	1	0	26.6	31
2	8	0	23.3	32
3	1	94	28.1	21
4	0	168	43.1	33
...	...	...	...	...
763	10	180	32.9	63
764	2	0	36.8	27
765	5	112	26.2	30
766	1	0	30.1	47
767	1	0	30.4	23

768 rows × 4 columns

In [97]:

```
#Create the feature objecte()
x_feature=df[["pregnancy","insuline","bmi","age"]]
```

In [99]:

```
#Create the reponse object
y_target=df["label"]
```

In [98]:

```
#View the shape of the feature object
x_feature.shape
```

Out[98]:

```
(768, 4)
```

In [87]:

```
#View the shape of the target object
y_target.shape
```

Out[87]:

```
(768,)
```

5: Use training and testing datasets to train the model

In [100]:

```
#Split the dataset to test and train the model
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x_feature,y_target,random_state=1)
```

6: Create a model to predict the diabetes outcome

In [102]:

```
# Create a logistic regression model using the training set
from sklearn.linear_model import LogisticRegression
logReg=LogisticRegression()
logReg.fit(x_train,y_train)
```

Out[102]:

```
LogisticRegression()
```

In [104]:

```
#Make predictions using the testing set
y_pred=logReg.predict(x_test)
```

7: Check the accuracy of the model

In [106]:

```
#Evaluate the accuracy of your model
from sklearn import metrics
print(metrics.accuracy_score(y_test,y_pred))
```

```
0.6927083333333334
```

In [107]:

```
#Print the first 30 actual and predicted responses
print("actual values:",y_test.values[0:30])
```

```
actual values: [0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 1 1 0 0 0 1 1 1 1 0 0 0 1 0 1]
```

In [108]:

```
print("Successfully completed a Project on Diabetes dataset")
```

```
Successfully completed a Project on Diabetes dataset
```

In [109]:

```
print("Thank you Simplilearn")
```

```
Thank you Simplilearn
```

In [ ]:

```

```