Machine Learning Assignment

Selahaddin HONİ
040160046

Mahmut Emin ERTÜRK
040160212

Random Signals & Noise
Spring 2021

DATASET & PROBLEM

TURKNET CHURN Dataset
125 Features (192,000 rows)
Binary Classification

FEATURE EXTRACTION

Selecting Best Features
Hyperparameter: #Features

DATA PREPROCESSING

Min-Max Scaling Normalization
Train Set Class Balance

CLASSIFICATION MODELS

Random Forest Classifier
Support Vector Machine
Multi Layer Perceptron

İTÜ

FEATURE EXTRACTION

Separating the dataset for two target classes: ‘ACIK’ and ‘KAPALI’

```
rows_durum_K = []

for row in range(dataset.shape[0]):
    if dataset["DURUM"][row] == 'K':
        rows_durum_K.append(row)

dataset_K = dataset.iloc[rows_durum_K]
K = dataset_K.describe().transpose()

dataset_A = dataset.drop(rows_durum_K)
A = dataset_A.describe().transpose()
```

Describe
First-order statistics of these classes

A

	count	mean	std	min	25%	50%	75%	max
KALDIGI_AY_SAYISI	180341.0	0.249360	0.248144	0.0	0.094340	0.132075	0.339623	1.000000
RISKLIMUSTERI	180341.0	0.005900	0.076584	0.0	0.000000	0.000000	0.000000	1.000000
KAPASITE	180341.0	0.029591	0.028455	0.0	0.019309	0.019309	0.019309	1.000000
currentDown	178605.0	0.014549	0.009399	0.0	0.007153	0.015646	0.015648	1.000000
ARKADASINIGETIR	180341.0	0.000156	0.002620	0.0	0.000000	0.000000	0.000207	1.000000
...	...	...	...	...	...	...	...	...
PORTERROR_SAYISI_3	180341.0	0.003991	0.012434	0.0	0.000637	0.001591	0.003819	1.000000
MAX_SESSIONTIME_3	180341.0	0.002022	0.031984	0.0	0.001052	0.001084	0.001089	1.000000
MIN_SESSIONTIME_3	180341.0	0.016706	0.048209	0.0	0.000000	0.000000	0.008765	0.993471
TOTALUPLOADGB_3	180341.0	0.005309	0.010864	0.0	0.001460	0.003208	0.006056	1.000000
TOTALDOWNLOADGB_3	180341.0	0.029435	0.033002	0.0	0.009022	0.020480	0.039049	1.000000

