تکلیف اول درس مبانی داده کاوی

امید رئیسی (۹۶۲۱۱۶۰۰۱۵)

۱)

a) یادگیری با ناظر

b) یادگیری بدون ناظر

c) یادگیری با ناظر

d) یادگیری بدون ناظر

e) یادگیری با ناظر

f) یادگیری با ناظر

g) یادگیری با ناظر

h) یادگیری بدون ناظر

۲) داده‌های آموزشی داده‌های برچسب داری هستند که در الگوریتم‌های با ناظر برای یادگیری الگوریتم بکار گرفته می‌شوند و در واقع روابط میان متغییرهای پیشگو(ورودی) و خروجی را به الگوریتم نشان می‌دهند اما در مقابل داده‌های اعتبارسنجی داده‌های برچسب داری هستند که پس از اتمام یادگیری الگوریتم برای ارزیابی آن استفاده می‌شوند.

۳) با بررسی اولیه داده‌ها متوجه می‌شویم که کلاس هدف به صورت نامتوازن می‌باشد به صورتی که ۷۸٪ از داده‌ها از طبقه هدف **0** و مابقی از طبقه هدف **1** می‌باشند. ابتدا باید داده‌ها را در یک جدول مرتب سازی کنیم و پس از بررسی اینکه آیا متغییر بدون مقداری وجود دارد یا نه (بررسی **missing value** ها) باید برای داده‌های آموزشی خود به صورتی نمونه برداری کنیم که تعادل میان هر دو طبقه هدف برقرار باشد.پس از متعادل کردن داده‌ها حدود **80**٪ آن‌ها را به عنوان داده آموزشی و مابقی را به عنوان داده‌های اعتبارسنجی انتخاب می‌کنیم.

حال برای مدل خود با استفاده از کتابخانه‌های پایتون و توابع رگرسیون خطی تأثیر هر متغییر ورودی را بر متغییر هدف پیدا کرده و مدل خود را با داده‌های اعتبارسنجی آزمایش می‌کنیم.(ممکن است در مراحل مدل سازی با مصور سازی داده‌ها و استفاده از خلاصه سازی داده‌ها مانند ماتریس همبستگی از روش های کاهش ابعاد نیز استفاده کنیم.)

۴) تعداد رکوردها برابر با **1000** و تعداد متغیرها برابر با **50** می‌باشد پس در کل **50000** مقدار داریم که **5**٪ آن برابر با **2500** می باشد:($\mathbf{50 * 1000 = 50000}$)

اگر بدترین حالت را در نظر بگیریم به صورتی که هر رکورد به صورت میانگین به اندازه **2.5** مقدار نداشته باشد آنگاه همه رکوردها حذف می‌شوند.($\frac{\mathbf{2500}}{\mathbf{1000}} = \mathbf{2.5}$)

اگر بهترین حالت را در نظر بگیریم به صورتی که رکوردها همه متغییرهایشان بی مقدار باشند آنگاه باید **50** رکورد را حذف کنیم.($\frac{\mathbf{2500}}{\mathbf{50}} = \mathbf{50}$)

۵)

الف)متغییرهای **Color** و **Fuel_Type** متغیرهای طبقه‌ای هستند که به ترتیب دارای **10** و **3** کلاس می باشند.

Color = ['Beige', 'Black', 'Blue', 'Green', 'Grey', 'Red', 'Silver', 'Violet', 'White', 'Yellow']

Fuel_Type = ['CNG', 'Petrol', 'Diesel']

برای مثال اگر متغییر **Color** را در نظر بگیریم می‌توانیم این **10** کلاس را در **9** متغییر دودویی که هر کدام نماینده یکی از این رنگ‌ها می‌باشد ذخیره کنیم (به جز رنگ **Beige**) بدین ترتیب که اگر متغییر مربوط به رنگ داری مقدار **1** بود بدین ترتیب رنگ ماشین برابر با آن رنگ است (**0** بودن تمامی متغییرها نشان می‌دهد که رنگ ماشین **Beige** است.)

