

```
# import necessary libraries
import pandas as pd
import numpy as np
import pickle
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns
import sklearn
from sklearn.preprocessing import LabelEncoder, OneHotEncoder
from sklearn.linear_model import LogisticRegression
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn.svm import SVC
from sklearn.model_selection import RandomizedSearchCV
import imblearn
from imblearn.over_sampling import SMOTE
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import accuracy_score, classification_report, confusion_matrix, f1_score
data = pd.read_csv(r"/content/Churn_Modelling[1].csv")
data
```

	RowNumber	CustomerId	Surname	CreditScore	Geography	Gender	Age	Tenure	Balar
0	1	15634602	Hargrave	619	France	Female	42	2	0
1	2	15647311	Hill	608	Spain	Female	41	1	83807
2	3	15619304	Onio	502	France	Female	42	8	159660
3	4	15701354	Boni	699	France	Female	39	1	0
4	5	15737888	Mitchell	850	Spain	Female	43	2	125510
...	...	...	...	...	...	...	...	...	...
9995	9996	15606229	Obijiaku	771	France	Male	39	5	0
9996	9997	15569892	Johnstone	516	France	Male	35	10	57369
9997	9998	15584532	Liu	709	France	Female	36	7	0
9998	9999	15682355	Sabbatini	772	Germany	Male	42	3	75075
9999	10000	15628319	Walker	792	France	Female	28	4	130142

10000 rows x 14 columns

```
data.head()
```

	RowNumber	CustomerId	Surname	CreditScore	Geography	Gender	Age	Tenure	Balance
0	1	15634602	Hargrave	619	France	Female	42	2	0.00
1	2	15647311	Hill	608	Spain	Female	41	1	83807.86
2	3	15619304	Onio	502	France	Female	42	8	159660.80
3	4	15701354	Boni	699	France	Female	39	1	0.00
4	5	15737888	Mitchell	850	Spain	Female	43	2	125510.82

```
import warnings
warnings.filterwarnings("ignore")
from sklearn import metrics
from sklearn.metrics import accuracy_score
```

```
# importing .csv files using Pandas
train = pd.read_csv('/content/Churn_Modelling[1].csv')
test = pd.read_csv('/content/Churn_Modelling[1].csv')
```

```
train['Balance'] = train['Balance'].apply(lambda x: 1 if x == 'male' else 2)
```

```
train.drop(columns=['RowNumber', 'CustomerId', 'Surname', 'Gender', 'Age'], inplace=True)
```

```
X = train.drop(["Balance"], axis=1)
y = train.Balance
```

```
X_train, X_test, y_train, y_test = train_test_split(X, y, random_state=42)
```

```
import lazypredict
from lazypredict.Supervised import LazyClassifier
```

```
clf = LazyClassifier(verbose=0,ignore_warnings=True)
models, predictions = clf.fit(X_train, X_test, y_train, y_test)
models
```

100% |██████████| 29/29 [00:07<00:00, 4.14it/s]

	Accuracy	Balanced Accuracy	ROC AUC	F1 Score	Time Taken
Model					
AdaBoostClassifier	1.00	1.00	None	1.00	0.04
BaggingClassifier	1.00	1.00	None	1.00	0.06
BernoulliNB	1.00	1.00	None	1.00	0.04
DecisionTreeClassifier	1.00	1.00	None	1.00	0.04
DummyClassifier	1.00	1.00	None	1.00	0.04
ExtraTreeClassifier	1.00	1.00	None	1.00	0.03
ExtraTreesClassifier	1.00	1.00	None	1.00	0.18
GaussianNB	1.00	1.00	None	1.00	0.03
KNeighborsClassifier	1.00	1.00	None	1.00	0.16
LabelPropagation	1.00	1.00	None	1.00	1.87
LabelSpreading	1.00	1.00	None	1.00	3.64
LinearDiscriminantAnalysis	1.00	1.00	None	1.00	0.08
RandomForestClassifier	1.00	1.00	None	1.00	0.26
RidgeClassifier	1.00	1.00	None	1.00	0.04
RidgeClassifierCV	1.00	1.00	None	1.00	0.08
LGBMClassifier	1.00	1.00	None	1.00	0.05

```
!pip install matplotlib-venn
```