111 rows × 8 columns

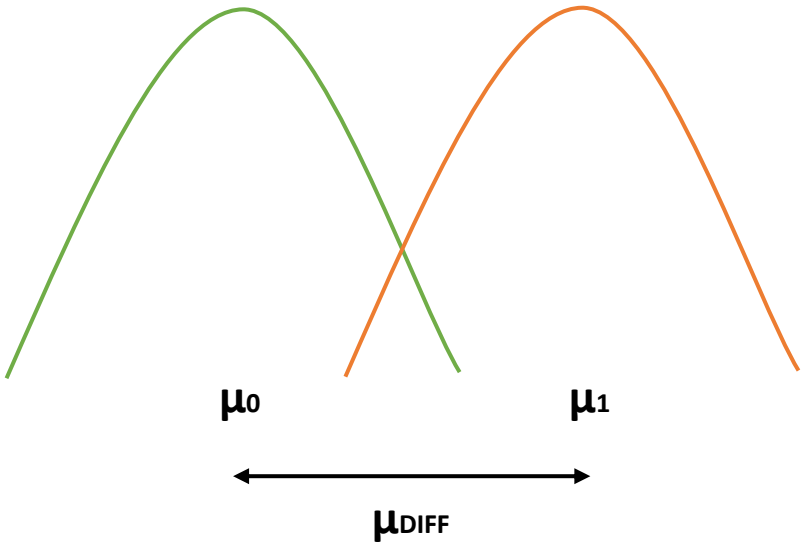
K

	count	mean	std	min	25%	50%	75%	max
KALDIGI_AY_SAYISI	11953.0	0.292397	0.213871	0.037736	0.113208	0.226415	0.415094	1.000000
RISKLIMUSTERI	11953.0	0.038651	0.192771	0.000000	0.000000	0.000000	0.000000	1.000000
KAPASITE	11953.0	0.031754	0.030740	0.000000	0.019309	0.019309	0.059959	0.085366
currentDown	11828.0	0.010086	0.009182	0.000000	0.001797	0.007153	0.015647	0.052392
ARKADASINIGETIR	11953.0	0.000172	0.000450	0.000000	0.000000	0.000000	0.000207	0.012648
...	...	...	...	...	...	...	...	...
PORTERROR_SAYISI_3	11953.0	0.006624	0.022126	0.000000	0.000318	0.001910	0.005729	0.804583
MAX_SESSIONTIME_3	11953.0	0.001264	0.020405	0.000000	0.000993	0.001081	0.001088	0.999965
MIN_SESSIONTIME_3	11953.0	0.016789	0.060534	0.000000	0.000000	0.000000	0.003354	1.000000
TOTALUPLOADGB_3	11953.0	0.006525	0.015491	0.000000	0.000587	0.003484	0.007755	0.952430
TOTALDOWNLOADGB_3	11953.0	0.035843	0.050390	0.000000	0.003352	0.020203	0.047469	0.775082

111 rows × 8 columns

Obtaining the difference & convert the 'mean' variable to absolute values

```
DIFF = A - K

# Convert the absolute values
for row in range(DIFF.shape[0]):
    val = DIFF["mean"][row]
    if val:
        DIFF["mean"][row] = abs(val)
```



DIFF

	count	mean	std	min	25%	50%	75%	max
SAYISI	168388.0	0.043037	0.034273	-0.037736	-0.018868	-0.094340	-0.075472	0.000000
RISKLIMUSTERI	168388.0	0.032751	-0.116186	0.000000	0.000000	0.000000	0.000000	0.000000
KAPASITE	168388.0	0.002163	-0.002285	0.000000	0.000000	0.000000	-0.040650	0.914634
currentDown	166777.0	0.004463	0.000216	0.000000	0.005356	0.008492	0.000001	0.947608
ARKADASINIGETIR	168388.0	0.000016	0.002170	0.000000	0.000000	0.000000	0.000000	0.987352
...	...	...	...	...	...	...	...	...
PORTERROR_SAYISI_3	168388.0	0.002634	-0.009692	0.000000	0.000318	-0.000318	-0.001910	0.195417
MAX_SESSIONTIME_3	168388.0	0.000758	0.011579	0.000000	0.000059	0.000003	0.000001	0.000035
MIN_SESSIONTIME_3	168388.0	0.000082	-0.012324	0.000000	0.000000	0.000000	0.005411	-0.006529
TOTALUPLOADGB_3	168388.0	0.001216	-0.004627	0.000000	0.000873	-0.000277	-0.001700	0.047570
TOTALDOWNLOADGB_3	168388.0	0.006408	-0.017388	0.000000	0.005670	0.000277	-0.008420	0.224918

111 rows × 8 columns

FEATURE EXTRACTION

The best features having more power to classify two targets

```
mean_sorted = DIFF.sort_values(['mean'],ascending=False)['mean']
best_features = mean_sorted[0:10].index
dataset = dataset.dropna(subset=best_features)
```

