



دانشگاه صنعتی امیرکبیر

(پلی تکنیک تهران)

دانشکده ریاضی و علوم کامپیوتر

گزارش پنجم درس هوش مصنوعی

نگارش

آرمان صالحی

استاد راهنما

دکتر مهدی قطعی

بهار 1403

چکیده

در این برنامه ما دیتاست Banck Account Fraud Dataset Suit را بررسی میکنیم، و از طریق الگوریتم K-Means کاربران خود را بر اساس اطلاعاتی که از آنها داریم به سه دسته کلی تقسیم میکنیم.

صفحه

فهرست مطالب

چکیده.....أ

گزارش کار3

گزارش کار

- انتخاب دیتاست و پردازش پیش داده

```
1 import pandas as pd
2 from sklearn.model_selection import train_test_split
3 from sklearn.preprocessing import StandardScaler
4 from sklearn.cluster import KMeans
5 import matplotlib.pyplot as plt
6 from lazypredict.Supervised import LazyClassifier
7 from sklearn.metrics import accuracy_score, f1_score
8
9 data = pd.read_csv('bank_account_fraud.csv')
```

این کد کتابخانه‌های مورد نیاز را وارد می‌کند و سپس دیتاست را از مسیر مشخص شده بارگذاری می‌کند.

```
11 print(data.info())
```

این کد اطلاعات کلی از ساختار داده‌ها را نمایش می‌دهد، مانند تعداد رکوردها و نوع داده‌های هر ستون.

```
13 X = data.drop('target', axis=1)
14 y = data['target']
```

در اینجا فرض شده است که ستون هدف (label) به نام 'target' است. ستون 'target' از داده‌ها جدا شده و بقیه ستون‌ها به عنوان ویژگی‌ها ('features') استفاده می‌شوند.

```
16 X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)
```

داده‌ها به دو قسمت 'train' و 'test' تقسیم می‌شوند. ۷۰٪ داده‌ها برای آموزش و ۳۰٪ برای تست مدل استفاده می‌شوند.

```
18 scaler = StandardScaler()
19 X_train = scaler.fit_transform(X_train)
20 X_test = scaler.transform(X_test)
```

ویژگی‌ها استانداردسازی می‌شوند تا مقادیر آنها به مقیاسی مشابه تبدیل شوند. این کار کمک می‌کند تا مدل‌ها بهتر عمل کنند.

- پیاده‌سازی مدل خوشه‌بندی

```
22 kmeans = KMeans(n_clusters=3, random_state=42)
23 kmeans.fit(X_train)
```

در این بخش از الگوریتم K-Means برای خوشه‌بندی داده‌ها استفاده می‌شود. این کد الگوریتم K-Means را پیاده‌سازی می‌کند و تعداد خوشه‌ها را ۳ تنظیم می‌کند.

- اضافه کردن برچسب خوشه به داده‌ها

```
25 data['cluster'] = kmeans.labels_
```

برچسب خوشه‌های ایجاد شده به داده‌ها اضافه می‌شود.

```
27 plt.scatter(X_train[:, 0], X_train[:, 1], c=kmeans.labels_, cmap='viridis')
28 plt.xlabel('Feature 1')
29 plt.ylabel('Feature 2')
30 plt.title('K-Means Clustering')
31 plt.show()
```

این کد نمودار پراکندگی داده‌ها را با رنگ‌بندی بر اساس خوشه‌های ایجاد شده نمایش می‌دهد. این نمودار می‌تواند به شما کمک کند تا نتایج خوشه‌بندی را تحلیل کنید.

- پیاده‌سازی مدل طبقه‌بندی

```

33 clf = LazyClassifier()
34 models, predictions = clf.fit(X_train, X_test, y_train, y_test)
35
36 print(models)

```

در این بخش از کتابخانه `lazypredict` برای مقایسه مدل‌های مختلف طبقه‌بندی استفاده می‌شود. این کد از کتابخانه `lazypredict` استفاده می‌کند تا مدل‌های مختلف طبقه‌بندی را با یکدیگر مقایسه کند و بهترین مدل‌ها را براساس داده‌های شما معرفی کند. نتایج این مقایسه در متغیر `models` ذخیره می‌شود.

```

40 model = RandomForestClassifier(random_state=42)
41 model.fit(X_train, y_train)
42 y_pred = model.predict(X_test)

```

در اینجا یک مدل خاص (مثلاً RandomForestClassifier) انتخاب و پیاده‌سازی می‌شود. شما می‌توانید مدل دیگری که بهترین عملکرد را دارد انتخاب کنید.

```

44 accuracy = accuracy_score(y_test, y_pred)
45 f1 = f1_score(y_test, y_pred, average='weighted')
46
47 print(f'Accuracy: {accuracy}')
48 print(f'F1 Score: {f1}')

```

در این قسمت، مدل ارزیابی می‌شود و معیارهای دقت (accuracy) و امتیاز F1 محاسبه می‌شوند. این معیارها به شما کمک می‌کنند تا عملکرد مدل را بسنجید.