به نام خدا

درس مبانی علم داده

پروژه امتیازی

مبیناسمعیل پور

99441029

زمستان ۱۴۰۲

بخش اول (خواندن دیتاست):

خواندن فایل smoke.csv با کتابخانه pandas

```
data = pd.read_csv('smoke.csv')
```

بخش سوم (engineering Feature):

```
data.drop('Unnamed: 0',axis=1,inplace=True)
```

```
data.drop(['UTC', 'CNT'],axis=1,inplace=True)
```

ستونی به نام 'Unnamed: 0' را از DataFrame حذف کرده.

در حال حذف ستون‌هایی با نام «UTC» و «CNT» از DataFrame است.

پس از اجرای این دو خط کد، ستون های مشخص شده از DataFrame «داده» حذف خواهند شد.

بخش چهارم (مدل سازی و پیش بینی) :

```
X = data.drop('Fire Alarm', axis=1)
y=data['Fire Alarm']
```

جداسازی ویژگی های (X) و متغیر هدف (y): ستون "Fire Alarm " از DataFrame "نمونه ها" حذف می شود و به متغیر "X" اختصاص می یابد. خود ستون 'Fire Alarm ' به متغیر 'y' اختصاص داده می شود.

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=1)
print(len(X_train))
```

```
50104
```

تقسیم داده ها به مجموعه های آموزشی و آزمایشی: تابع 'train_test_split' از ماژول 'sklearn.model_selection' برای تقسیم داده های 'X' و 'y' به مجموعه های آموزشی و آزمایشی استفاده می شود. مجموعه تست 20 درصد کل داده ها است و حالت تصادفی 1 برای تکرارپذیری استفاده می شود.

```
model = MLPClassifier(hidden_layer_sizes=(1500, 1000,500,500), random_state=85, activation = 'relu')
model.fit(X_train, y_train)