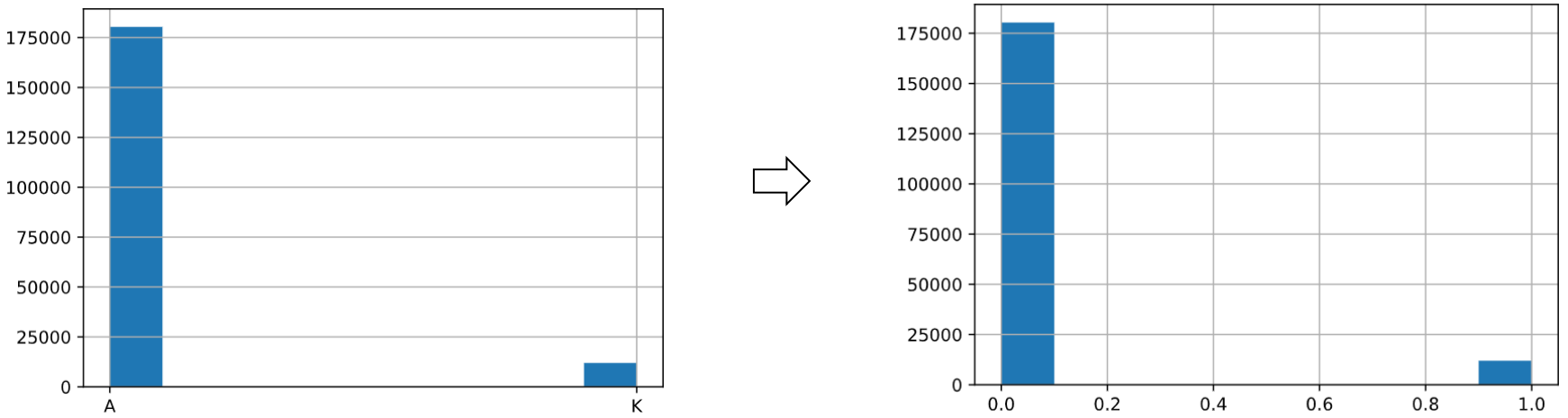
```
Index(['TDU_TICKETSL_1', 'ADSLARIZA_TICKETSL_1', 'KALDIGI_AY_SAYISI',
       'RISKLIMUSTERI', 'IKNATICKET_1', 'TDU_TICKETKAPANMASURESI_1',
       'ADSLARIZA_TICKETSL_2', 'ADSLARIZA_TICKETKAPANMASURESI_1',
       'TDU_TICKETSL_2', 'TDU_DESTEKSAYISI_1'],
      dtype='object')
```

Min-Max Scaling Normalization

```
dataset = dataset.select_dtypes(include=np.number)

for column in dataset.columns:
    dataset[column] = (dataset[column] - dataset[column].min()) / (dataset[column].max() - dataset[column].min())
```

Label Encoding for Target Variable

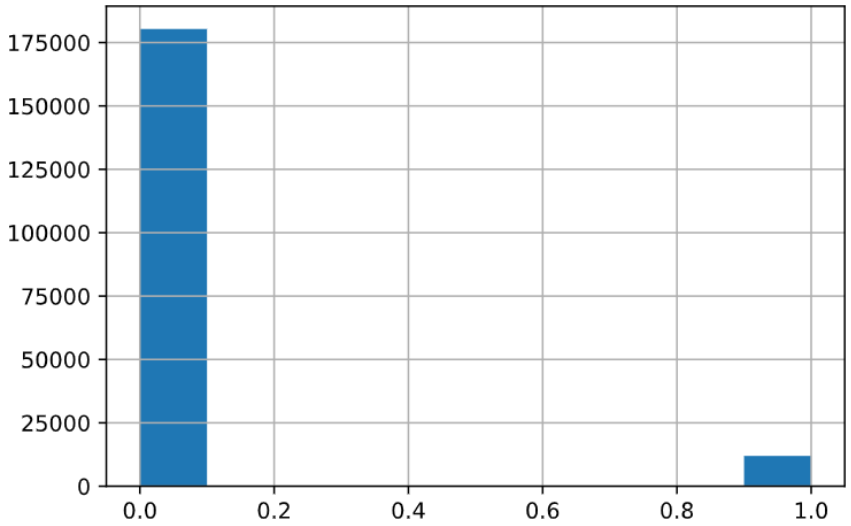
```
from sklearn.preprocessing import LabelEncoder
labelencoder = LabelEncoder()
dataset["DURUM"] = labelencoder.fit_transform(dataset["DURUM"])
```



