silas_assignment1

April 19, 2025

Leon Silas

Intro to Data Science - CAP 5768

0.0.1 Assignment 1

[130]:
```
#imports
import numpy as np
import pandas as pd
import scipy.stats as stats
import matplotlib.pyplot as plt
from sklearn.metrics import ConfusionMatrixDisplay

#import csv into df
df = pd.read_csv('Assignment 1 - Automobile_data.csv')
```

Question 1 :

a) print the first and last five rows (5 Points)

[131]:
```
# print first 5 rows
print("First 5 rows of the dataframe:")
print(df.head(5))

# print last 5 rows
print("\nLast 5 rows of the dataframe:")
print(df.tail(5))
```

```
First 5 rows of the dataframe:
   index      company   body-style  wheel-base  length engine-type  \
0      0  alfa-romero  convertible        88.6   168.8        dohc
1      1  alfa-romero  convertible        88.6   168.8        dohc
2      2  alfa-romero    hatchback        94.5   171.2        ohcv
3      3         audi        sedan        99.8   176.6         ohc
4      4         audi        sedan        99.4   176.6         ohc

  num-of-cylinders  horsepower  average-mileage    price
0             four         111               21  13495.0
1             four         111               21  16500.0
```

```
2              six         154               19  16500.0
3             four         102               24  13950.0
4             five         115               18  17450.0

Last 5 rows of the dataframe:
    index     company body-style  wheel-base  length engine-type  \
56     81  volkswagen      sedan        97.3   171.7         ohc
57     82  volkswagen      sedan        97.3   171.7         ohc
58     86  volkswagen      sedan        97.3   171.7         ohc
59     87       volvo      sedan       104.3   188.8         ohc
60     88       volvo      wagon       104.3   188.8         ohc

   num-of-cylinders  horsepower  average-mileage    price
56             four          85               27   7975.0
57             four          52               37   7995.0
58             four         100               26   9995.0
59             four         114               23  12940.0
60             four         114               23  13415.0
```

b) print most expensive car's company name and price (5 Points)

[132]:
```
# find most expensive car
top = df.loc[df['price'].idxmax()]
print(top[['company', 'price']])
```

```
company    mercedes-benz
price            45400.0
Name: 35, dtype: object
```

c) print All Toyota Cars details (5 Points)

[133]:
```
# extract toyota cars
companies = df.groupby('company')
toyota = companies.get_group('toyota')
print(toyota)
```

```
    index company body-style  wheel-base  length engine-type num-of-cylinders  \
48     66  toyota  hatchback        95.7   158.7         ohc             four
49     67  toyota  hatchback        95.7   158.7         ohc             four
50     68  toyota  hatchback        95.7   158.7         ohc             four
51     69  toyota      wagon        95.7   169.7         ohc             four
52     70  toyota      wagon        95.7   169.7         ohc             four
53     71  toyota      wagon        95.7   169.7         ohc             four
54     79  toyota      wagon       104.5   187.8        dohc              six

    horsepower  average-mileage    price
48          62               35   5348.0
49          62               31   6338.0
50          62               31   6488.0
51          62               31   6918.0
```

```
52          62               27   7898.0
53          62               27   8778.0
54         156               19  15750.0
```

d) print total cars count per company (5 Points)

[134]:
```
# count the number of cars for each company
counts = companies.size()
print(counts)
```

```
company
alfa-romero      3
audi             4
bmw              6
chevrolet        3
dodge            2
honda            3
isuzu            3
jaguar           3
mazda            5
mercedes-benz    4
mitsubishi       4
nissan           5
porsche          3
toyota           7
volkswagen       4
volvo            2
dtype: int64
```

e) sort all cars by Price column (5 Points)