accuracy = model.score(X_test, y_test)
print('Accuracy:', accuracy)
```

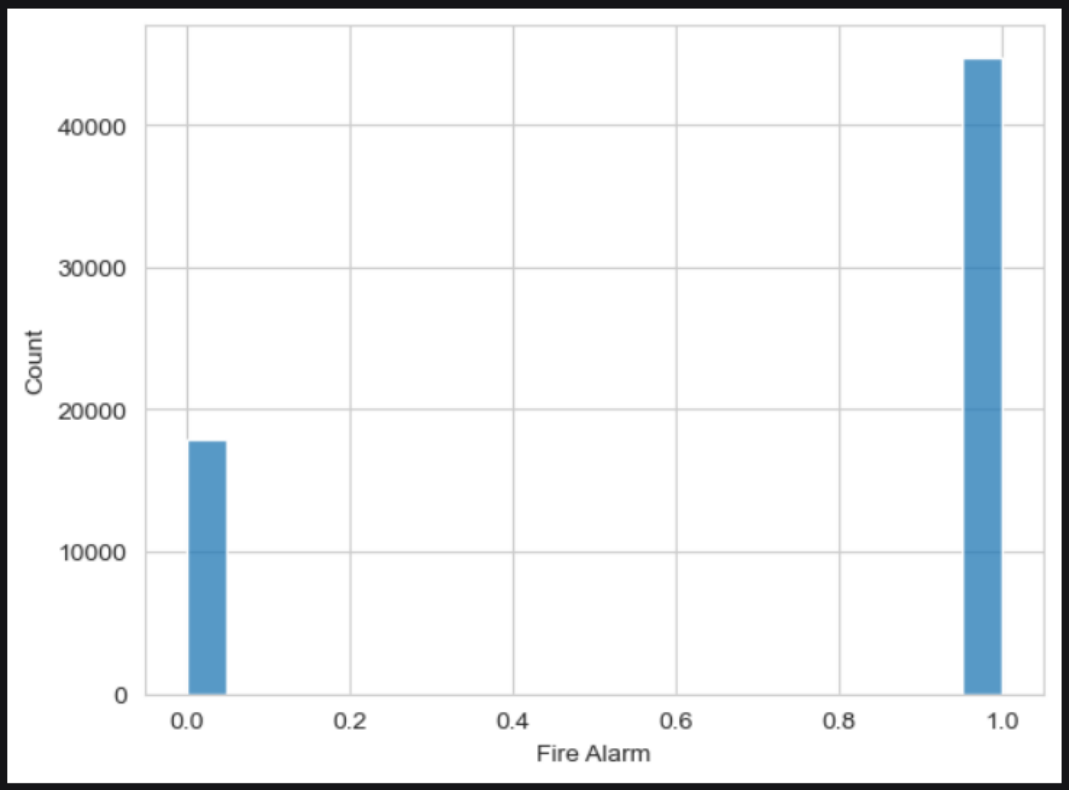
```
Accuracy: 0.9873862366278141
```

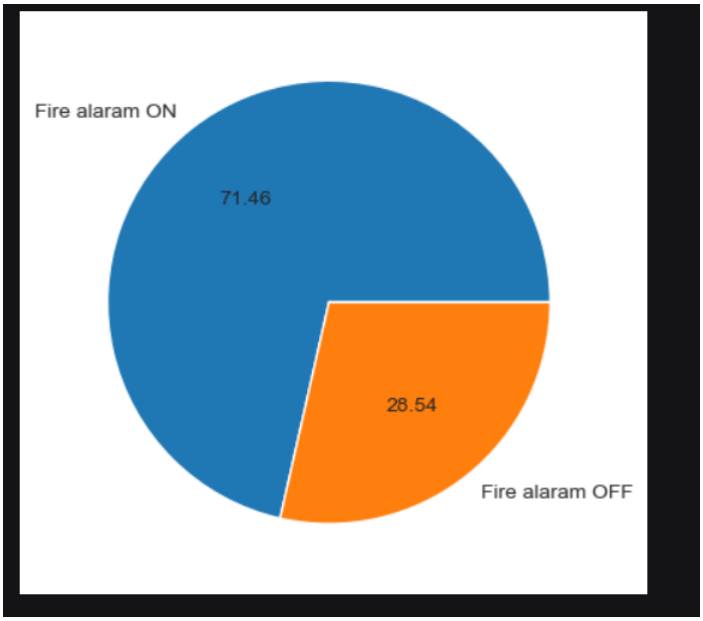
ایجاد و آموزش یک مدل MLPClassifier: یک مدل MLPClassifier با دو لایه مخفی که هر یک از 1500 نورون تشکیل شده است، با استفاده از کلاس 'MLPClassifier' از ماژول 'sklearn.neural_network' ایجاد می شود. مدل از تابع فعال‌سازی relu و حالت تصادفی 85 استفاده می‌کند. سپس مدل بر روی داده‌های آموزشی با استفاده از روش "fit" آموزش داده می‌شود.

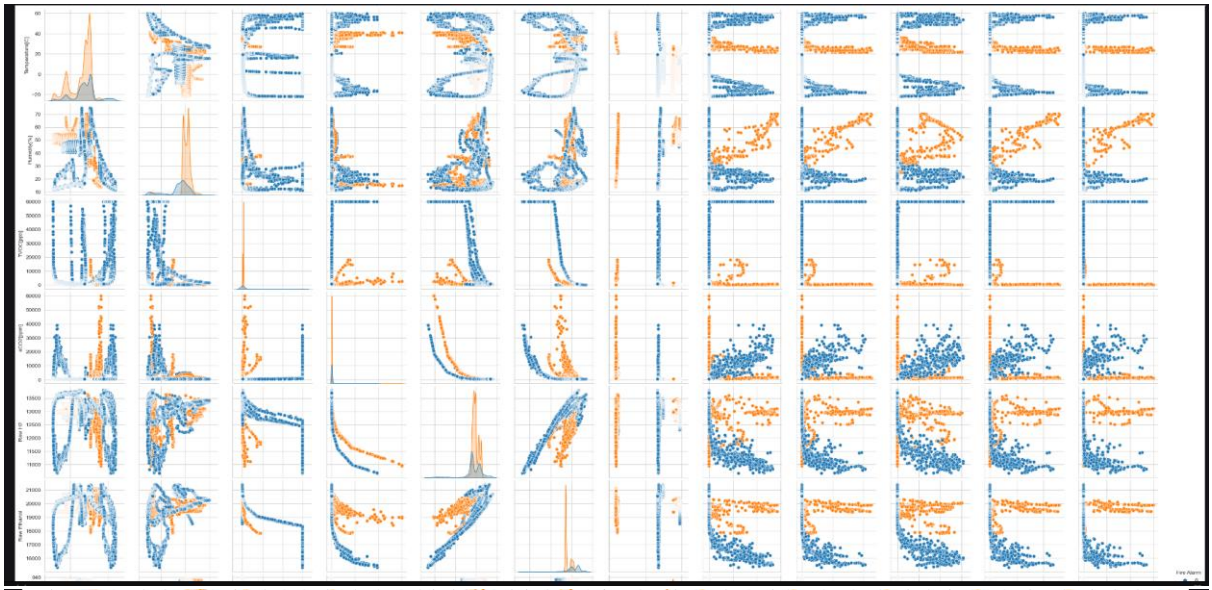
ارزیابی مدل: دقت مدل با فراخوانی روش 'score' بر روی مدل آموزش دیده، با عبور داده های تست محاسبه می شود. سپس دقت در کنسول چاپ می شود.

