

```
In [1]: pip install pandas
```

Requirement already satisfied: pandas in c:\users\reddappa reddy\anaconda3\lib\site-packages (2.0.3)
Requirement already satisfied: python-dateutil>=2.8.2 in c:\users\reddappa reddy\anaconda3\lib\site-packages (from pandas) (2.8.2)
Requirement already satisfied: pytz>=2020.1 in c:\users\reddappa reddy\anaconda3\lib\site-packages (from pandas) (2023.3.post1)
Requirement already satisfied: tzdata>=2022.1 in c:\users\reddappa reddy\anaconda3\lib\site-packages (from pandas) (2023.3)
Requirement already satisfied: numpy>=1.21.0 in c:\users\reddappa reddy\anaconda3\lib\site-packages (from pandas) (1.24.3)
Requirement already satisfied: six>=1.5 in c:\users\reddappa reddy\anaconda3\lib\site-packages (from python-dateutil>=2.8.2->pandas) (1.16.0)
Note: you may need to restart the kernel to use updated packages.

```
In [2]: pip install sklearn
```

Requirement already satisfied: sklearn in c:\users\reddappa reddy\anaconda3\lib\site-packages (0.0.post10)
Note: you may need to restart the kernel to use updated packages.

Step1:Import Libraries

```
In [17]: import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score, classification_report
```

Step 2: Load Data

```
In [18]: data = pd.read_csv("diabetes_prediction(1).csv")
data
```

Out[18]:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
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

Step 3: Define Features and Target Variables

```
In [19]: X = data.drop('Outcome', axis=1)
y = data['Outcome']
y
```

Out[19]:

0 1
1 0
2 1
3 0
4 1
..
763 0
764 0
765 0
766 1
767 0
Name: Outcome, Length: 768, dtype: int64

Step 4: Split Data into Training and Testing Sets

```
In [20]: X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)
```

Step 5: Standardize/Normalize Features

```
In [21]: scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)
```

selecting algorithm

Step 6: # Create a logistic regression model

```
In [22]: model = LogisticRegression()
```

Step 7: Train the Model

```
In [23]: model.fit(X_train, y_train)
```

```
Out[23]: LogisticRegression
LogisticRegression()
```

Step 8: Make Predictions on the Test Data

```
In [24]: y_pred = model.predict(X_test)
```

Step 9: Calculate Accuracy

```
In [26]: accuracy = accuracy_score(y_test, y_pred)
print(f'Accuracy: {accuracy:.2f}')

report = classification_report(y_test, y_pred)
print(report)
```

```
Accuracy: 0.74
```

	precision	recall	f1-score	support
0	0.80	0.79	0.80	151
1	0.62	0.62	0.62	80
accuracy			0.74	231
macro avg	0.71	0.71	0.71	231
weighted avg	0.74	0.74	0.74	231

```
In [ ]:
```

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js