In [3]:

```
# SVM Classification
import pandas as pd
import numpy as np

# from sklearn import svm
# from sklearn.svm import SVC

from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import GridSearchCV,RandomizedSearchCV
from sklearn.metrics import classification_report


from sklearn.metrics import accuracy_score, confusion_matrix
from sklearn.model_selection import train_test_split, cross_val_score

import matplotlib.pyplot as plt
```

In [4]:

```
ls
```

```
 Volume in drive C is Windows-SSD
 Volume Serial Number is 08B4-FA6D

 Directory of C:\Users\Agnel Sharon Jerald\Machine learning

22-01-2026  15:50    <DIR>          .
22-01-2026  15:48    <DIR>          ..
22-01-2026  15:48    <DIR>          .ipynb_checkpoints
05-01-2026  17:57             5,492 app.py
20-01-2026  13:06           192,551 Basic SVC Implementation.ipynb
20-01-2026  12:59            21,121 boosting.ipynb
03-12-2025  13:32             3,527 Cars.csv
02-01-2026  14:52           480,038 Churn.csv
04-01-2026  23:12         1,719,610 churn_complete_pipeline.pkl
16-12-2025  17:08         8,127,824 churn_coustomer_prediction.zip
04-01-2026  23:16           260,757 CHURN_CUSTOMER_PREDICTION_WITH_FEATURE_ENGINEERING.ipynb
31-12-2025  16:16           146,557 CHURN_CUSTOMER_WITHOUT_FEATURE_ENGINEERING.ipynb
01-12-2025  09:16            29,822 claimants.csv
23-10-2025  10:52           252,200 Data Visualisations.ipynb
09-12-2025  10:56           135,333 Decision Tree Classifier.ipynb
10-12-2025  10:54         1,256,893 Decision Tree_C5.0_CART.ipynb
22-01-2026  15:50            60,249 GSCV.ipynb
20-01-2026  12:51           132,825 Iris Species decicision Tree.ipynb
20-01-2026  12:53            81,208 KNN_updated.ipynb
20-01-2026  12:39            95,160 Linear_Regression.ipynb
20-01-2026  12:44           146,684 Logistic Regression.ipynb
18-11-2025  09:21            57,420 Machine Learning basics.ipynb
06-12-2025  16:02           603,995 Multi Linear Regression.ipynb
04-01-2026  23:10            28,024 mywt.weights.h5
22-10-2025  17:51            26,046 Numpy.ipynb
03-12-2025  09:43            23,279 pima-indians-diabetes.csv
20-01-2026  12:41            75,878 polynomial Regression.ipynb
23-10-2025  16:20            54,750 Practice libraries.ipynb
17-12-2025  01:04            24,287 Predicting Customer Purchase Behavior in the Travel Industry.ipynb
11-12-2025  20:41            50,316 Random Forest Regression Implementation.ipynb
20-01-2026  13:06            14,584 Support Vector Regression Implementation.ipynb
20-01-2026  13:06         5,005,147 SVM Kernels Implementation.ipynb
16-12-2025  17:09           463,609 Travel.csv
              30 File(s)     19,575,186 bytes
               3 Dir(s)  96,951,734,272 bytes free
```

In [5]:

```
df=pd.read_csv(r"C:\Users\Agnel Sharon Jerald\OneDrive\Desktop\Machine learning\pima-indians-diabetes.csv",name
```

In [6]:

```
df
```

Out[6]:

	preg	plas	pres	skin	test	mass	pedi	age	class
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 [7]:

```
X_train, X_test, y_train, y_test = train_test_split(df.iloc[:,:-1],df.iloc[:,-1], test_size=0.33, random_state=
```

In [8]:

```
model=RandomForestClassifier()
paramters={

    "n_estimators":[10,20,30,40,100],
    "max_features":[2,4,6]
}
```

In [9]:

```
grcv=GridSearchCV(model,paramters,cv=5)
```

In [10]:

```
grcv.fit(X_train,y_train)
```

Out[10]:

```
▸ GridSearchCV
▸ best_estimator_:
  RandomForestClassifier
  ▸ RandomForestClassifier
```

In [11]:

```
grcv.best_score_
```

Out[11]:

```
np.float64(0.7821435370264611)
```

In [12]:

```
grcv.best_params_
```

Out[12]:

```
{'max_features': 4, 'n_estimators': 40}
```

In [13]:

```
final_model=RandomForestClassifier(max_features=4, n_estimators=20)
```

In [14]:

```
final_model.fit(X_train,y_train)
```

Out[14]:

```
▾ RandomForestClassifier
RandomForestClassifier(max_features=4, n_estimators=20)
```

In [15]:

```
print(classification_report(y_train,final_model.predict(X_train)))
```

```
              precision    recall  f1-score   support

           0       1.00      1.00      1.00       332
           1       1.00      0.99      1.00       182

    accuracy                           1.00       514
   macro avg       1.00      1.00      1.00       514
weighted avg       1.00      1.00      1.00       514
```

In [16]:

```
final_model.feature_importances_
```

Out[16]:

```
array([0.06775241, 0.32397841, 0.07706851, 0.04685164, 0.05534427,
       0.17724992, 0.12170478, 0.13005006])
```

In [17]:

```
df.columns[:-1]
```

Out[17]:
```
Index(['preg', 'plas', 'pres', 'skin', 'test', 'mass', 'pedi', 'age'], dtype='object')
```

In [18]:
```
rrcv=RandomizedSearchCV(model,paramters,cv=5)
```

In [19]:
```
rrcv.fit(X_train,y_train)
```

Out[19]:
```
▸ RandomizedSearchCV

▸ best_estimator_:
  RandomForestClassifier

  ▸ RandomForestClassifier
```

In [20]:
```
rrcv.best_score_
```

Out[20]:
```
np.float64(0.7763373310489244)
```

In [21]:
```
rrcv.best_params_
```

Out[21]:
```
{'n_estimators': 40, 'max_features': 6}
```

In [22]:
```
print(classification_report(y_test,final_model.predict(X_test)))
```

```
              precision    recall  f1-score   support

           0       0.81      0.84      0.82       168
           1       0.66      0.62      0.64        86

    accuracy                           0.76       254
   macro avg       0.74      0.73      0.73       254
weighted avg       0.76      0.76      0.76       254
```