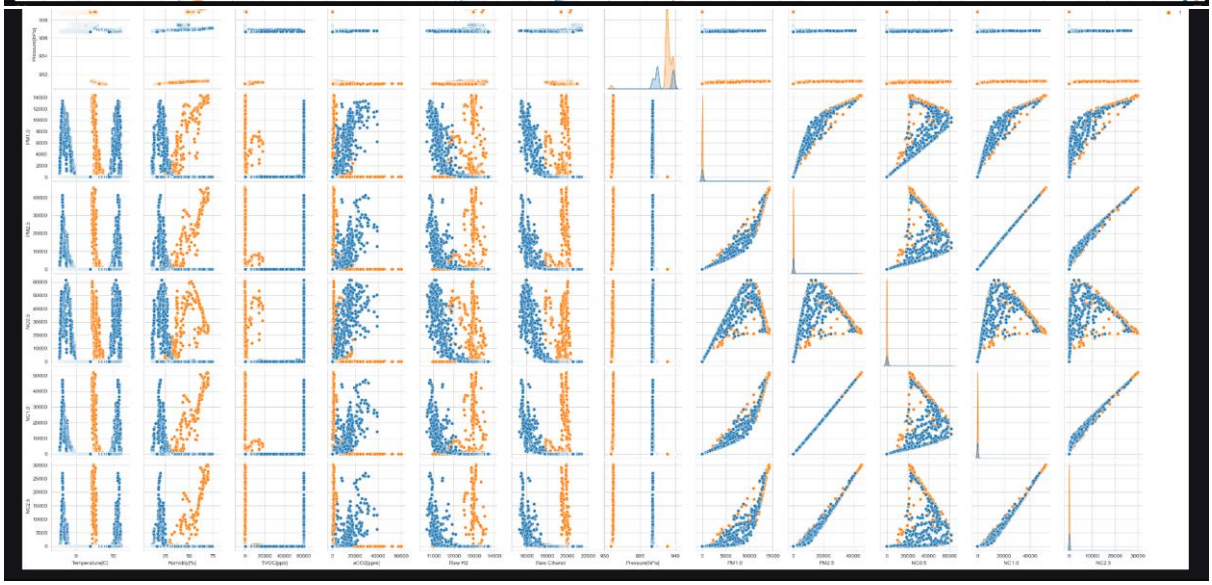
یک شبکه عصبی (MLPRegressor) را برای تقریب یک تابع خطی با نویز اضافه ایجاد می کرده و منطبق می کند. سپس تابع خطی واقعی، تابع آموخته شده از شبکه عصبی و داده های آموزشی را با نویز ترسیم می کند. در نهایت، میانگین مربعات خطا (MSE) بین تابع واقعی و مقادیر پیش بینی شده را محاسبه و چاپ می کند.

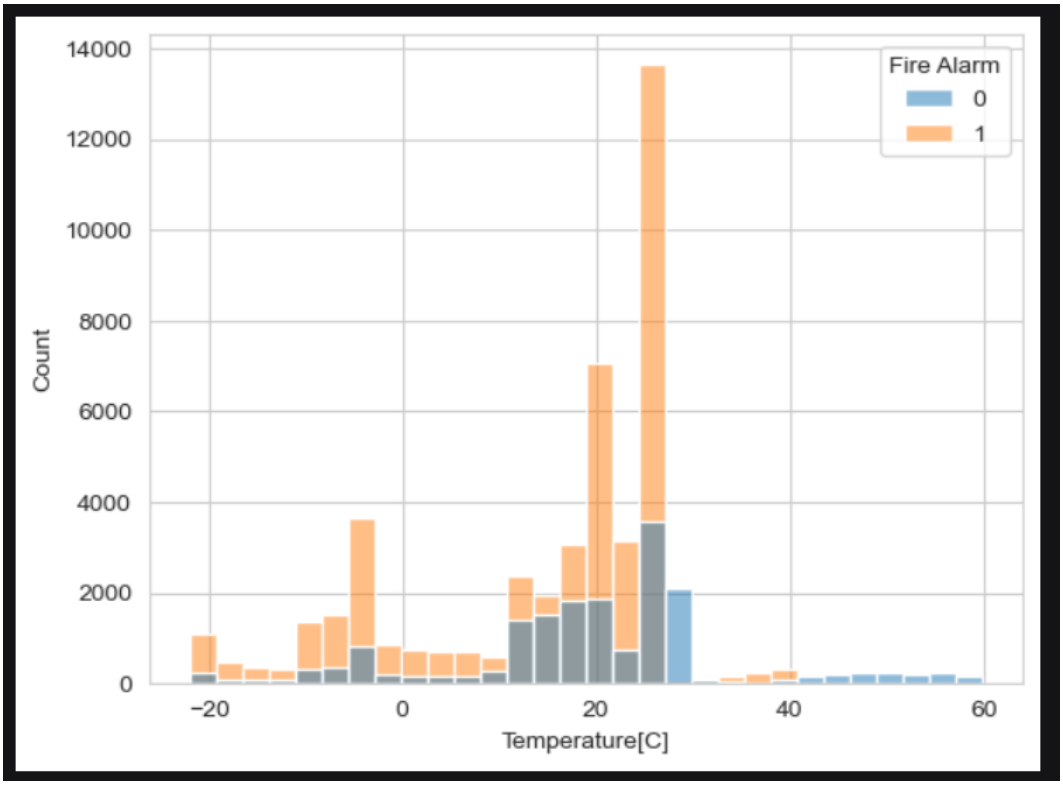
```
<Axes: xlabel='Fire Alarm', ylabel='Count'>
```




Fire alaram ON
71.46
28.54
Fire alaram OFF







مرحله پنجم(گزارش دقت وEvaluation):

یک مدل رگرسیون لجستیک را آموزش می دهد، عملکرد آن را با استفاده از معیارهای مختلف ارزیابی می کند، و معیارهای ارزیابی را از طریق نمودارها به تصویر می کشد.

