Sabancı University Fall 2023

CS210 Fall 2023 Course Project Report

İlayda Saydam

29035

Chapter 1: Project Description

Aim of the project is to analyze the dataset collected from my personal small-business which had been selling jewelry products for the last 3 months by using different data analysis techniques to explore the underlying trends, relationships, and distributions within the dataset. The purpose of this analysis is to understand how efficient and so far successful the business is evolving in its first 3 months.

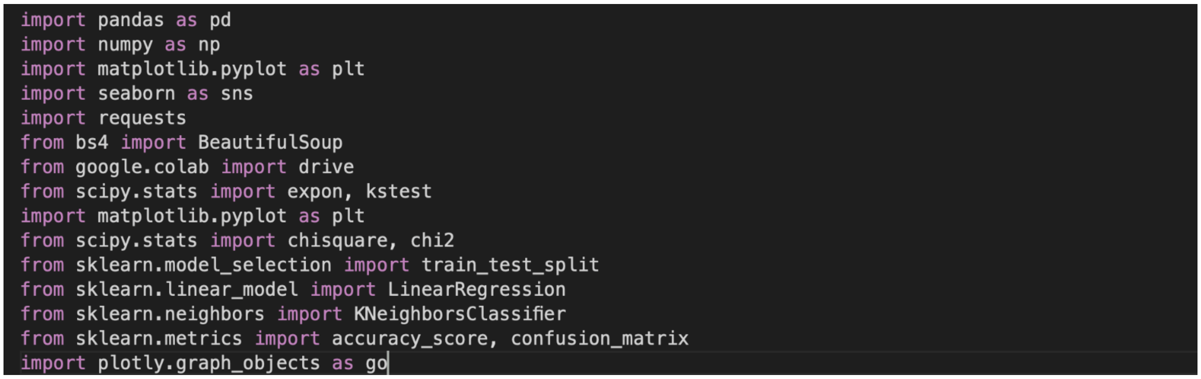
The data set has been filled out by me for 3 months in Excel Sheet, the data is completely my own work from my sales on Shopier. The excel sheet included the number of products bought from wholesalers, the cost of them, the price of sale and how many products had been sold until today. Also the product code is made by me to represent the types of products with the initials. Based on this data set that is converted to CSV file will be imported into Python, the following analysis will be done by using following techniques. The analysis techniques of cleaning dataset, exploratory data analysis, visualization and machine learning will be used in steps in this order.

Chapter 2: Data Analysis

2.1 Exploratory Data Analysis

The Python-based data analysis starts with importing necessary libraries in Figure 1.

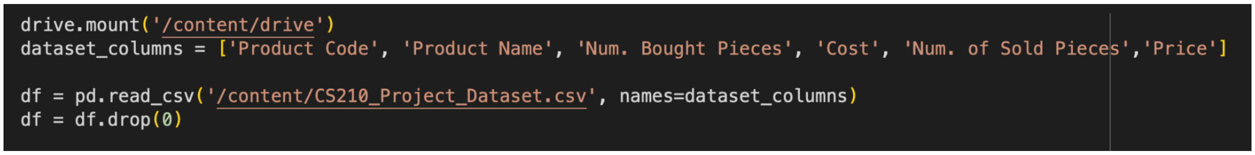
Figure 1. *Imported libraries*



```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import requests
from bs4 import BeautifulSoup
from google.colab import drive
from scipy.stats import expon, kstest
import matplotlib.pyplot as plt
from scipy.stats import chisquare, chi2
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import accuracy_score, confusion_matrix
import plotly.graph_objects as go
```