DATA PREPROCESSING

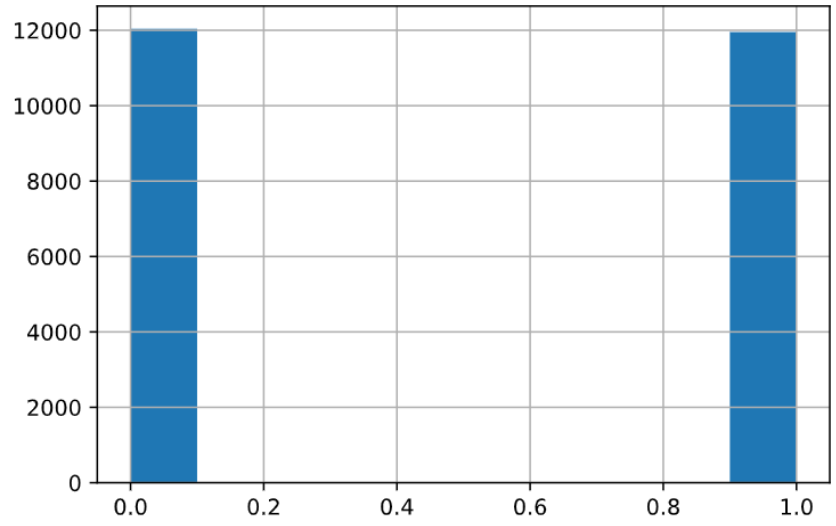
Class Balance

```
drop_rows = []
for row in range(dataset.shape[0]):
    if dataset["DURUM"][row] == 0:
        if not row%15 == 0:
            drop_rows.append(row)

Qset = dataset.drop(drop_rows)
```



Train Test Split

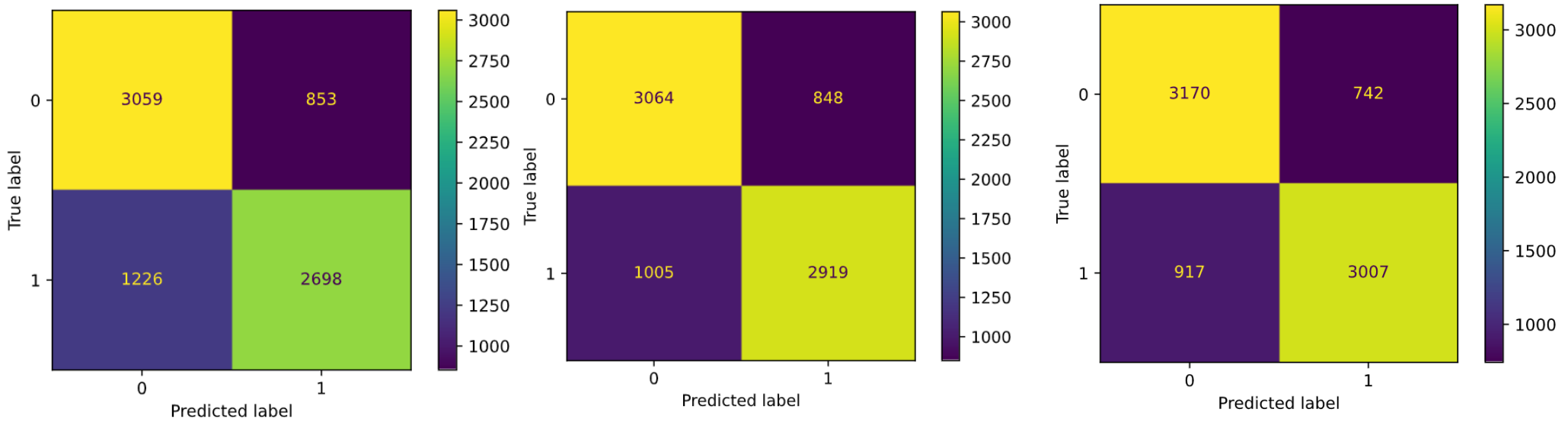
```
from sklearn.model_selection import train_test_split

X = Qset[best_features]
y = Qset["DURUM"]

X_train, X_val, y_train, y_val = train_test_split(X, y, test_size=0.33, random_state=1)
```



Hyper Parameter: Number of Features

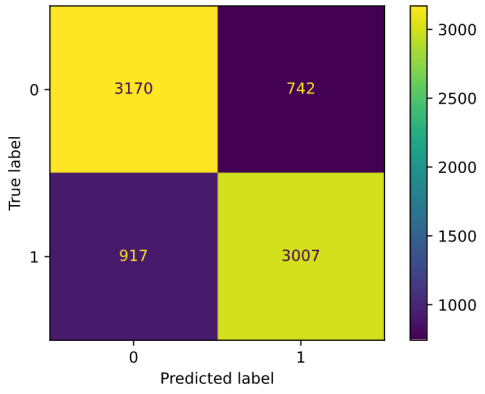
```
from sklearn.ensemble import RandomForestClassifier

clf=RandomForestClassifier(n_estimators=100)

clf.fit(X_train, y_train)

y_pred = clf.predict(X_val)
```