Looking in indexes: <https://pypi.org/simple>, <https://us-python.pkg.dev/colab-wheels/public/simple/>  
Requirement already satisfied: matplotlib-venn in /usr/local/lib/python3.9/dist-packages (0.11.9)  
Requirement already satisfied: scipy in /usr/local/lib/python3.9/dist-packages (from matplotlib-venn) (1.10.1)  
Requirement already satisfied: matplotlib in /usr/local/lib/python3.9/dist-packages (from matplotlib-venn) (3.7.1)  
Requirement already satisfied: numpy in /usr/local/lib/python3.9/dist-packages (from matplotlib-venn) (1.22.4)  
Requirement already satisfied: importlib-resources>=3.2.0 in /usr/local/lib/python3.9/dist-packages (from matplotlib->matplotlib-venn) (5.12.0)  
Requirement already satisfied: fonttools>=4.22.0 in /usr/local/lib/python3.9/dist-packages (from matplotlib->matplotlib-venn) (4.38.0)  
Requirement already satisfied: pyparsing>=2.3.1 in /usr/local/lib/python3.9/dist-packages (from matplotlib->matplotlib-venn) (3.0.9)  
Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.9/dist-packages (from matplotlib->matplotlib-venn) (0.11.0)  
Requirement already satisfied: kiwisolver>=1.0.1 in /usr/local/lib/python3.9/dist-packages (from matplotlib->matplotlib-venn) (1.4.5)  
Requirement already satisfied: contourpy>=1.0.1 in /usr/local/lib/python3.9/dist-packages (from matplotlib->matplotlib-venn) (1.0.7)  
Requirement already satisfied: python-dateutil>=2.7 in /usr/local/lib/python3.9/dist-packages (from matplotlib->matplotlib-venn) (2.8.2)  
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.9/dist-packages (from matplotlib->matplotlib-venn) (23.0)  
Requirement already satisfied: pillow>=6.2.0 in /usr/local/lib/python3.9/dist-packages (from matplotlib->matplotlib-venn) (8.4.0)  
Requirement already satisfied: zipp>=3.1.0 in /usr/local/lib/python3.9/dist-packages (from importlib-resources>=3.2.0->matplotlib-venn) (3.15.0)  
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.9/dist-packages (from python-dateutil>=2.7->matplotlib-venn) (1.16.0)

```
!apt-get -qq install -y libfluidsynth1
```

E: Package 'libfluidsynth1' has no installation candidate

```
# https://pypi.python.org/pypi/libarchive
!apt-get -qq install -y libarchive-dev && pip install -U libarchive
import libarchive
```

