

National Technical University of Ukraine
“Igor Sikorsky Kyiv Polytechnic Institute”
Faculty of Informatics and Computer Science
Department of Information Systems and Technologies

Lab work No7

Classification in scikit-learn

Performed by:
student of group IM-14
Full name KULUBEĆIOGLU Mehmet

Compiled by:
Yulia Timofeeva

Tasks

Create a Python program that trains and tests the model which does binary classification according to your number in the group; evaluate the model with according metrics and try to create a better model.

Make a report in doc-file. Report must contain:

- Title;
- Program code;
- Results (screenshots) of code execution.

Demonstrate the working program and answer questions about theoretical information and program working.

3: employee.csv. Predict by using classification whether the employee will leave work (last column), using any independent variables.

My All codes:

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score, classification_report, confusion_matrix
```

```
# Veri kümesini yükle
df = pd.read_csv('employee.csv')
```

```
# Kategorik sütunları sayısalı dönüştür
df = pd.get_dummies(df, columns=['department', 'salary'], drop_first=True)
```

```
# Bağımsız değişkenleri ve hedef değişkeni seç
X = df[['satisfaction_level', 'last_evaluation', 'number_project', 'average_monthly_hours', 'time_spent_company', 'Work_accident',
        'promotion_last_5years', 'department_RandD', 'department_accounting', 'department_hr', 'department_management', 'department_marketing',
        'department_product_mng', 'department_sales', 'department_support', 'department_technical', 'salary_low', 'salary_medium']]
y = df['left']
```

```
# Veriyi eğitim ve test setlerine böl
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

```
# Lojistik Regresyon modelini başlat ve max_iter parametresini artırarak modeli eğit
model = LogisticRegression(max_iter=1000) # veya daha büyük bir değer belirleyebilirsiniz
```

```
# Modeli eğit
model.fit(X_train, y_train)
```

```
# Test setinde tahminler yap
y_pred = model.predict(X_test)
```

```
# Modeli değerlendir
accuracy = accuracy_score(y_test, y_pred)
conf_matrix = confusion_matrix(y_test, y_pred)
classification_rep = classification_report(y_test, y_pred)
```

```
# Değerlendirme metriklerini yazdır
print(f'Accuracy: {accuracy}')
print('Confusion Matrix:')
print(conf_matrix)
print('Classification Report:')
print(classification_rep)
```

Accuracy: 0.7833333333333333

Confusion Matrix:

```
[[2119  175]
 [ 475  231]]
```

Classification Report:

	precision	recall	f1-score	support
0	0.82	0.92	0.87	2294
1	0.57	0.33	0.42	706
accuracy			0.78	3000
macro avg	0.69	0.63	0.64	3000
weighted avg	0.76	0.78	0.76	3000