Московский государственный технический университет им. Н.Э. Баумана

Кафедра «Системы обработки информации и управления»

Лабораторная работа №1

по дисциплине

«Методы машинного обучения»

на тему

«Разведочный анализ данных. Исследование и визуализация данных»

Выполнила:

студентка Цзян Юхуэй

группы ИУ5И-23М

Москва — 2024 г.

1. Цель лабораторной работы

Изучение различных методов визуализация данных и создание истории на основе данных.

2. Задание

- Выбрать набор данных (датасет). Вы можете найти список свободно распространяемых датасетов здесь.

Для лабораторных работ не рекомендуется выбирать датасеты очень большого размера.

- Создать "историю о данных" в виде юпитер-ноутбука, с учетом следующих требований:

1. История должна содержать не менее 5 шагов (где 5 - рекомендуемое количество шагов). Каждый шаг содержит график и его текстовую интерпретацию.
2. На каждом шаге наряду с удачным итоговым графиком рекомендуется в юпитер-ноутбуке оставлять результаты предварительных "неудачных" графиков.
3. Не рекомендуется повторять виды графиков, желательно создать 5 графиков различных видов.
4. Выбор графиков должен быть обоснован использованием методологии data-to-viz. Рекомендуется учитывать типичные ошибки построения выбранного вида графика по методологии data-to-viz. Если методология Вами отвергается, то просьба обосновать Ваше решение по выбору графика.
5. История должна содержать итоговые выводы. В реальных "историях о данных" именно эти выводы представляют собой основную ценность для предприятия.

- Сформировать отчет и разместить его в своем репозитории на github.

3. Ход выполнения работы

3.1. Текстовое описание набора данных

В этом эксперименте я буду визуализировать и анализировать отобранные данные всех футбольных матчей, начиная с 1872 года и до наших дней.

- **The dataset includes the following columns:**
  - Date
  - Host Team
  - Away Team
  - Host Team Score
  - Away Team Score
  - Match Type
  - Host City
  - Country

3.2. Основные характеристики набора данных

Из-за огромного объема данных мы сначала отфильтровали их, чтобы сохранить только данные о совпадениях для десяти типов совпадений с наибольшим количеством типов совпадений.

```
# 统计比赛类型举行次数最多的十种比赛类型
top10_match_types = football_data['比赛类型'].value_counts().head(10).index.tolist()

# 过滤数据，只保留这十种比赛类型的数据
football_data_top10 = football_data[football_data['比赛类型'].isin(top10_match_types)]
```

3.2.1 Процент побед домашней команды

```
total_matches = len(football_data)

home_wins = football_data[football_data['主队得分'] > football_data['客队得分']]

home_win_percentage = len(home_wins) / total_matches * 100

plt.figure(figsize=(8, 8))

plt.pie([home_win_percentage, 100 - home_win_percentage], labels=['主队胜利', '客队胜利'], autopct='%1.1f%%', colors=['skyblue', 'salmon'])

plt.title('主队胜利比率')

plt.show()
```

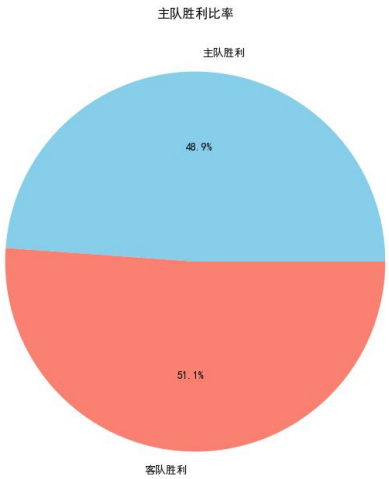

Рис 1. Процент побед домашней команды

3.2.2 Процент побед домашней команды в различных типах матчей

```
match_type_win_percentage_top10 = football_data_top10.groupby('比赛类型').apply(lambda x: (x['主队得分'] > x['客队得分']).mean() * 100)

plt.figure(figsize=(10, 6))

match_type_win_percentage_top10.plot(kind='bar', color='lightgreen')

plt.title('各种类型的比赛主队的胜利比率')

plt.ylabel('主队胜利比率')

plt.xlabel('比赛类型')

plt.xticks(rotation=45)

plt.show()
```

