```
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
```

```
url = "https://raw.githubusercontent.com/jbrownlee/Datasets/master/pima-indians-diabetes.data.csv"
```

```
columns = ["Pregnncies", "Glucose", "BloodPressure", "SkinThickness", "Insulin", "BMI", "DiabetesPedigreeFunction", "Age", "Outcom
```

```
df = pd.read_csv(url, names = columns)
df
```

	Pregnncies	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

Next steps: Generate code with df View recommended plots New interactive sheet

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

```
x_train, x_test, y_train, y_test = train_test_split(x,y,test_size =0.2, random_state = 42)
```

```
dt_clf = DecisionTreeClassifier(criterion = 'entropy', max_depth = 3, random_state = 42)
dt_clf.fit(x_train, y_train)
```

```
▾ DecisionTreeClassifier
DecisionTreeClassifier(criterion='entropy', max_depth=3, random_state=42)
```

```
rf_clf = RandomForestClassifier(n_estimators = 100, random_state = 42)
rf_clf.fit(x_train, y_train)
```

```
▾ RandomForestClassifier
RandomForestClassifier(random_state=42)
```

```
print("Decision Tree - Training Accuracy: ", dt_clf.score(x_train, y_train))
print("Decision Tree - Testing Accuracy: ", dt_clf.score(x_test, y_test))
```

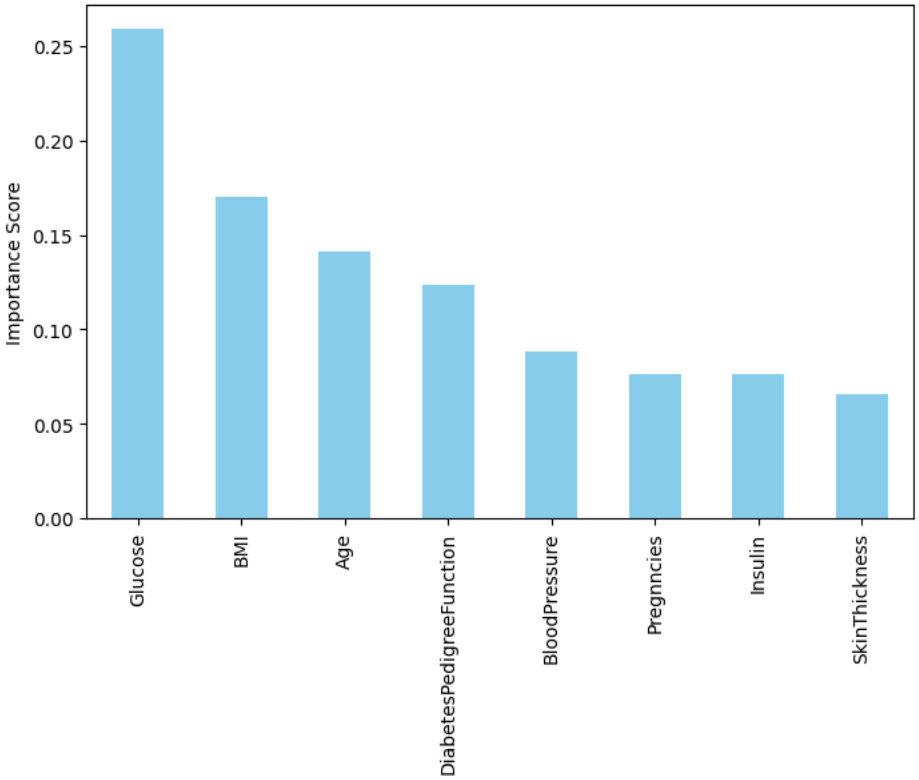
```
Decision Tree - Training Accuracy:  0.7752442996742671
Decision Tree - Testing Accuracy:  0.7662337662337663
```

```
print("Random Forest - Training Accuracy: ", rf_clf.score(x_train, y_train))
print("Decision Tree - Testing Accuracy: ", rf_clf.score(x_test, y_test))
```

```
Random Forest - Training Accuracy:  1.0
Decision Tree - Testing Accuracy:  0.7207792207792207
```

```
importances = rf_clf.feature_importances_
feat_importance = pd.Series(importances, index =x.columns).sort_values(ascending = False)
```

```
plt.figure(figsize=(8,5))
feat_importance.plot(kind='bar', color = "skyblue")
plt.ylabel(" Feature Importance - Random Forest")
plt.ylabel("Importance Score")
plt.show()
```


Importance Score
Glucose
BMI
Age
DiabetesPedigreeFunction
BloodPressure
Pregnncies
Insulin
SkinThickness