Bank_customerExitPredict

February 20, 2021

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

```
[2]: #Importing Dataset
```

```
[3]: dataset = pd.read_csv('BankCustomers.csv')
```

```
[4]: dataset.shape
```

```
[4]: (10000, 14)
```

```
[5]: X= dataset.iloc[:,3:13]
     y=dataset.iloc[:,13]
```

```
[6]: # convert categorical feature into dummy variables
```

```
[7]: dataset.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10000 entries, 0 to 9999
Data columns (total 14 columns):
 #   Column           Non-Null Count  Dtype
---  ------           --------------  -----
 0   RowNumber        10000 non-null  int64
 1   CustomerId       10000 non-null  int64
 2   Surname          10000 non-null  object
 3   CreditScore      10000 non-null  int64
 4   Geography        10000 non-null  object
 5   Gender           10000 non-null  object
 6   Age              10000 non-null  int64
 7   Tenure           10000 non-null  int64
 8   Balance          10000 non-null  float64
 9   NumOfProducts    10000 non-null  int64
 10  HasCrCard        10000 non-null  int64
 11  IsActiveMember   10000 non-null  int64
 12  EstimatedSalary  10000 non-null  float64
 13  Exited           10000 non-null  int64
```

```
dtypes: float64(2), int64(9), object(3)
memory usage: 1.1+ MB
```

[8]:
```
states = pd.get_dummies(X['Geography'],drop_first=True)
gender = pd.get_dummies(X['Gender'],drop_first=True)
```

[9]:
```
#concatenate the remaining dummies columns
```

[10]:
```
X=X.drop(['Gender','Geography'],axis=1)
```

[11]:
```
X=pd.concat([X,states,gender],axis=1)
```

[12]:
```
# Splitting the dataset into the Training set and Test set
```

[13]:
```
from sklearn.model_selection import train_test_split
```

[14]:
```
X_train,X_test,y_train,y_test = train_test_split(X,y,test_size = 0.
 ↪2,random_state=0)
```

[15]:
```
# Feature Scaling
```

[16]:
```
from sklearn.preprocessing import StandardScaler

sc = StandardScaler()

X_train = sc.fit_transform(X_train)
X_test = sc.transform(X_test)
```

[17]:
```
# Importing the Keras libraries and packages

import keras
from keras.models import Sequential
from keras.layers import Dense
```

[18]:
```
# Initialising the ANN
classifier = Sequential()
```

[19]:
```
# Adding the input layer and the first hidden layer
```

[20]:
```
classifier.
 ↪add(Dense(activation='relu',input_dim=11,units=6,kernel_initializer='uniform'))
```

[21]:
```
# Adding the second hidden layer
```

[22]:
```
classifier.add(Dense(activation='relu',units=6,kernel_initializer='uniform',))
```

[23]:
```
# Adding the output layer
```

[24]:

```
classifier.
 ↪add(Dense(activation='sigmoid',units=1,kernel_initializer='uniform',))
```

[25]:

```
# Compiling the ANN
```

[26]:

```
classifier.compile(optimizer='adam',loss= 'binary_crossentropy',metrics␣
 ↪=['accuracy'])
```

[27]:

```
# Fitting the ANN to the Training set
```

[ ]:

```
classifier.fit(X_train,y_train,batch_size = 10, nb_epoch = 100)
```

[ ]:

```
# Predicting the Test set results
```

[ ]:

```
y_pred = classifier.predict(X_test)
y_pred = (y_pred > 0.5)
```

[ ]:

```
# Making the Confusion Matrix
```

[ ]:

```
from sklearn.metrics import confusion_matrix,accuracy_score
cm=confusion_matrix(y_test, y_pred)
accuracy=accuracy_score(y_test, y_pred)
```

[ ]:

```

```