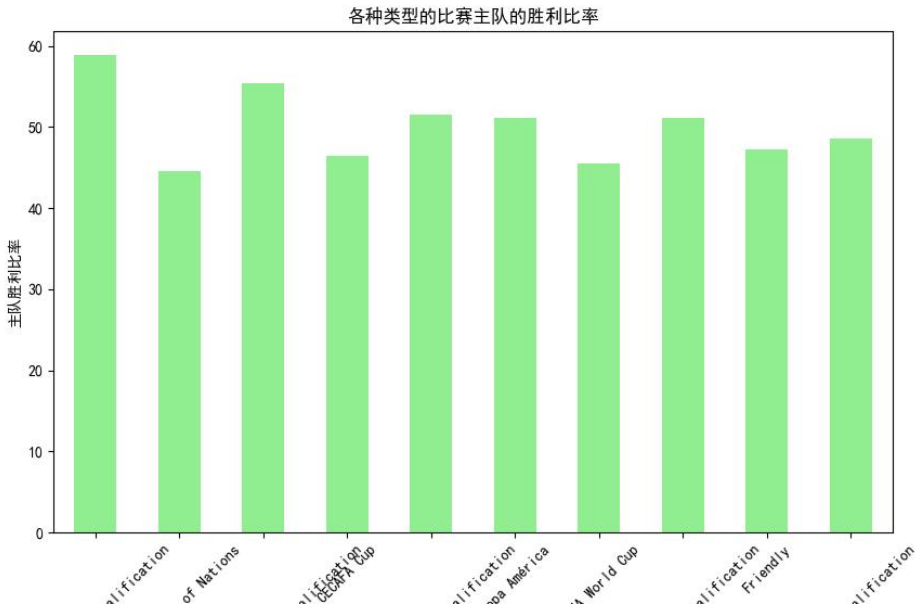

Рис 2. Процент побед домашней команды в различных типах матчей

3.2.3 Количество игр, в которых разница между командами хозяев и гостей во всех типах игр превышала 5 очков

```
high_scoring_matches = football_data_top10[abs(football_data_top10['主队得分'] - football_data_top10['客队得分']) > 5]

high_scoring_matches_by_type = high_scoring_matches['比赛类型'].value_counts()

plt.figure(figsize=(10, 6))

sns.boxplot(x='比赛类型', y='主队得分', data=high_scoring_matches, color='orange')
```

```
plt.title('比赛分差超过 5 分的比赛主队得分分布')
plt.xlabel('比赛类型')
plt.ylabel('主队得分')
plt.xticks(rotation=45)
plt.show()
```

