

```
In [1]: #Importing the modules Pandas and Numpy
import pandas as pd
import numpy as np
from sklearn.tree import DecisionTreeClassifier
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
from sklearn.externals import joblib

C:\Users\Admin\Anaconda3\lib\site-packages\sklearn\externals\joblib\__in
it__.py:15: DeprecationWarning: sklearn.externals.joblib is deprecated i
n 0.21 and will be removed in 0.23. Please import this functionality dir
ectly from joblib, which can be installed with: pip install joblib. If t
his warning is raised when loading pickled models, you may need to re-se
rialize those models with scikit-learn 0.21+.
    warnings.warn(msg, category=DeprecationWarning)
```

```
In [2]: #Load the data
diabetes = pd.read_csv('diabetes.csv')
diabetes.head()
```

Out[2]:

| | Pregnancies | Glucose | BloodPressure | SkinThickness | Insulin | Age | Outcome |
|---|-------------|---------|---------------|---------------|---------|-----|---------|
| 0 | 6 | 148 | 72 | 35 | 0 | 45 | 1 |
| 1 | 1 | 85 | 66 | 29 | 0 | 31 | 0 |
| 2 | 8 | 183 | 64 | 0 | 0 | 32 | 1 |
| 3 | 1 | 89 | 66 | 23 | 94 | 21 | 0 |
| 4 | 0 | 137 | 40 | 35 | 168 | 33 | 1 |

```
In [11]: from sklearn.preprocessing import LabelEncoder
labelencoder_diabetes=LabelEncoder()
diabetes['Pregnancies']=labelencoder_diabetes.fit_transform(diabetes['Pr
egnancies'])
diabetes['Glucose']=labelencoder_diabetes.fit_transform(diabetes['Glucos
e'])
diabetes['BloodPressure']=labelencoder_diabetes.fit_transform(diabetes['
BloodPressure'])
diabetes['SkinThickness']=labelencoder_diabetes.fit_transform(diabetes['
SkinThickness'])
diabetes['Insulin']=labelencoder_diabetes.fit_transform(diabetes['Insuli
n'])
diabetes['Age']=labelencoder_diabetes.fit_transform(diabetes['Age'])
diabetes['Outcome']=labelencoder_diabetes.fit_transform(diabetes['Outcom
e'])
diabetes.head()
```

Out[11]:

| | Pregnancies | Glucose | BloodPressure | SkinThickness | Insulin | Age | Outcome |
|---|-------------|---------|---------------|---------------|---------|-----|---------|
| 0 | 6 | 86 | 22 | 28 | 0 | 24 | 1 |
| 1 | 1 | 23 | 19 | 22 | 0 | 10 | 0 |
| 2 | 8 | 121 | 17 | 0 | 0 | 11 | 1 |

| | | | | | | | |
|---|---|----|----|----|-----|----|---|
| 3 | 1 | 27 | 19 | 16 | 62 | 0 | 0 |
| 4 | 0 | 75 | 4 | 28 | 102 | 12 | 1 |

```
In [3]: #Predict
#Outcome will be target column

columns_target=['Outcome']

columns_train=['Pregnancies', 'Glucose', 'BloodPressure', 'SkinThickness', '
Age']
```

```
In [12]: X = diabetes.drop(columns=['Outcome'])
y = diabetes['Outcome']

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2)
```

```
In [13]: model = DecisionTreeClassifier()
model.fit(X_train, y_train)
predictions = model.predict(X_test)

score = accuracy_score(y_test, predictions)
score
```

```
Out[13]: 0.6883116883116883
```

Prediction

```
In [6]: model = DecisionTreeClassifier()
model.fit(X, y)
joblib.dump(model, 'diabetes-recommender.joblib')
```

```
Out[6]: ['diabetes-recommender.joblib']
```

```
In [7]: model = joblib.load('diabetes-recommender.joblib')
```

```
In [ ]: from sklearn import tree
X = diabetes.drop(columns=['Outcome'])
y = diabetes['Outcome']

model = DecisionTreeClassifier()
model.fit(X, y)

tree.export_graphviz(model, out_file='diabetes-recommender.dot',
                      feature_names=['Pregnancies', 'Glucose', 'BloodPress
ure', 'SkinThickness', 'Insulin', 'Age' ],
                      class_names=sorted(y.unique()),
                      label='all',
                      rounded=True,
                      filled=False)
```

```
In [ ]:
```