Selecting previously unselected package libarchive-dev:amd64.  
(Reading database ... 122349 files and directories currently installed.)  
Preparing to unpack .../libarchive-dev\_3.4.0-2ubuntu1.2\_amd64.deb ...  
Unpacking libarchive-dev:amd64 (3.4.0-2ubuntu1.2) ...  
Setting up libarchive-dev:amd64 (3.4.0-2ubuntu1.2) ...  
Processing triggers for man-db (2.9.1-1) ...  
Looking in indexes: <https://pypi.org/simple>, <https://us-python.pkg.dev/colab-wheels/public/simple/>  
Collecting libarchive

```
Downloading libarchive-0.4.7.tar.gz (23 kB)
Preparing metadata (setup.py) ... done
Collecting nose
  Downloading nose-1.3.7-py3-none-any.whl (154 kB)
    154.7/154.7 KB 13.9 MB/s eta 0:00:00
Building wheels for collected packages: libarchive
  Building wheel for libarchive (setup.py) ... done
  Created wheel for libarchive: filename=libarchive-0.4.7-py3-none-any.whl size=31644 sha256=8aa27359d42622d6ee6a6021fc3298a8e201f
  Stored in directory: /root/.cache/pip/wheels/c9/a5/cc/cb20f1314d4cdec0001fd72baa1efe93e1542a81bdea2fc639
Successfully built libarchive
Installing collected packages: nose, libarchive
Successfully installed libarchive-0.4.7 nose-1.3.7
```

```
pip install lazypredict
```

```
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
Requirement already satisfied: lazypredict in /usr/local/lib/python3.9/dist-packages (0.2.12)
Requirement already satisfied: joblib in /usr/local/lib/python3.9/dist-packages (from lazypredict) (1.1.1)
Requirement already satisfied: lightgbm in /usr/local/lib/python3.9/dist-packages (from lazypredict) (3.3.5)
Requirement already satisfied: pandas in /usr/local/lib/python3.9/dist-packages (from lazypredict) (1.4.4)
Requirement already satisfied: xgboost in /usr/local/lib/python3.9/dist-packages (from lazypredict) (1.7.5)
Requirement already satisfied: scikit-learn in /usr/local/lib/python3.9/dist-packages (from lazypredict) (1.2.2)
Requirement already satisfied: tqdm in /usr/local/lib/python3.9/dist-packages (from lazypredict) (4.65.0)
Requirement already satisfied: click in /usr/local/lib/python3.9/dist-packages (from lazypredict) (8.1.3)
Requirement already satisfied: scipy in /usr/local/lib/python3.9/dist-packages (from lightgbm->lazypredict) (1.10.1)
Requirement already satisfied: numpy in /usr/local/lib/python3.9/dist-packages (from lightgbm->lazypredict) (1.22.4)
Requirement already satisfied: wheel in /usr/local/lib/python3.9/dist-packages (from lightgbm->lazypredict) (0.40.0)
Requirement already satisfied: threadpoolctl>=2.0.0 in /usr/local/lib/python3.9/dist-packages (from scikit-learn->lazypredict) (3.1.0)
Requirement already satisfied: python-dateutil>=2.8.1 in /usr/local/lib/python3.9/dist-packages (from pandas->lazypredict) (2.8.2)
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.9/dist-packages (from pandas->lazypredict) (2022.7.1)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.9/dist-packages (from python-dateutil>=2.8.1->pandas->lazypredict) (1.16.0)
```

```
### importing lazypredict library
import lazypredict
### importing LazyClassifier for classification problem
from lazypredict.Supervised import LazyClassifier
### importing LazyClassifier for classification problem because here we are solving Classification use case.
from lazypredict.Supervised import LazyClassifier
### importing breast Cancer Dataset from sklearn
from sklearn.datasets import load_breast_cancer
### splitting dataset into training and testing part
from sklearn.model_selection import train_test_split
```

```
from tensorflow.keras.models import Model
import numpy as np
import pandas as pd
import tensorflow
import keras
import tensorflow.keras
```

```
veri=pd.read_csv("/content/Churn_Modelling[1].csv")
```

```
data=veri.copy()
```

```
data.head()
```

	RowNumber	CustomerId	Surname	CreditScore	Geography	Gender	Age	Tenure	Balance	NumOfProducts	HasCrCard	IsActiveMember
0	1	15634602	Hargrave	619	France	Female	42	2	0.00	1	1	1
1	2	15647311	Hill	608	Spain	Female	41	1	83807.86	1	0	1
2	3	15619304	Onio	502	France	Female	42	8	159660.80	3	1	0
3	4	15701354	Boni	699	France	Female	39	1	0.00	2	0	0
4	5	15737888	Mitchell	850	Spain	Female	43	2	125510.82	1	1	1

```
len(data.columns)
```

```
14
```

```
data.columns
```

```
Index(['RowNumber', 'CustomerId', 'Surname', 'CreditScore', 'Geography',
      'Gender', 'Age', 'Tenure', 'Balance', 'NumOfProducts', 'HasCrCard',
      'IsActiveMember', 'EstimatedSalary', 'Exited'],
      dtype='object')
```

```
data.isnull()
```

	RowNumber	CustomerId	Surname	CreditScore	Geography	Gender	Age	Tenure	Balance	NumOfProducts	HasCrCard	IsActiveMember
0	False	False	False	False	False	False	False	False	False	False	False	False
1	False	False	False	False	False	False	False	False	False	False	False	False
2	False	False	False	False	False	False	False	False	False	False	False	False
3	False	False	False	False	False	False	False	False	False	False	False	False
4	False	False	False	False	False	False	False	False	False	False	False	False
...	...	...	...	...	...	...	...	...	...	...	...	...
9995	False	False	False	False	False	False	False	False	False	False	False	False
9996	False	False	False	False	False	False	False	False	False	False	False	False
9997	False	False	False	False	False	False	False	False	False	False	False	False
9998	False	False	False	False	False	False	False	False	False	False	False	False
9999	False	False	False	False	False	False	False	False	False	False	False	False

