

```
In [130... #pip3 install xgboost
#conda install -c conda-forge xgboost

import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
```

```
In [131... dataset = pd.read_csv("Churn_Modelling.csv")
dataset
```

Out[131...

	RowNumber	CustomerId	Surname	CreditScore	Geography	Gender	Age	Tenure	Balance	NumOfPr
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	
...	
9995	9996	15606229	Obijaku	771	France	Male	39	5	0.00	
9996	9997	15569892	Johnstone	516	France	Male	35	10	57369.61	
9997	9998	15584532	Liu	709	France	Female	36	7	0.00	
9998	9999	15682355	Sabbatini	772	Germany	Male	42	3	75075.31	
9999	10000	15628319	Walker	792	France	Female	28	4	130142.79	

10000 rows × 14 columns

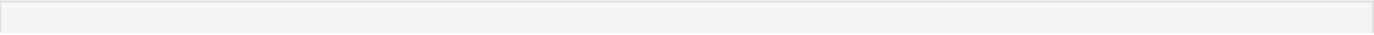


```
In [132... dataset = dataset.drop(["RowNumber", "CustomerId", "Surname", "Geography"],axis=1)
dataset
```

Out[132...

	CreditScore	Gender	Age	Tenure	Balance	NumOfProducts	HasCrCard	IsActiveMember	EstimatedSala
0	619	Female	42	2	0.00	1	1	1	101348.
1	608	Female	41	1	83807.86	1	0	1	112542.
2	502	Female	42	8	159660.80	3	1	0	113931.
3	699	Female	39	1	0.00	2	0	0	93826.
4	850	Female	43	2	125510.82	1	1	1	79084.
...
9995	771	Male	39	5	0.00	2	1	0	96270.
9996	516	Male	35	10	57369.61	1	1	1	101699.
9997	709	Female	36	7	0.00	1	0	1	42085.
9998	772	Male	42	3	75075.31	2	1	0	92888.
9999	792	Female	28	4	130142.79	1	1	0	38190.

10000 rows × 10 columns



```
In [133... from sklearn.preprocessing import LabelEncoder
encoder = LabelEncoder()
encoder.fit(dataset["Gender"].values)
gender_encoded = encoder.transform(dataset["Gender"].values)
dataset["Gender"] = gender_encoded # Female->0 , Male->1
dataset
```

```
Out[133...      CreditScore  Gender  Age  Tenure  Balance  NumOfProducts  HasCrCard  IsActiveMember  EstimatedSala
```

0	619	0	42	2	0.00	1	1	1	101348.
1	608	0	41	1	83807.86	1	0	1	112542.
2	502	0	42	8	159660.80	3	1	0	113931.
3	699	0	39	1	0.00	2	0	0	93826.
4	850	0	43	2	125510.82	1	1	1	79084.
...
9995	771	1	39	5	0.00	2	1	0	96270.
9996	516	1	35	10	57369.61	1	1	1	101699.
9997	709	0	36	7	0.00	1	0	1	42085.
9998	772	1	42	3	75075.31	2	1	0	92888.
9999	792	0	28	4	130142.79	1	1	0	38190.

10000 rows × 10 columns



```
In [134... Y = dataset["Exited"].values # müşterinin dönüşü olumlu (1) veya olumsuz (0)
X = dataset.drop("Exited",axis=1)
```

```
In [135... from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test = train_test_split(X,Y,test_size=0.33)
```

```
In [136... from xgboost import XGBClassifier
model = XGBClassifier() #parametreler ayarlanabilir
model.fit(x_train,y_train)
```

C:\ProgramData\Anaconda3\lib\site-packages\xgboost\sklearn.py:1224: UserWarning: The use of label encoder in XGBClassifier is deprecated and will be removed in a future release. To remove this warning, do the following: 1) Pass option use_label_encoder=False when constructing XGBClassifier object; and 2) Encode your labels (y) as integers starting with 0, i.e. 0, 1, 2, ..., [num_class - 1].

warnings.warn(label_encoder_deprecation_msg, UserWarning)

[21:06:05] WARNING: D:\bld\xgboost-split_1637426510059\work\src\learner.cc:1115: Starting in XGBoost 1.3.0, the default evaluation metric used with the objective 'binary:logistic' was changed from 'error' to 'logloss'. Explicitly set eval_metric if you'd like to restore the old behavior.

```
Out[136... XGBClassifier(base_score=0.5, booster='gbtree', colsample_bylevel=1,
               colsample_bynode=1, colsample_bytree=1, enable_categorical=False,
               gamma=0, gpu_id=-1, importance_type=None,
               interaction_constraints='', learning_rate=0.300000012,
               max_delta_step=0, max_depth=6, min_child_weight=1, missing=nan,
               monotone_constraints='()', n_estimators=100, n_jobs=8,
               num_parallel_tree=1, predictor='auto', random_state=0,
               reg_alpha=0, reg_lambda=1, scale_pos_weight=1, subsample=1,
               tree_method='exact', validate_parameters=1, verbosity=None)
```

```
In [137... y_pred = model.predict(x_test)
y_pred
```

```
Out[137... array([1, 1, 1, ..., 0, 0, 1], dtype=int64)
```

In [138... `y_test`

Out[138... `array([1, 0, 0, ..., 0, 0, 1], dtype=int64)`

In [139... `from sklearn.metrics import confusion_matrix`
`result = confusion_matrix(y_pred,y_test)`
`result`

Out[139... `array([[2474, 362],`
 `[161, 303]], dtype=int64)`

In [140... *# en iyi performans için hiperparametrelerle tuning edilmesi gereklidir*