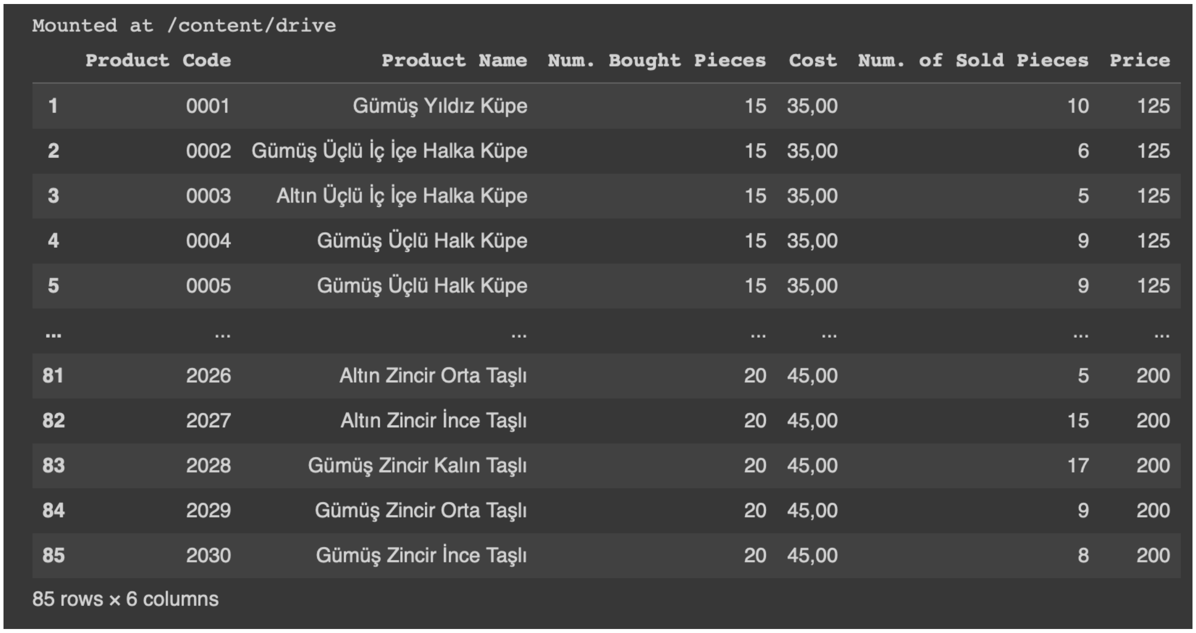
After initializing the required libraries, the Google Drive is mounted using "drive.mount" to import the CSV data file. The dataset columns are explicitly defined as ['Product Code', 'Product Name', 'Num. Bought Pieces', 'Cost', 'Num. of Sold Pieces', 'Price']. The dataset is represented as "df," and the "read_csv" function is employed to import the data into Google Colab. To eliminate any potential confusion arising from the header row in the original dataset, the first row is dropped and the data is printed in Figure 3.

Figure 2. *Reading CSV file and changing the data type*



```
drive.mount('/content/drive')
dataset_columns = ['Product Code', 'Product Name', 'Num. Bought Pieces', 'Cost', 'Num. of Sold Pieces','Price']

df = pd.read_csv('/content/CS210_Project_Dataset.csv', names=dataset_columns)
df = df.drop(0)
```