به همین ترتیب برای متغییر **Fuel_Type** اگر تمام متغییرهای جایگزین **0** باشند نوع سوخت مصرفی ماشین **CNG** می‌باشد.

این جایگزینی متغییرها را (تبدیل متغییرهای طبقه‌ای به چند متغییر باینری و عددی را که به آن‌ها **Dummy Variables** می‌گوییم) در پایتون با استفاده از کتابخانه **pandas** و متود **get_dummies** انجام می‌شود.

```
In [15]: import pandas as pd


Toyota_Corolla_df = pd.read_csv(r"./ToyotaCorolla.csv")

Toyota_Corolla_df.columns = [s.strip().replace(" ", "_") for s in Toyota_Corolla_df.columns]

Toyota_Corolla_df_temp = pd.get_dummies(Toyota_Corolla_df.iloc[:, 2:], prefix_sep="_", drop_first=True)

Toyota_Corolla_df = pd.concat([Toyota_Corolla_df.iloc[:, :2], Toyota_Corolla_df_temp], axis=1)

print(Toyota_Corolla_df)
```

```
      Fuel_Type_Petrol  Color_Black  Color_Blue  Color_Green  Color_Grey  \
0                    0            0           1            0           0
1                    0            0           0            0           0
2                    0            0           1            0           0
3                    0            1           0            0           0
4                    0            1           0            0           0
...                ...          ...         ...          ...         ...
1431                 1            0           1            0           0
1432                 1            0           0            0           1
1433                 1            0           1            0           0
1434                 1            0           0            0           1
1435                 1            0           0            1           0

      Color_Red  Color_Silver  Color_Violet  Color_White  Color_Yellow
0             0             0             0            0             0
1             0             1             0            0             0
2             0             0             0            0             0
3             0             0             0            0             0
4             0             0             0            0             0
```

ب)برای آماده سازی داده‌ها ابتدا باید بررسی کنیم که آیا **missing value** داریم یا نه اما چون داده‌ها کامل هستند مستقیم سراغ افراز داده‌ها می‌رویم.

افراز داده‌ها را با استفاده از متود **train_test_split** از کتابخانه **scikit-learn** در پایتون انجام می‌دهیم.

```
In [16]: import pandas as pd
         from sklearn.model_selection import train_test_split

         Toyota_Corolla_df = pd.read_csv(r"./ToyotaCorolla.csv")

         Toyota_Corolla_df.columns = [s.strip().replace(" ", "_") for s in Toyota_Corolla_df.columns]

         training_data, temp_data = train_test_split(Toyota_Corolla_df, test_size=0.5, random_state=1)

         test_data, evaluation_data = train_test_split(temp_data, test_size=0.4, random_state=1)

         print("training_data : ", training_data.shape)
         print("test_data : ", test_data.shape)
         print("evaluation_data : ", evaluation_data.shape)

         training_data :  (718, 39)
         test_data :  (430, 39)
         evaluation_data :  (288, 39)
```

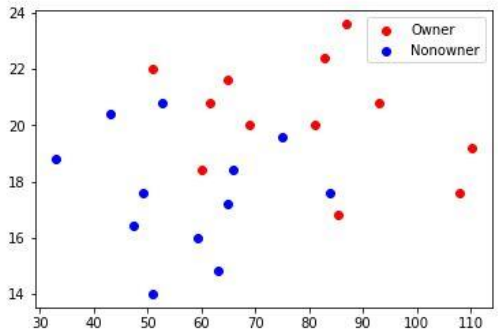
training_data (داده‌های آموزشی) : داده‌هایی که برای آموزش الگوریتم‌های خود استفاده می‌کنیم تا آنها را تبدیل به مدل کنیم.

test_data (داده‌های اعتبارسنجی) : داده‌هایی که برای مقایسه مدل‌های بدست آمده استفاده می‌کنیم تا بهترین مدل را پیدا کنیم.

evaluation_data (داده‌های ارزشیابی) : داده‌هایی که برای ارزیابی مدل منتخب بر روی داده‌های جدید استفاده می‌شوند تا از عدم رخداد بیش برازش بر روی داده‌های **test_data** مطمئن شویم.