```
from sklearn.linear_model import LogisticRegression
reg = LogisticRegression()
reg.fit(X_train, y_train)
```

```
c:\Users\esmae\anaconda3\Lib\site-packages\sklearn\linear_model\_logistic.py:460: C
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression
  n_iter_i = _check_optimize_result(
```

```
▾ LogisticRegression
LogisticRegression()
```

یک مدل رگرسیون لجستیک را با استفاده از مجموعه آموزشی آموزش می دهد. کلاس LogisticRegression() برای ایجاد مدل و متد fit() برای آموزش مدل بر روی داده های آموزشی استفاده می شود.

```
from sklearn.metrics import confusion_matrix, classification_report,auc,recall_score,precision_score,roc_curve,accuracy_score,precision_recall_curve
print("Confusion Matrix:\n",confusion_matrix(y_test,prediction))
print()
print("Classification Report:\n",classification_report(y_test,prediction))
```

```
Confusion Matrix:
 [[2251 1395]
 [ 113 8767]]

Classification Report:
              precision    recall  f1-score   support

           0       0.95      0.62      0.75      3646
           1       0.86      0.99      0.92      8880

    accuracy                           0.88     12526
   macro avg       0.91      0.80      0.83     12526
weighted avg       0.89      0.88      0.87     12526
```

این بخش عملکرد مدل را با استفاده از معیارهای مختلف ارزیابی می کند، از جمله:

confusion_matrix: جدولی که برچسب های واقعی و پیش بینی شده برای هر کلاس را خلاصه می کند

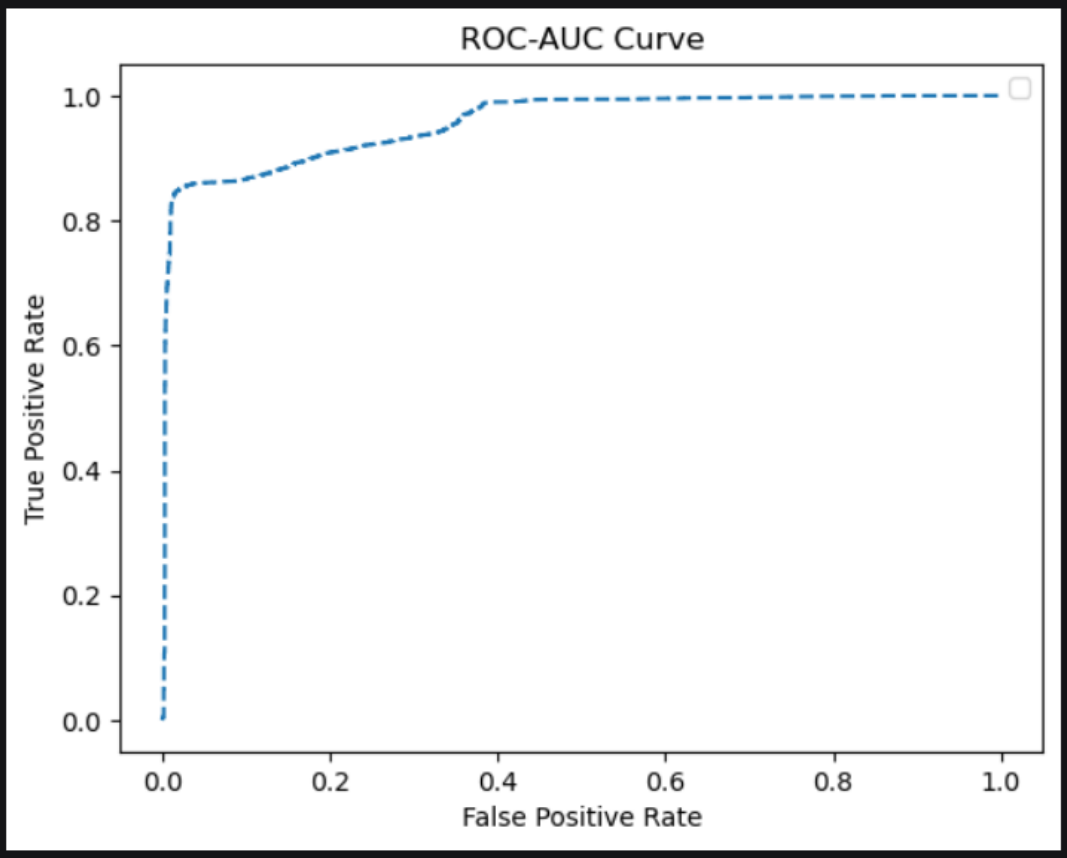
Classification report: تجزیه و تحلیل دقیق تر از عملکرد مدل، از جمله precision, recall, and F1-score

