

```
In [2]: import pandas as pd
import numpy as np
import matplotlib.pyplot as pt

In [3]: data2 = pd.read_csv('C://users/admin/data2.csv')

In [4]: data2
Out[4]:
```

	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
10	4	110	92	0	0	37.6	0.191	30	0
11	10	168	74	0	0	38.0	0.537	34	1
12	10	139	80	0	0	27.1	1.441	57	0
13	1	189	60	23	846	30.1	0.398	59	1
14	5	166	72	19	175	25.8	0.587	51	1
15	7	100	0	0	0	30.0	0.484	32	1
16	0	118	84	47	230	45.8	0.551	31	1
17	7	107	74	0	0	29.6	0.254	31	1
18	1	103	30	38	83	43.3	0.183	33	0
19	1	115	70	30	96	34.6	0.529	32	1
20	3	126	88	41	235	39.3	0.704	27	0
21	8	99	84	0	0	35.4	0.388	50	0
22	7	196	90	0	0	39.8	0.451	41	1
23	9	119	80	35	0	29.0	0.263	29	1
24	11	143	94	33	146	36.6	0.254	51	1
25	10	125	70	26	115	31.1	0.205	41	1
26	7	147	76	0	0	39.4	0.257	43	1
27	1	97	66	15	140	23.2	0.487	22	0
28	13	145	82	19	110	22.2	0.245	57	0
29	5	117	92	0	0	34.1	0.337	38	0
...
738	2	99	60	17	160	36.6	0.453	21	0
739	1	102	74	0	0	39.5	0.293	42	1
740	11	120	80	37	150	42.3	0.785	48	1
741	3	102	44	20	94	30.8	0.400	26	0
742	1	109	58	18	116	28.5	0.219	22	0
743	9	140	94	0	0	32.7	0.734	45	1
744	13	153	88	37	140	40.6	1.174	39	0
745	12	100	84	33	105	30.0	0.488	46	0
746	1	147	94	41	0	49.3	0.358	27	1
747	1	81	74	41	57	46.3	1.096	32	0
748	3	187	70	22	200	36.4	0.408	36	1
749	6	162	62	0	0	24.3	0.178	50	1
750	4	136	70	0	0	31.2	1.182	22	1
751	1	121	78	39	74	39.0	0.261	28	0
752	3	108	62	24	0	26.0	0.223	25	0
753	0	181	88	44	510	43.3	0.222	26	1
754	8	154	78	32	0	32.4	0.443	45	1
755	1	128	88	39	110	36.5	1.057	37	1
756	7	137	90	41	0	32.0	0.391	39	0
757	0	123	72	0	0	36.3	0.258	52	1
758	1	106	76	0	0	37.5	0.197	26	0
759	6	190	92	0	0	35.5	0.278	66	1
760	2	88	58	26	16	28.4	0.766	22	0
761	9	170	74	31	0	44.0	0.403	43	1
762	9	89	62	0	0	22.5	0.142	33	0
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 x 9 columns

```
In [5]: data2.info()
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 768 entries, 0 to 767
Data columns (total 9 columns):
Pregnancies          768 non-null int64
Glucose              768 non-null int64
BloodPressure        768 non-null int64
SkinThickness        768 non-null int64
Insulin              768 non-null int64
BMI                  768 non-null float64
DiabetesPedigreeFunction 768 non-null float64
Age                  768 non-null int64
Outcome              768 non-null int64
dtypes: float64(2), int64(7)
memory usage: 54.1 KB

In [8]: x=data2[['Glucose','BloodPressure','SkinThickness','Insulin','BMI','Age']]

In [9]: x.shape
Out[9]: (768, 6)

In [10]: y=data2['Outcome']

In [11]: y.shape
Out[11]: (768,)
```

Random Forest

```
In [50]: from sklearn.model_selection import train_test_split

In [51]: x=data2[['Glucose','BloodPressure','SkinThickness','Insulin','BMI','Age']]

In [55]: y=data2['Outcome']

