

```
In [2]: import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

```
In [3]: "C:\\Users\\user\\Downloads\\C5_health care diabetes - C5_health care diabetes.csv")
```

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
...
763	10	101	76	48	180	32.9	0.171
764	2	122	70	27	0	36.8	0.340
765	5	121	72	23	112	26.2	0.245
766	1	126	60	0	0	30.1	0.349
767	1	93	70	31	0	30.4	0.315

768 rows × 9 columns

```
In [4]: 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
```

```
In [7]: df['Outcome'].value_counts()
```

```
Out[7]: 0    500
        1    268
        Name: Outcome, dtype: int64
```

```
In [12]: x=df.drop('Outcome',axis=1)
        y=df['Outcome']
```

```
In [14]: g1={"1":{"1":0}}
        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	
..	
763	10	101	76	48	180	32.9	
764	2	122	70	27	0	36.8	
765	5	121	72	23	112	26.2	
766	1	126	60	0	0	30.1	
767	1	93	70	31	0	30.4	

	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
..
763	0.171	63	0
764	0.340	27	0
765	0.245	30	0
766	0.349	47	1
767	0.315	23	0

[768 rows x 9 columns]

```
In [15]: from sklearn.model_selection import train_test_split
        x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=45)
```

```
In [16]: from sklearn.ensemble import RandomForestClassifier
        rfc = RandomForestClassifier()
        rfc.fit(x_train,y_train)
```

```
Out[16]: RandomForestClassifier()
```

```
In [17]: parameters = {'max_depth':[1,2,3,4,5],
                        'min_samples_leaf':[5,10,15,20,25],
                        'n_estimators':[10,20,30,40,50]}
```

```
In [18]: from sklearn.model_selection import GridSearchCV
```

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

```
Out[18]: 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 [19]: grid_search.best_score_
```

```
Out[19]: 0.7690232778806568
```

```
In [20]: rfc_best = grid_search.best_estimator_
```

```
In [21]: # drawing decision tree
from sklearn.tree import plot_tree

plt.figure(figsize=(80,40))
plot_tree(rfc_best.estimators_[5],feature_names=x.columns,class_names=['Yes', 'No']
text(4362.545454545454, 181.19999999999982, gini = 0.0\nsamples = 19\nvalue
= [0, 33]\nclass = No')]
```