10000 rows × 14 columns

```
data.isnull().sum()
```

```
RowNumber      0
CustomerId      0
Surname         0
CreditScore    0
Geography       0
Gender          0
Age            0
Tenure         0
Balance        0
NumOfProducts  0
HasCrCard      0
IsActiveMember 0
EstimatedSalary 0
Exited         0
dtype: int64
```

```
x=data.iloc[:,3:-1].values
y=data.Exited.values
```

```
x
```

```
array([[619, 'France', 'Female', ..., 1, 1, 101348.88],
       [608, 'Spain', 'Female', ..., 0, 1, 112542.58],
       [502, 'France', 'Female', ..., 1, 0, 113931.57],
       ...,
       [709, 'France', 'Female', ..., 0, 1, 42085.58],
       [772, 'Germany', 'Male', ..., 1, 0, 92888.52],
       [792, 'France', 'Female', ..., 1, 0, 38190.78]], dtype=object)
```

```
y
```

```
array([1, 0, 1, ..., 1, 1, 0])
```

```
from sklearn.preprocessing import LabelEncoder
le= LabelEncoder()
x[:,2]=le.fit_transform(x[:,2])
```

```
x
```

```
array([[619, 'France', 0, ..., 1, 1, 101348.88],
       [608, 'Spain', 0, ..., 0, 1, 112542.58],
       [502, 'France', 0, ..., 1, 0, 113931.57],
       ...,
       [709, 'France', 0, ..., 0, 1, 42085.58],
```

```
[772, 'Germany', 1, ..., 1, 0, 92888.52],  
[792, 'France', 0, ..., 1, 0, 38190.78]], dtype=object)
```

```
from sklearn.compose import ColumnTransformer  
from sklearn.preprocessing import OneHotEncoder  
ct=ColumnTransformer(transformers=[("encoder",OneHotEncoder(),[1])],remainder="passthrough")  
x=np.array(ct.fit_transform(x))
```