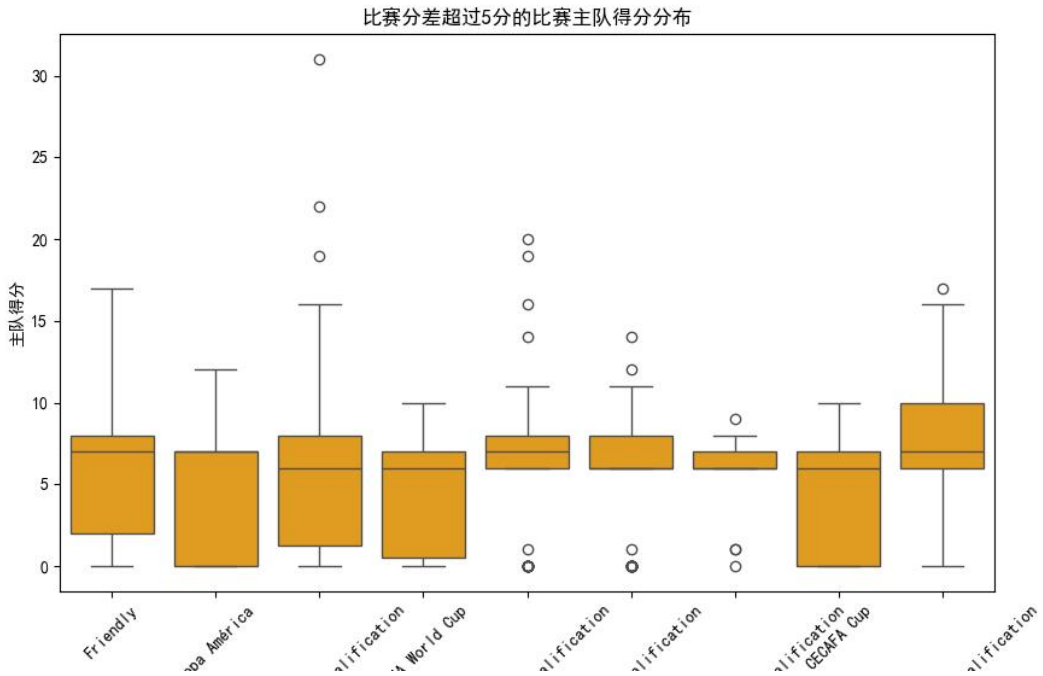

Рис 3. Количество игр, в которых разница между командами хозяев и гостей во всех типах игр превышала 5 очков

3.2.4 Изменение количества сыгранных игр с течением времени в статистике за 20 лет

```
football_data['年份'] = football_data['日期'].dt.year
matches_by_year_top10 = football_data_top10.groupby(football_data_top10['日期'].dt.year // 20 * 20)['日期'].count()
plt.figure(figsize=(10, 6))
matches_by_year_top10.plot(kind='line', marker='o', color='lightblue')
plt.title('20 年为一组统计的比赛场数随时间的变化')
plt.ylabel('比赛场数')
plt.xlabel('年份组')
```

```
plt.xticks(rotation=45)

plt.grid(True)  # 添加网格线

plt.show()
```

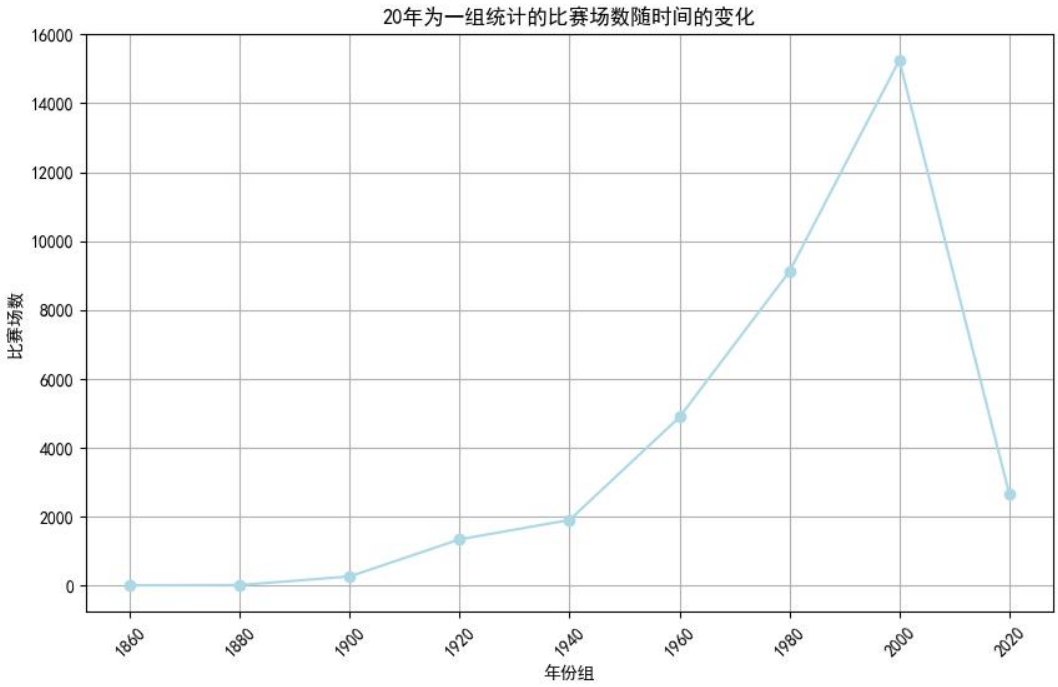

Рис 4. Изменение количества сыгранных игр с течением времени в статистике за 20 лет

