```
import numpy as np #linear algebra library of Python
import pandas as pd # build on top of numpy for data analysis, data manipulation and d
import matplotlib.pyplot as plt #plotting library of Python
```

Now let's mount Google drive so that we can upload the diabetes.csv file. You can find the code in the 'Code s

```
from google.colab import drive
drive.mount('/content/gdrive')
```


First thing that we do is take a look at the shape of the dataframe (df.shape) and take a look at first 5 lines th

```
df=pd.read_csv('/content/gdrive/My Drive/Colab Notebooks/creditcard.csv') #import file
df.head() #shows first 5 lines including column namesdf.shape # number of rows and col
```

```
df.shape # provides # rows and # columns of the dataframe df - 768 rows and 9 columns
```

```
Go to this URL in a browser: https://accounts.google.com/o/oauth2/auth?clien
```

Let's create numpy arrays, one for the features (X) and one for the label (y)

```
X=df.drop('Class', 1).values #drop 'Outcome' column but you keep the index column
y=df['Class'].values
```

```
from sklearn.model_selection import train_test_split #method to split training and tes
X_train, X_test, y_train, y_test=train_test_split(X, y, test_size=0.2, random_state=42
```

In KNN we need to scale the features but this is not needed when we deal with Random Forrests so we comm

```
#from sklearn.preprocessing import StandardScaler
#model=StandardScaler()
#X_train=model.fit_transform(X_train)
#X_test=model.transform(X_test)
```

Now we are ready to use the Random Forests algorithm - only parameter that we need to select is n_estimato

```
from sklearn.ensemble import RandomForestClassifier
```

```
model=RandomForestClassifier(n_estimators=100, random_state=42)
model.fit(X_train,y_train)
```

```
from sklearn import metrics
from sklearn.metrics import roc_auc_score
from sklearn.metrics import roc_curve

y_pred_proba=model.predict_proba(X_test)[:,1]
fpr, tpr, thresholds=roc_curve(y_test, y_pred_proba)
print(roc_auc_score(y_test, y_pred_proba)) # ROC score

plt.plot([0,1], [0,1], 'k--')
plt.plot(fpr, tpr, label='RF')
plt.xlabel('fpr')
plt.ylabel('tpr')
plt.title('ROC curve')
plt.show()
```