x

```
array([[1.0, 0.0, 0.0, ..., 1, 1, 101348.88],  
       [0.0, 0.0, 1.0, ..., 0, 1, 112542.58],  
       [1.0, 0.0, 0.0, ..., 1, 0, 113931.57],  
       ...,  
       [1.0, 0.0, 0.0, ..., 0, 1, 42085.58],  
       [0.0, 1.0, 0.0, ..., 1, 0, 92888.52],  
       [1.0, 0.0, 0.0, ..., 1, 0, 38190.78]], dtype=object)
```

```
from sklearn.model_selection import train_test_split  
xtrain,xtest,ytrain,ytest=train_test_split(x,y,test_size=0.2,random_state=43)
```

```
from sklearn.preprocessing import StandardScaler  
sc=StandardScaler()  
xtrain1=sc.fit_transform(xtrain)  
xtest1=sc.transform(xtest)
```

```
ann=tensorflow.keras.models.Sequential()
```

```
ann.add(tensorflow.keras.layers.Dense(units=6,activation="relu"))  
ann.add(tensorflow.keras.layers.Dense(units=6,activation="relu"))  
ann.add(tensorflow.keras.layers.Dense(units=1,activation="sigmoid"))
```

```
ann.compile(optimizer="adam",loss="binary_crossentropy",metrics=["accuracy"])
```

```
ann.fit(xtrain1,ytrain,epochs=100)
```

```
Epoch 1/100  
250/250 [=====] - 2s 2ms/step - loss: 0.6054 - accuracy: 0.6876  
Epoch 2/100  
250/250 [=====] - 0s 2ms/step - loss: 0.4852 - accuracy: 0.8014  
Epoch 3/100  
250/250 [=====] - 0s 2ms/step - loss: 0.4453 - accuracy: 0.8096  
Epoch 4/100  
250/250 [=====] - 0s 2ms/step - loss: 0.4212 - accuracy: 0.8176  
Epoch 5/100  
250/250 [=====] - 0s 2ms/step - loss: 0.4011 - accuracy: 0.8292  
Epoch 6/100  
250/250 [=====] - 0s 2ms/step - loss: 0.3839 - accuracy: 0.8419  
Epoch 7/100  
250/250 [=====] - 0s 2ms/step - loss: 0.3713 - accuracy: 0.8470  
Epoch 8/100  
250/250 [=====] - 0s 2ms/step - loss: 0.3621 - accuracy: 0.8512  
Epoch 9/100  
250/250 [=====] - 0s 2ms/step - loss: 0.3558 - accuracy: 0.8536  
Epoch 10/100  
250/250 [=====] - 0s 2ms/step - loss: 0.3515 - accuracy: 0.8543  
Epoch 11/100  
250/250 [=====] - 0s 2ms/step - loss: 0.3481 - accuracy: 0.8549  
Epoch 12/100  
250/250 [=====] - 0s 2ms/step - loss: 0.3452 - accuracy: 0.8587  
Epoch 13/100  
250/250 [=====] - 0s 2ms/step - loss: 0.3432 - accuracy: 0.8581  
Epoch 14/100  
250/250 [=====] - 0s 2ms/step - loss: 0.3413 - accuracy: 0.8587  
Epoch 15/100  
250/250 [=====] - 0s 2ms/step - loss: 0.3403 - accuracy: 0.8591  
Epoch 16/100  
250/250 [=====] - 0s 2ms/step - loss: 0.3391 - accuracy: 0.8608  
Epoch 17/100  
250/250 [=====] - 0s 2ms/step - loss: 0.3382 - accuracy: 0.8601  
Epoch 18/100  
250/250 [=====] - 0s 2ms/step - loss: 0.3377 - accuracy: 0.8612  
Epoch 19/100  
250/250 [=====] - 1s 2ms/step - loss: 0.3373 - accuracy: 0.8604  
Epoch 20/100  
250/250 [=====] - 1s 3ms/step - loss: 0.3366 - accuracy: 0.8625  
Epoch 21/100  
250/250 [=====] - 1s 3ms/step - loss: 0.3357 - accuracy: 0.8622  
Epoch 22/100
```

```
250/250 [=====] - 1s 3ms/step - loss: 0.3357 - accuracy: 0.8621
Epoch 23/100
250/250 [=====] - 1s 3ms/step - loss: 0.3352 - accuracy: 0.8636
Epoch 24/100
250/250 [=====] - 0s 2ms/step - loss: 0.3346 - accuracy: 0.8631
Epoch 25/100
250/250 [=====] - 0s 2ms/step - loss: 0.3342 - accuracy: 0.8625
Epoch 26/100
250/250 [=====] - 0s 2ms/step - loss: 0.3341 - accuracy: 0.8618
Epoch 27/100
250/250 [=====] - 0s 2ms/step - loss: 0.3336 - accuracy: 0.8635
Epoch 28/100
250/250 [=====] - 0s 2ms/step - loss: 0.3334 - accuracy: 0.8635
Epoch 29/100
```

```
ypred=ann.predict(xtest1)
ypred=(ypred>0.5)
```

```
63/63 [=====] - 0s 2ms/step
```

```
from sklearn.metrics import accuracy_score
accuracy_score(ytest,ypred)
```

```
0.8545
```

```
from sklearn.linear_model import LogisticRegression
x_train = train.iloc[:,0:4]
y_train = train['HasCrCard']
log_reg= LogisticRegression()
```

```
print(x_train)
print(X_test)
print(y_train)
```

	CreditScore	Geography	Tenure	Balance
0	619	France	2	2
1	608	Spain	1	2
2	502	France	8	2
3	699	France	1	2
4	850	Spain	2	2
...	...	...	...	...
9995	771	France	5	2
9996	516	France	10	2
9997	709	France	7	2
9998	772	Germany	3	2
9999	792	France	4	2

```
[10000 rows x 4 columns]
```

	CreditScore	Geography	Tenure	NumOfProducts	HasCrCard	IsActiveMember	\
6252	596	Germany	3	2	0	0	
4684	623	France	1	2	1	1	
1731	601	Spain	4	2	1	0	
4742	506	Germany	8	2	1	1	
4521	560	Spain	7	1	1	1	
...	...	...	...	...	...	...	
4862	645	Spain	1	1	0	1	
7025	569	Spain	3	3	1	0	
7647	768	France	0	1	1	1	
7161	690	France	6	1	0	0	
73	604	Germany	5	2	1	1	

	EstimatedSalary	Exited
6252	41788.37	0
4684	146379.30	0
1731	58561.31	0
4742	170679.74	0
4521	114669.79	0
...	...	...
4862	17095.49	0
7025	75084.96	1
7647	8316.19	0
7161	81292.33	0
73	58426.81	0

```
[2500 rows x 8 columns]
```

0	1
1	0
2	1
3	0
4	1
..	

```
9995    1
9996    1
9997    0
9998    1
9999    1
Name: HasCrCard, Length: 10000, dtype: int64
```