Load the data from CSV file and split it into training and test datasets.

```
import pandas as pd
```

```
df = pd.read_csv("diabetes.csv")
```

```
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
```

```
df.head(10)
```

	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
5	5	116	74	0	0	25.6	0.201	30	0
6	3	78	50	32	88	31.0	0.248	26	1
7	10	115	0	0	0	35.3	0.134	29	0
8	2	197	70	45	543	30.5	0.158	53	1
9	8	125	96	0	0	0.0	0.232	54	1

```
from sklearn.model_selection import train_test_split
```

```
x = df.drop("Outcome", axis = 1)
y = df['Outcome']
```

```
print(y)
```

```
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
```

```
x_train,x_test,y_train,y_test = train_test_split(x,y,test_size=0.30, random_state=1)
```

Summarize the properties in the training dataset so that we can calculate probabilities and make Predictions

x_train

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age
88	15	136	70	32	110	37.1	0.153	43
467	0	97	64	36	100	36.8	0.600	25
550	1	116	70	28	0	27.4	0.204	21
147	2	106	64	35	119	30.5	1.400	34
481	0	123	88	37	0	35.2	0.197	29
...	...	...	...	...	...	...	...	...
645	2	157	74	35	440	39.4	0.134	30
715	7	187	50	33	392	33.9	0.826	34
72	13	126	90	0	0	43.4	0.583	42
235	4	171	72	0	0	43.6	0.479	26
37	9	102	76	37	0	32.9	0.665	46

537 rows × 8 columns

y_train

```
88     1
467    0
550    0
147    0
481    0
      ..
645    0
715    1
72     1
235    1
37     1
Name: Outcome, Length: 537, dtype: int64
```

df.describe()

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
count	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000
mean	3.845052	120.894531	69.105469	20.536458	79.799479	31.992578	0.471876	33.240885	0.348958
std	3.369578	31.972618	19.355807	15.952218	115.244002	7.884160	0.331329	11.760232	0.476951
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.078000	21.000000	0.000000
25%	1.000000	99.000000	62.000000	0.000000	0.000000	27.300000	0.243750	24.000000	0.000000
50%	3.000000	117.000000	72.000000	23.000000	30.500000	32.000000	0.372500	29.000000	0.000000
75%	6.000000	140.250000	80.000000	32.000000	127.250000	36.600000	0.626250	41.000000	1.000000
max	17.000000	199.000000	122.000000	99.000000	846.000000	67.100000	2.420000	81.000000	1.000000

Classify samples from the test dataset and a summarised traning dataset

```
from sklearn.naive_bayes import GaussianNB

# Creating a Gaussian classifier
model = GaussianNB()

model.fit(x_train, y_train)

y_pred = model.predict(x_test)


from sklearn import metrics


print("accuracy: ", metrics.accuracy_score(y_test,y_pred))
```

```
accuracy:  0.7835497835497836
```

Start coding or generate with AI.