In [56]: x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.25,random_state=50)

In [57]: x_train.shape
Out[57]: (576, 6)

In [58]: y_train.shape
Out[58]: (576,)

In [59]: x_test.shape
Out[59]: (192, 6)

In [23]: y_test.shape
Out[23]: (192, )

In [60]: from sklearn.ensemble import RandomForestClassifier

In [61]: rf = RandomForestClassifier(n_estimators=100, random_state=0)

In [62]: rf.fit(x_train, y_train)
Out[62]: RandomForestClassifier(bootstrap=True, class_weight=None, criterion='gini',
                                max_depth=None, max_features='auto', max_leaf_nodes=None,
                                min_impurity_decrease=0.0, min_impurity_split=None,
                                min_samples_leaf=1, min_samples_split=2,
                                min_weight_fraction_leaf=0.0, n_estimators=100,
                                n_jobs=None, oob_score=False, random_state=0, verbose=0,
                                warm_start=False)

In [63]: print("Accuracy on training set: {:.3f}".format(rf.score(x_train, y_train)))
Accuracy on training set: 1.000

In [64]: print("Accuracy on test set: {:.3f}".format(rf.score(x_test, y_test)))
Accuracy on test set: 0.750

In [71]: rf1 = RandomForestClassifier(max_depth=3, n_estimators=100, random_state=5)

In [72]: rf1.fit(x_train, y_train)
Out[72]: RandomForestClassifier(bootstrap=True, class_weight=None, criterion='gini',
                                max_depth=3, max_features='auto', max_leaf_nodes=None,
                                min_impurity_decrease=0.0, min_impurity_split=None,
                                min_samples_leaf=1, min_samples_split=2,
                                min_weight_fraction_leaf=0.0, n_estimators=100,
                                n_jobs=None, oob_score=False, random_state=5, verbose=0,
                                warm_start=False)

In [73]: print("Accuracy on training set: {:.3f}".format(rf1.score(x_train, y_train)))
Accuracy on training set: 0.814

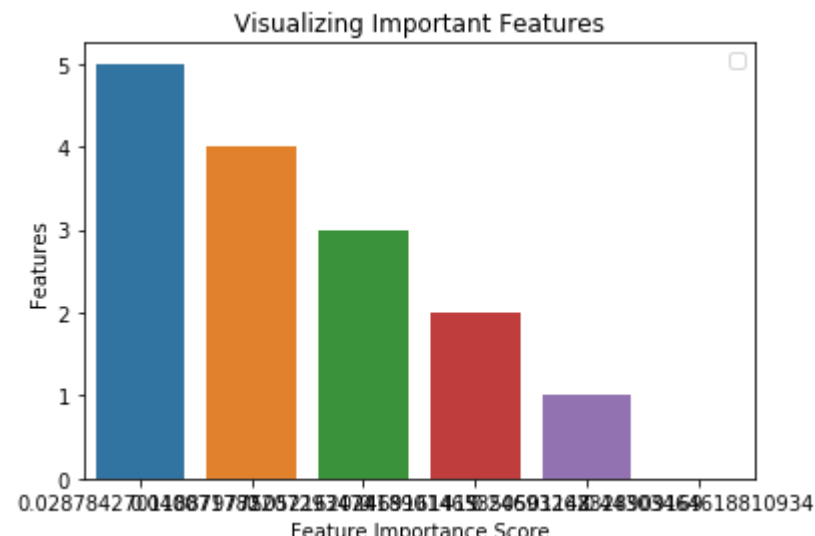
In [74]: print("Accuracy on test set: {:.3f}".format(rf1.score(x_test, y_test)))
Accuracy on test set: 0.734

In [36]: import pandas as pd

In [75]: feature_imp = pd.Series(rf1.feature_importances_).sort_values(ascending=False)

In [76]: feature_imp
Out[76]: 0    0.443035
4    0.246933
5    0.189015
3    0.052162
1    0.040072
2    0.028784
dtype: float64

In [77]: import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
# Creating a bar plot
sns.barplot(x=feature_imp, y=feature_imp.index)
# Add labels to your graph
plt.xlabel('Feature Importance Score')
plt.ylabel('Features')
plt.title("Visualizing Important Features")
plt.legend()
plt.show()
No handles with labels found to put in legend.
```



```
In [ ]:
```