Random Forest Classifier | Confusion Maps for using 10, 25 and 50 features



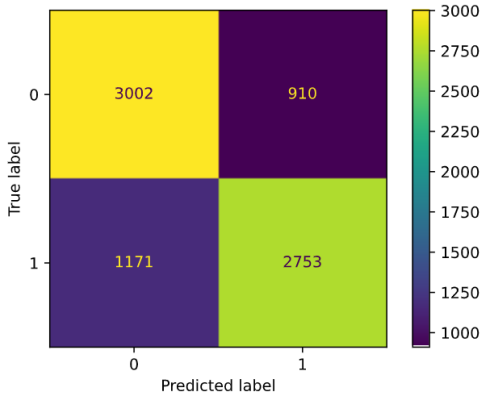
CLASSIFICATION MODELS

Random Forest Classifier



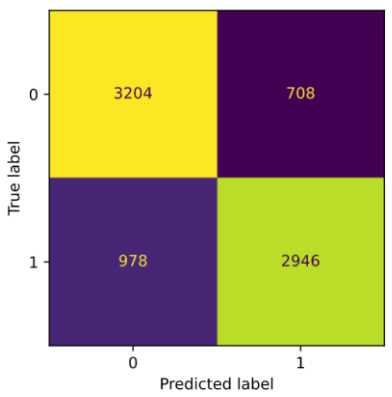
	precision	recall	f1-score	support
ACIK	0.78	0.81	0.79	3912
KAPALI	0.80	0.77	0.78	3924
accuracy			0.79	7836
macro avg	0.79	0.79	0.79	7836
weighted avg	0.79	0.79	0.79	7836

SVM (RBF)



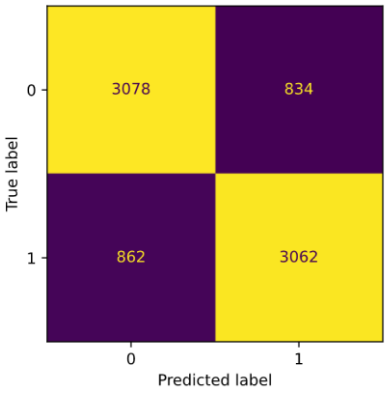
	precision	recall	f1-score	support
ACIK	0.72	0.77	0.74	3912
KAPALI	0.75	0.70	0.73	3924
accuracy			0.73	7836
macro avg	0.74	0.73	0.73	7836
weighted avg	0.74	0.73	0.73	7836

MLP-32



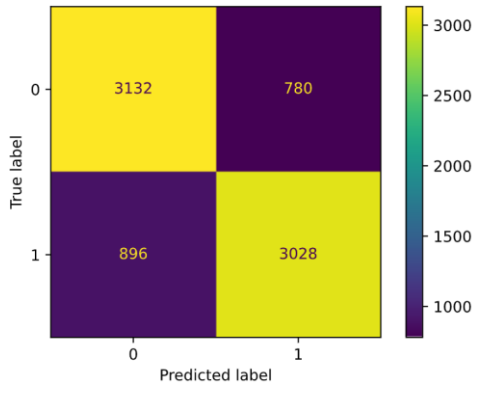
	precision	recall	f1-score	support
ACIK	0.77	0.82	0.79	3912
KAPALI	0.81	0.75	0.78	3924
accuracy			0.78	7836
macro avg	0.79	0.78	0.78	7836
weighted avg	0.79	0.78	0.78	7836

MLP-250



	precision	recall	f1-score	support
ACIK	0.78	0.79	0.78	3912
KAPALI	0.79	0.78	0.78	3924
accuracy			0.78	7836
macro avg	0.78	0.78	0.78	7836
weighted avg	0.78	0.78	0.78	7836

MLP-500



	precision	recall	f1-score	support
ACIK	0.78	0.80	0.79	3912
KAPALI	0.80	0.77	0.78	3924
accuracy			0.79	7836
macro avg	0.79	0.79	0.79	7836
weighted avg	0.79	0.79	0.79	7836

CONCLUSION

They are very similar results; however, for the 'KAPALI' class:

best precision MLP-32
best recall MLP-250

Precision = TP/TP+FP

Recall = TP/TP+FN

POSSIBLE IMPROVEMENT

Adding deeper layers to neural network model may increase the accuracy.

THANK YOU

İTÜ