Fundamentals of AI & ML

Monsoon Semester V 2021-22

Lab - 2

Date: 1st October 2021

Topic: Bivariate Association

AIM

Consider two data sets given i.e., *Customer Behaviour and House Price Prediction*.

I. Find Bivariate Association between numeric variables using Covariance and Simple Correlation for the given "House Price Prediction" Data set. Represent the results of covariance and correlation into n*n matrices. Where n is the number of numeric variables.

II. Find Bivariate Association between categorical variable "Gender" and numerical variable "Salary" using Point Biserial Correlation for the given Data set i.e., "Customer Behaviour".

THEORY

COVARIANCE FORMULA:

$$cov_{x,y} = \frac{\sum(x_i - \bar{x})(y_i - \bar{y})}{N - 1}$$

$cov_{x,y}$ = covariance between variable x and y

x_i = data value of x

y_i = data value of y

$\bar{x}$ = mean of x

$\bar{y}$ = mean of y

N = number of data values

CORRELATION FORMULA:

$$r = \frac{\sum (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum (x_i - \bar{x})^2 \sum (y_i - \bar{y})^2}}$$

r = correlation coefficient

x_i = values of the x-variable in a sample

$\bar{x}$ = mean of the values of the x-variable

y_i = values of the y-variable in a sample

$\bar{y}$ = mean of the values of the y-variable

FORMULA FOR POINT BISERIAL CORRELATION:

$$r_{pb} = \frac{M_1 - M_0}{s_n}\sqrt{pq}$$

- M_1 = mean (for the entire test) of the group that received the positive binary variable (i.e. the "1").
- M_0 = mean (for the entire test) of the group that received the negative binary variable (i.e. the "0").
- S_n = standard deviation for the entire test.
- p = Proportion of cases in the "0" group.
- q = Proportion of cases in the "1" group.

EXPERIMENT

Program CODE

Question 1:

In [1]:
```
import pandas as pd
```

In [10]:
```
house_pred_df = pd.read_csv("kc_house_data.csv")
print(house_pred_df.head())
```

```
           id     price  sqft_living  floors  zipcode
0  7129300520  221900.0         1180     1.0    98178
1  6414100192  538000.0         2570     2.0    98125
2  5631500400  180000.0          770     1.0    98028
3  2487200875  604000.0         1960     1.0    98136
4  1954400510  510000.0         1680     1.0    98074
```

In [12]:
```
def n_by_n_cov(df):
    def _covariance(x, y):
        x_mean, y_mean = sum(x) / len(x), sum(y) / len(y)
        return sum([(x_i - x_mean) * (y_i - y_mean) for x_i, y_i in zip(x, y)]) / (len(x) - 1)

    columns = df.columns
    covs = dict()
    for column1 in columns:
        covs[column1] = dict()
        for column2 in columns:
            covs[column1][column2] = _covariance(df[column1], df[column2])
    return covs
```

In [13]:
```
house_pred_covs = pd.DataFrame(n_by_n_cov(house_pred_df))
print(house_pred_covs)
```

```
                        id         price   sqft_living        floors  \
id           8.274629e+18 -1.775045e+13 -3.238447e+10  2.877549e+07
price       -1.775045e+13  1.349550e+11  2.368699e+08  5.093897e+04
sqft_living -3.238447e+10  2.368699e+08  8.435337e+05  1.755404e+02
floors       2.877549e+07  5.093897e+04  1.755404e+02  2.915880e-01
zipcode     -1.265812e+09 -1.045060e+06 -9.800232e+03 -1.708121e+00

                 zipcode
id          -1.265812e+09
price       -1.045060e+06
sqft_living -9.800232e+03
floors      -1.708121e+00
zipcode      2.862788e+03
```

In [14]:
```
def n_by_n_corr(df):
    def correlation(x, y):
        x_mean, y_mean = sum(x) / len(x), sum(y) / len(y)
        numerator_summation = sum([(x_i - x_mean) * (y_i - y_mean) for x_i, y_i in zip(x, y)])
        x_denominator_summation = sum([(x_i - x_mean) ** 2 for x_i in x])
        y_denominator_summation = sum([(y_i - y_mean) ** 2 for y_i in y])
        denominator = (x_denominator_summation * y_denominator_summation) ** 0.5
        return round(numerator_summation / denominator, 7)

    columns = df.columns
    corrs = dict()
    for column1 in columns:
        corrs[column1] = dict()
        for column2 in columns:
            corrs[column1][column2] = correlation(df[column1], df[column2])
    return corrs
```

In [15]:
```
house_pred_corr = pd.DataFrame(n_by_n_corr(house_pred_df))
print(house_pred_corr)
```

```
                   id     price  sqft_living    floors   zipcode
id           1.000000 -0.016797    -0.012258  0.018525 -0.008224
price       -0.016797  1.000000     0.702044  0.256786 -0.053168
sqft_living -0.012258  0.702044     1.000000  0.353949 -0.199430
floors       0.018525  0.256786     0.353949  1.000000 -0.059121
zipcode     -0.008224 -0.053168    -0.199430 -0.059121  1.000000
```

In [7]:
```
customer_behavior_df = pd.read_csv("Customer_Behaviour.csv")
print(customer_behavior_df.head())
```

```
    User ID  Gender  Age  Salary  Purchased
0  15624510    Male   19   19000          0
1  15810944    Male   35   20000          0
2  15668575  Female   26   43000          0
3  15603246  Female   27   57000          0
4  15804002    Male   19   76000          0
```

CONCLUSION: The bivariate association between the two variables here "price" and "sqft_living" is Positive, Strong and Linear.

Question 2:

In [8]:
```
def point_biserial_corr(binary_feature, numerical_feature):
    def standard_dev(arr):
        mean = sum(arr) / len(arr)
        summation = sum([(a - mean) ** 2 for a in arr])
        return (summation / len(arr)) ** (1 / 2)

    binary_feature = binary_feature.astype('category').cat.codes  # encoding categorical feature
    if 1 < len(set(binary_feature)) > 2:
        return "BINARY FEATURE IS NOT OF BINARY CATEGORICAL"
    binary_group_split = dict()
    for binary, numerical in zip(binary_feature, numerical_feature):
        if binary not in binary_group_split.keys():
            binary_group_split[binary] = list()
        binary_group_split[binary].append(numerical)
    mean_for_cat_1 = sum(binary_group_split[1]) / len(binary_group_split[1])
    mean_for_cat_0 = sum(binary_group_split[0]) / len(binary_group_split[0])

    numerical_std_dev = standard_dev(numerical_feature)
    proportion_for_cat_1 = len(binary_group_split[1]) / len(binary_feature)
    proportion_for_cat_0 = len(binary_group_split[0]) / len(binary_feature)
    # formula
    category_split_over_deviation = (mean_for_cat_1 - mean_for_cat_0) / numerical_std_dev
    categorical_proportion = (proportion_for_cat_1 * proportion_for_cat_0) ** (1 / 2)
    return category_split_over_deviation * categorical_proportion
```

In [9]:
```
point_biserial_corr = point_biserial_corr(customer_behavior_df['Gender'], customer_behavior_df['Salary'])
print("Point Biserial Correlation:", point_biserial_corr)
```

```
Point Biserial Correlation: -0.06043468529604843
```

CONCLUSION

The point biserial correlation between the Gender and Salary features **is almost perfectly 0**, which indicates **almost no association at all between the 2 aforementioned features**.