

Random Forest

```
In [1]: import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

```
In [2]: df1=pd.read_csv(r"C:\Users\user\Downloads\C5_health care diabetes.csv")
df1
```

Out[2]:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction
0	6	148	72	35	0	33.6	0.62
1	1	85	66	29	0	26.6	0.35
2	8	183	64	0	0	23.3	0.67
3	1	89	66	23	94	28.1	0.16
4	0	137	40	35	168	43.1	2.28
...
763	10	101	76	48	180	32.9	0.17
764	2	122	70	27	0	36.8	0.34
765	5	121	72	23	112	26.2	0.24
766	1	126	60	0	0	30.1	0.34
767	1	93	70	31	0	30.4	0.31

768 rows × 9 columns



```
In [3]: df=df1.head(10)
df
```

Out[3]:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction
0	6	148	72	35	0	33.6	0.627
1	1	85	66	29	0	26.6	0.351
2	8	183	64	0	0	23.3	0.672
3	1	89	66	23	94	28.1	0.167
4	0	137	40	35	168	43.1	2.288
5	5	116	74	0	0	25.6	0.201
6	3	78	50	32	88	31.0	0.248
7	10	115	0	0	0	35.3	0.134
8	2	197	70	45	543	30.5	0.158
9	8	125	96	0	0	0.0	0.232

```
In [5]: df['Outcome'].value_counts()
```

```
Out[5]: 1    6
        0    4
        Name: Outcome, dtype: int64
```

```
In [7]: x=df.drop('Outcome',axis=1)
        y=df['Outcome']
```

```
In [8]: g1={"g":{"g":1,'g':2}}
df=df.replace(g1)
print(df)
```

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	\
0	6	148	72	35	0	33.6	
1	1	85	66	29	0	26.6	
2	8	183	64	0	0	23.3	
3	1	89	66	23	94	28.1	
4	0	137	40	35	168	43.1	
5	5	116	74	0	0	25.6	
6	3	78	50	32	88	31.0	
7	10	115	0	0	0	35.3	
8	2	197	70	45	543	30.5	
9	8	125	96	0	0	0.0	

	DiabetesPedigreeFunction	Age	Outcome
0	0.627	50	1
1	0.351	31	0
2	0.672	32	1
3	0.167	21	0
4	2.288	33	1
5	0.201	30	0
6	0.248	26	1
7	0.134	29	0
8	0.158	53	1
9	0.232	54	1

```
In [9]: from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test = train_test_split(x,y,train_size=0.70)
```

```
In [10]: from sklearn.ensemble import RandomForestClassifier

rfc = RandomForestClassifier()
rfc.fit(x_train,y_train)
```

Out[10]: RandomForestClassifier()

```
In [11]: parameters = { 'max_depth':[1,2,3,4,5],
                        'min_samples_leaf':[5,10,15,20,25],
                        'n_estimators':[10,20,30,40,50]
                      }
```

```
In [12]: from sklearn.model_selection import GridSearchCV

grid_search = GridSearchCV(estimator=rfc,param_grid=parameters,cv=2,scoring="a
grid_search.fit(x_train,y_train)
```

Out[12]: GridSearchCV(cv=2, estimator=RandomForestClassifier(),
param_grid={'max_depth': [1, 2, 3, 4, 5],
'min_samples_leaf': [5, 10, 15, 20, 25],
'n_estimators': [10, 20, 30, 40, 50]},
scoring='accuracy')

```
In [13]: rf_best=grid_search.best_estimator_  
print(rf_best)
```

```
RandomForestClassifier(max_depth=1, min_samples_leaf=5, n_estimators=10)
```

```
In [14]: from sklearn.tree import plot_tree
```

```
plt.figure(figsize=(80,40))
```

```
plot_tree(rf_best.estimators_[5],feature_names=x.columns,class_names=['Yes','No'])
```

```
Out[14]: [Text(2232.0, 1087.2, 'gini = 0.49\nsamples = 5\nvalue = [4, 3]\nclass = Yes\ns')]
```

gini = 0.49
samples = 5
value = [4, 3]
class = Yes

```
In [ ]:
```