[135]:
```
# sort cars by price from highest to lowest
sorted_cars = df.sort_values(by='price', ascending=False)
print(sorted_cars)
```

```
    index        company   body-style  wheel-base  length engine-type  \
35     47  mercedes-benz      hardtop       112.0   199.2        ohcv
11     14            bmw        sedan       103.5   193.8         ohc
34     46  mercedes-benz        sedan       120.9   208.1        ohcv
46     62         porsche  convertible        89.5   168.9        ohcf
12     15            bmw        sedan       110.0   197.0         ohc
..    …             …          …         …     …         …
27     36          mazda    hatchback        93.1   159.1         ohc
13     16      chevrolet    hatchback        88.4   141.1           l
22     31          isuzu        sedan        94.5   155.9         ohc
23     32          isuzu        sedan        94.5   155.9         ohc
47     63         porsche    hatchback        98.4   175.7       dohcv

   num-of-cylinders  horsepower  average-mileage    price
35            eight         184               14  45400.0
```

```
11              six         182              16  41315.0
34            eight         184              14  40960.0
46              six         207              17  37028.0
12              six         182              15  36880.0
..              …           …                …      …
27             four          68              30   5195.0
13            three          48              47   5151.0
22             four          70              38      NaN
23             four          70              38      NaN
47            eight         288              17      NaN

[61 rows x 10 columns]
```

Question 2 : A manufacturing company claims that their new production process will result in a mean weight of 100 grams for their product packaging. To test this claim, a sample of 50 packaging units is taken, and the mean weight is found to be 98.5 grams with a standard deviation of 2.3 grams.

a) Formulate the null and alternative hypotheses for this scenario.

Null hypothesis (H0): There is not a significant difference in the mean weight of the product packaging. (Mean = 100)

Alternative hypotheses (H1): There is a significant difference in the mean weight of the product packaging. (Mean =/ 100)

b) Calculate the test statistic and p-value for testing the hypothesis (you can use a program code or you can calculate by-hand).

[136]:
```
# Set sample size, mean, and std
mean = 98.5
expected_mean = 100
std = 2.3
n = 50
df = n - 1

# Manual calculation of t-statistic and p-value
t_manual = (mean - expected_mean) / (std / np.sqrt(n))
p_manual = 2 * stats.t.sf(abs(t_manual), df)

print(f"Manual t-statistic: {t_manual}")
print(f"Manual p-value: {p_manual}")
```

```
Manual t-statistic: -4.611565964260093
Manual p-value: 2.8891515067376308e-05
```

c) At a significance level of 0.05, what is your conclusion regarding the company's claim?

At a significance level of 0.05 and a p-value of 0.0000028 we reject the null hypothesis and conclude there is a stastically significant difference in the mean of the packaging from 100g. The company's claim is not supported.

Question 3 : An educational institute claims that the average score of its students in a standardized test is at least 75. To test this claim, a random sample of 40 students is taken, and their average score is found to be 72.8 with a standard deviation of 5.6.

a) State the null and alternative hypotheses.

Null hypothesis (H0): There is not a significant difference in the average standardized test scores of the students. (Mean >= 75)

Alternative hypotheses (H1): There is a significant difference in the average standardized test scores of the students. (Mean < 75)

b) Compute the test statistic and p-value. (you can use a program code or you can calculate by-hand).

[137]:
```
# Set metrics
mean = 72.8
expected_mean = 75
std = 5.6
n = 40
df = n - 1

# Manual calculation of t-statistic and p-value
t_statistic = (mean - expected_mean) / (std / np.sqrt(n))
p_value = stats.t.cdf(t_statistic, df)

print(f"Manual t-statistic: {t_statistic}")
print(f"Manual p-value: {p_value}")
```

```
Manual t-statistic: -2.4846467329894444
Manual p-value: 0.00868357576692573
```

c) Based on a significance level of 0.01, what is your conclusion regarding the institute's claim?

At a significance level of 0.05 and a p-value of 0.008 we reject the null hypothesis and conclude there is a stastically significant difference in the student test scores being lower from the claimed 75 average. The institution's claim is not supported.

Question 4 : A machine learning model was trained to classify emails as either spam (positive class) or non-spam (negative class). After testing the model on a dataset of 200 emails, the following results were obtained: True Positives (TP): 85 , False Positives (FP): 10, True Negatives (TN): 90, False Negatives (FN): 15

a) Calculate the accuracy, precision, recall, and F1-score of the model.