```
fpr, tpr, threshold = roc_curve(y_test,y_prob[:,1])
print("ROC-AUC:",auc(fpr,tpr))
```

```
ROC-AUC: 0.9573065076870617
```

```
plt.plot(fpr,tpr,linestyle='--')
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title("ROC-AUC Curve")
plt.legend()
plt.show()
```

```
No artists with labels found to put in legend.  Note that artists whose label start
```



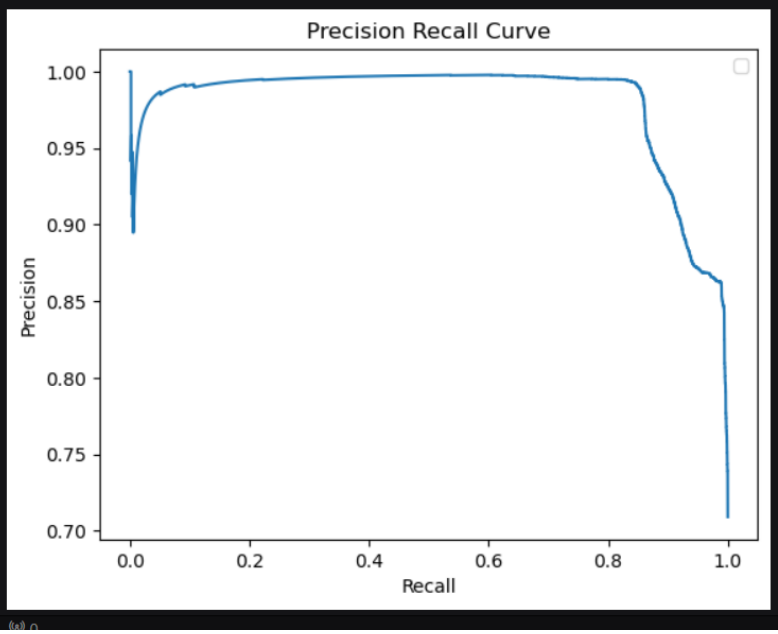
منحنی ROC-AUC: منحنی ROC (ویژگی عملیاتی گیرنده) را محاسبه می کند و امتیاز ناحیه زیر منحنی (AUC) را چاپ می کند. سپس، منحنی ROC را رسم می کند.

```
p, r, threshold = precision_recall_curve(y_test,y_prob[:,1])
print("PR-AUC:",auc(r,p))
```

```
PR-AUC: 0.9797335919299746
```

```
plt.plot(r, p)
plt.xlabel('Recall')
plt.ylabel('Precision')
plt.title("Precision Recall Curve")
plt.legend()
plt.show()
```

```
No artists with labels found to put in legend.  Note that artists whose label start with an underscore are ignored w
```



Precision-Recall Curve: منحنی فراخوان دقیق را محاسبه می کند و امتیاز AUC را چاپ می کند. سپس، منحنی فراخوان دقیق را ترسیم می کند.