3.2.5 Победы в десяти городах, принимающих наибольшее количество футбольных матчей

```
top10_cities = football_data['举办城市'].value_counts().head(10).index.tolist()

top10_cities_data = football_data[football_data['举办城市'].isin(top10_cities)]

city_wins_top10 = football_data_top10[football_data_top10['举办城市'].isin(top10_cities_top10)].groupby(['举办城市', '比赛类型
']).apply(lambda x: (x['主队得分'] > x['客队得分']).mean() * 100)

plt.figure(figsize=(12, 8))

sns.heatmap(city_wins_top10.unstack(), cmap='coolwarm', annot=True, fmt=".1f")

plt.title('举办足球比赛最多的十个城市的胜利情况')

plt.xlabel('比赛类型')

plt.ylabel('城市')
```

```
plt.tight_layout()

plt.show()
```

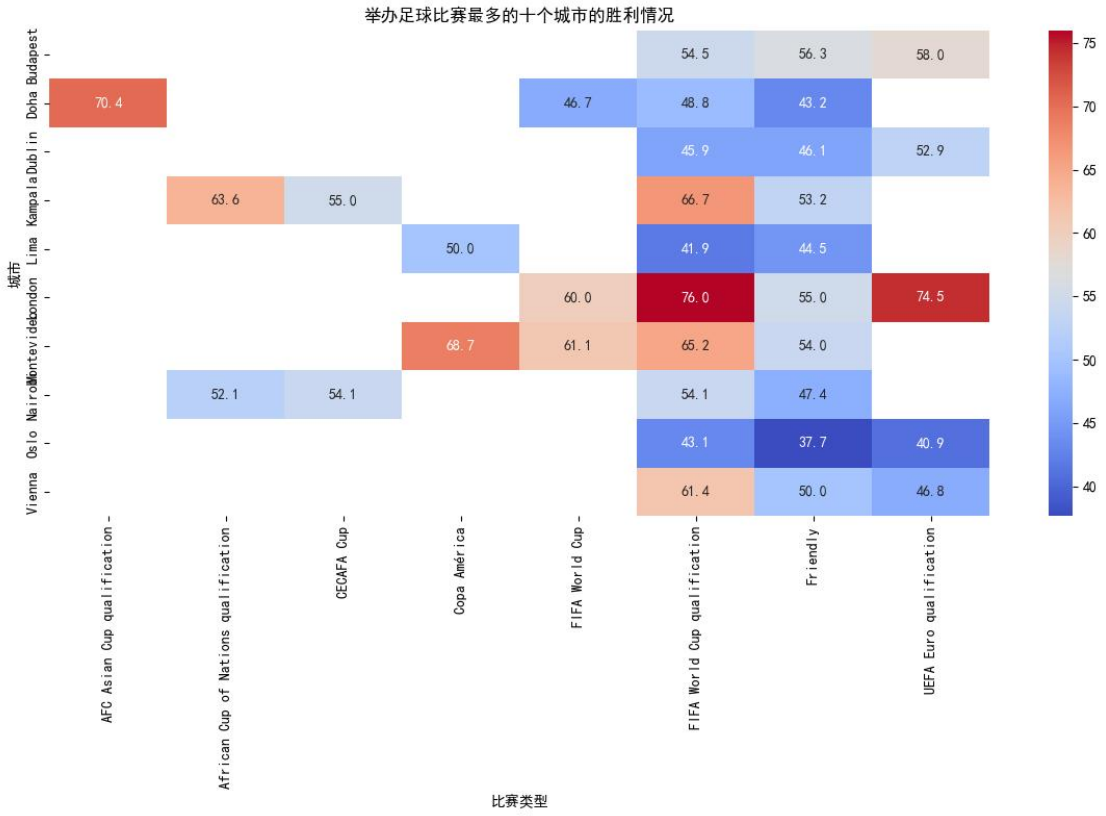

Рис 5. Победы в десяти городах, принимающих наибольшее количество футбольных матчей