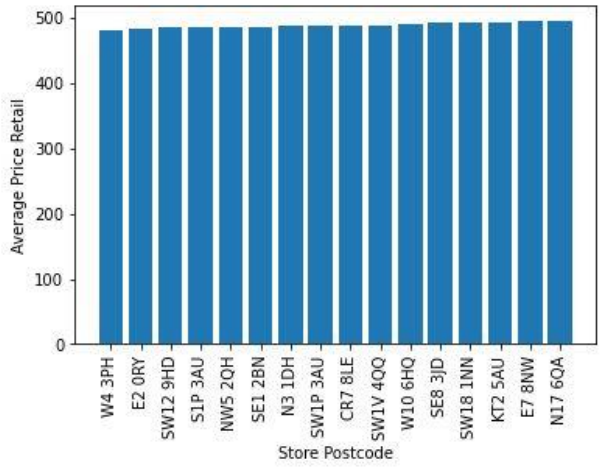
۶) برای رسم نمودار از کتابخانه **matplotlib** استفاده می‌کنیم.

In [14]:
```
import pandas as pd
import matplotlib.pyplot as plt


Riding_Mowers_df = pd.read_csv(r"./RidingMowers .csv")

plt.scatter(
    x=[data.Income for data in Riding_Mowers_df.iloc if data.Ownership == "Owner"],
    y=[data.Lot_Size for data in Riding_Mowers_df.iloc if data.Ownership == "Owner"],
    c="red",
)

plt.scatter(
    x=[data.Income for data in Riding_Mowers_df.iloc if data.Ownership == "Nonowner"],
    y=[data.Lot_Size for data in Riding_Mowers_df.iloc if data.Ownership == "Nonowner"],
    c="blue",
)

plt.legend(["Owner", "Nonowner"])
plt.show()
```



۷) الف) همانطور که از شکل زیر مشخص است بیشترین میانگین خرده فروشی مربوط به شعبه **N17 6QA** به ارزش **494** دلار و کمترین میانگین خرده فروشی مربوط به شعبه **W4 3PH** به ارزش**481** دلار می‌باشد.

In [4]:
```
import pandas as pd
import matplotlib.pyplot as plt


Laptop_Sales_df = pd.read_csv(r"./LaptopSalesJanuary2008.csv")

Laptop_Sales_df.rename(
    columns={"Store Postcode": "SP", "Retail Price": "RP"}, inplace=True
)

data_for_plot = Laptop_Sales_df.groupby("SP").mean()["RP"].sort_values()

plt.bar(data_for_plot.index, data_for_plot)
plt.xlabel("Store Postcode")
plt.ylabel("Average Price Retail")
plt.xticks(rotation=90)


plt.show()
```



ب) در مقایسه **N17 6QA** (بالاترین) با **W4 3PH** (پایین‌ترین) میانه ها مشابه هستند، اما **W4 3PH** محدوده قیمت‌های بزرگ‌تری دارد و خارج از چارک‌های **1** و **4** پرت‌تر است.

In [1]:
```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np


Laptop_Sales_df = pd.read_csv(r"./LaptopSalesJanuary2008.csv")

Laptop_Sales_df.rename(
    columns={"Store Postcode": "SP", "Retail Price": "RP"}, inplace=True
)


data_for_plot = []

for store_postcode in Laptop_Sales_df.SP.unique():
    data = [s.RP for s in Laptop_Sales_df.iloc if s.SP == store_postcode]
    data_for_plot.append(data)


plt.boxplot(data_for_plot, labels=Laptop_Sales_df.SP.unique())
plt.xlabel("Store Postcode")
plt.ylabel("Average Price Retail")
plt.xticks(rotation=90)


plt.show()
```