[138]:
```
# Set TP/TN/FP/FN values
total = 200
TP = 85
TN = 90
FP = 10
FN = 15
```

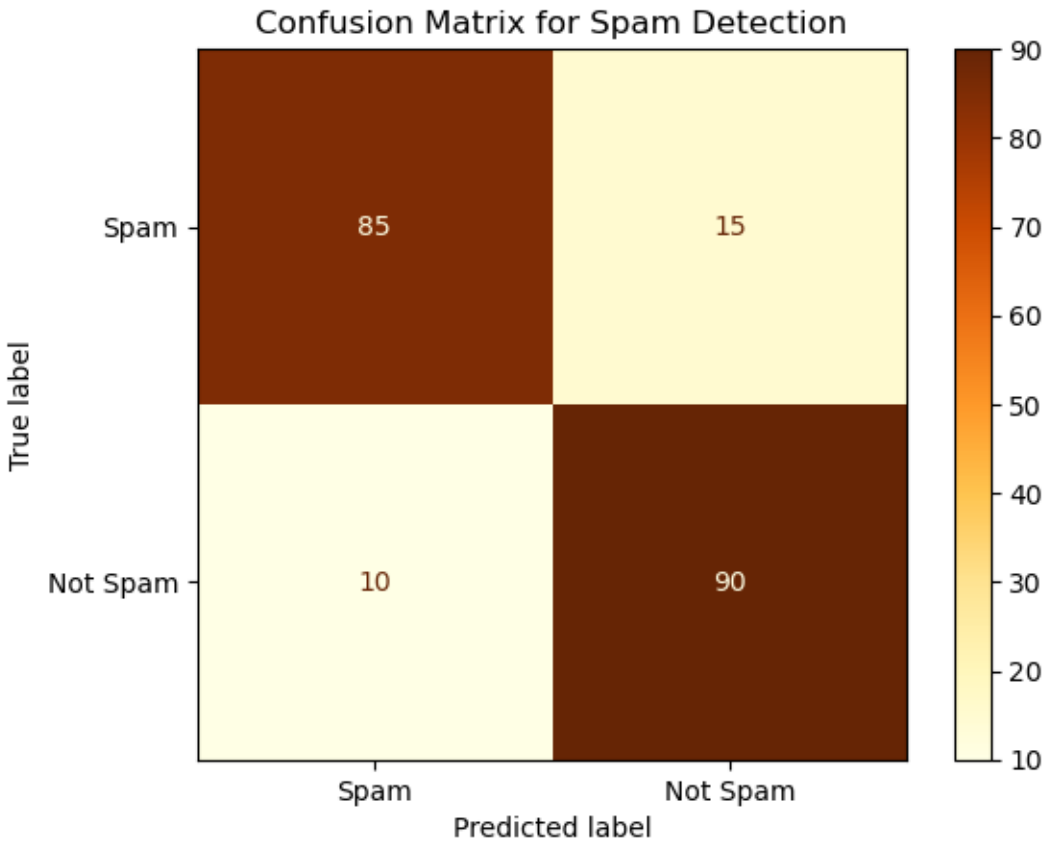
```
# Calculate accuracy, precision, recall, and F1 score
accuracy = (TP + TN) / total
precision = TP / (TP + FP)
recall = TP / (TP + FN)
f1_score = (2 * precision * recall) / (precision + recall)

# Output
print(f"Accuracy: {accuracy}")
print(f"Precision: {precision}")
print(f"Recall: {recall}")
print(f"F1 Score: {f1_score}")
```

```
Accuracy: 0.875
Precision: 0.8947368421052632
Recall: 0.85
F1 Score: 0.8717948717948718
```

b) Construct the confusion matrix based on the provided results.

[139]:
```
# Confusion matrix
confusion_matrix = np.array([[TP, FN], [FP, TN]])
display_matrix = ConfusionMatrixDisplay(confusion_matrix,␣
 ↪display_labels=['Spam', 'Not Spam'])
display_matrix.plot(cmap=plt.cm.YlOrBr)
plt.title('Confusion Matrix for Spam Detection')
plt.show()
```



c) Interpret the performance of the model based on the confusion matrix.

Based on the confusion matrix, the model performs very well, but is by no means perfect. This is supported with our metrics calculated earlier. Overall it labels correctly most of the time and mislabels very infrequently.