

# به نام خدا

علی میرش زاده جبدرق ۹۹۱۳۰۲۸

میان ترم عملی درس هوش مصنوعی خلاصه گزارش کد و پاسخ سوالات تشریحی

استاد درس: دکتر مهدی قطعی

استاد کارگاه : بهنام یوسفی مهر

1. ابتدا، داده‌های بازیکنان فوتبال از یک فایل CSV خوانده شده و فقط ویژگی‌های عددی بازیکنان با امتیاز بالاتر از 85 در نظر گرفته شده‌اند. سپس از روش PCA برای تحلیل مؤلفه‌های اصلی بهره گرفته شده و داده‌ها به دو مؤلفه اصلی کاهش داده شده‌اند. در مرحله بعدی، از روش خوشه‌بندی K-means برای تقسیم این بازیکنان به 5 خوشه مختلف استفاده شده و نموداری از این خوشه‌ها به همراه نام بازیکنان ترسیم شده است.

در بخش دیگر، برخی از بازیکنان خاص (مانند امپایه، رونالدو، مسی و مهدی طارمی) به خوشه‌های مربوطه تعلقشان پیش‌بینی شده است. و در پایان، با استفاده از موقعیت بهترین بازیکنان بر روی زمین، از روش خوشه‌بندی K-means با 16 خوشه برای تجزیه و تحلیل دقت خوشه‌بندی بر اساس موقعیت بازیکنان استفاده شده است. دقت این خوشه‌بندی برای هر موقعیت بهترین نمایش داده شده است.

2. این کد برای بارگیری و نمایش تصاویر از فایل‌های cdb که دارای اعداد دست‌نوشته فارسی هستند استفاده می‌کند. ابتدا داده‌های آموزشی، آزمایشی و باقیمانده را بارگیری و نمایش داده، سپس از داده‌های آموزشی برای محاسبه پروتوتایپ هر کلاس استفاده می‌شود و نتایج نمایش داده می‌شوند.

در بخش بعدی، از نمونه‌های آزمایشی برای ارزیابی یک طبقه‌بند (MDC) استفاده می‌شود که بر اساس پروتوتایپ‌های کلاس‌ها عمل می‌کند. نرخ خطا محاسبه شده و پنج پیش‌بینی نادرست نمایش داده می‌شوند.

3. a)

```
top_n_per_group = df.groupby('A')['C'].nlargest(N).reset_index()
```

b)

```
# Calculate total sales per month
```

```
total_sales_per_month = sales_data.groupby('Month')['Sales'].sum()
```

```
# Merge this back into the original dataframe
```

```
sales_data = sales_data.merge(total_sales_per_month, on='Month', suffixes=('', '_Total'))
```

```
# Calculate the percentage contribution
```

```
sales_data['Percentage_Contribution'] = (sales_data['Sales'] / sales_data['Sales_Total']) * 100
```

c)

```
from scipy.ndimage import median_filter
```

```
filtered_image = median_filter(arr, size=3)
```

d)

```
import numpy as np
```

```
# Calculate the absolute differences between each element in b and all elements in a
```

```
differences = np.abs(a[:, np.newaxis] - b)
```

```
# Find the index of the minimum value in each column
```

```
closest_indices = np.argmin(differences, axis=0)
```