Figure 3. *Printed data*



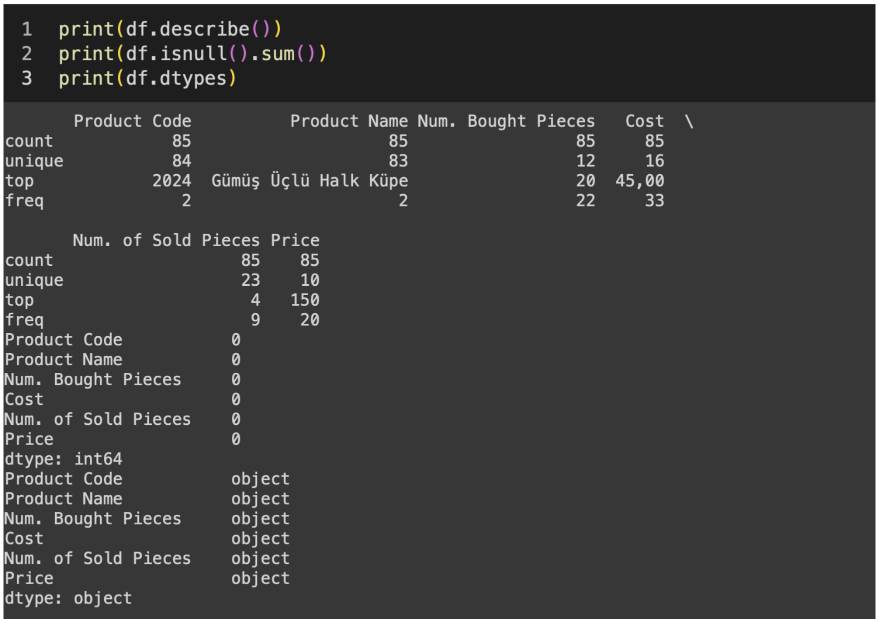
```
Mounted at /content/drive
```

	Product Code	Product Name	Num. Bought Pieces	Cost	Num. of Sold Pieces	Price
1	0001	Gümüş Yıldız Küpe	15	35,00	10	125
2	0002	Gümüş Üçlü İç İçe Halka Küpe	15	35,00	6	125
3	0003	Altın Üçlü İç İçe Halka Küpe	15	35,00	5	125
4	0004	Gümüş Üçlü Halk Küpe	15	35,00	9	125
5	0005	Gümüş Üçlü Halk Küpe	15	35,00	9	125
...	...	...	...	...	...	...
81	2026	Altın Zincir Orta Taşlı	20	45,00	5	200
82	2027	Altın Zincir İnce Taşlı	20	45,00	15	200
83	2028	Gümüş Zincir Kalın Taşlı	20	45,00	17	200
84	2029	Gümüş Zincir Orta Taşlı	20	45,00	9	200
85	2030	Gümüş Zincir İnce Taşlı	20	45,00	8	200

85 rows × 6 columns

After importing data, exploratory data analysis is done for understanding the general structure of the dataset. For these firstly the column numbers are specified as using "df". Then the data summary is obtained by using describe(), isnull(), and dtypes() functions like in Figure 4.

Figure 4. *Exploratory data analysis*



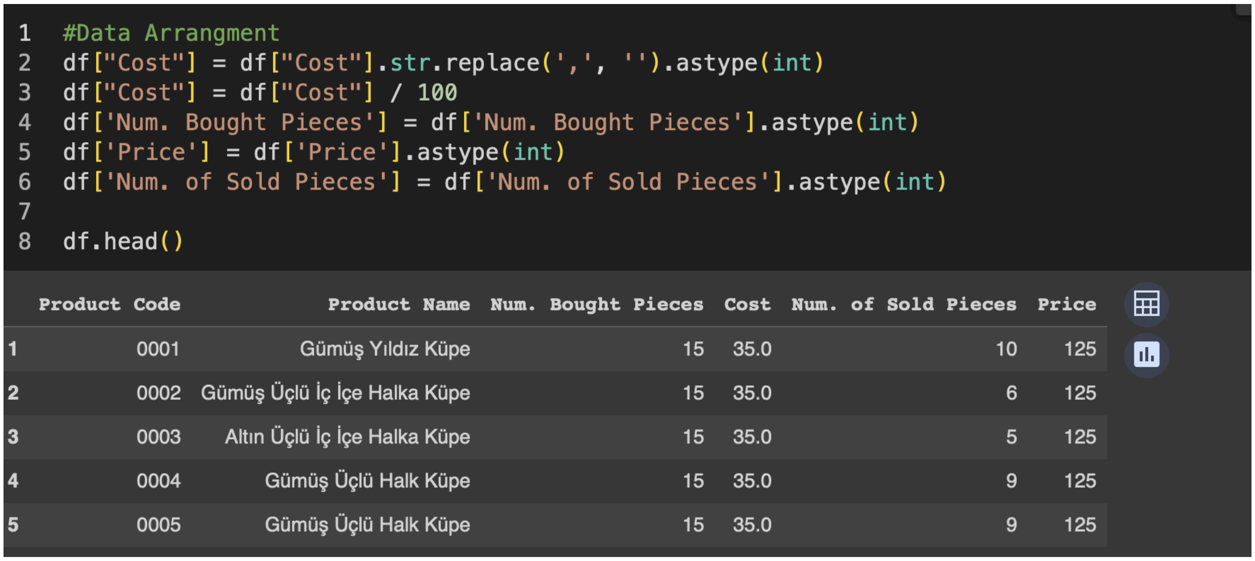
```
print(df.describe())
print(df.isnull().sum())
print(df.dtypes)
```

```
       Product Code          Product Name Num. Bought Pieces   Cost  \
count            85                    85                 85     85   
unique           84                    83                 12     16   
top            2024  Gümüş Üçlü Halk Küpe                 20  45,00   
freq              2                     2                 22     33   

       Num. of Sold Pieces Price  
count                   85    85  
unique                  23    10  
top                      4   150  
freq                     9    20  
Product Code           0
Product Name           0
Num. Bought Pieces     0
Cost                   0
Num. of Sold Pieces    0
Price                  0
dtype: int64
Product Code           object
Product Name           object
Num. Bought Pieces     object
Cost                   object
Num. of Sold Pieces    object
Price                  object
dtype: object
```

2.2 Organizing the Dataset

After importing the csv file to Python, it is realized that the data variables are in object type. Due to the data type, the mathematical expression’s outcome firstly emerged with errors. After getting the precise errors, the type of the data is changed into integer like in Figure 2. Since the “Cost” column is written with a comma, it was not applicable in Python, so it was also separated and changed into integer like in Figure 5.

Figure 5. *Organization of dataset*

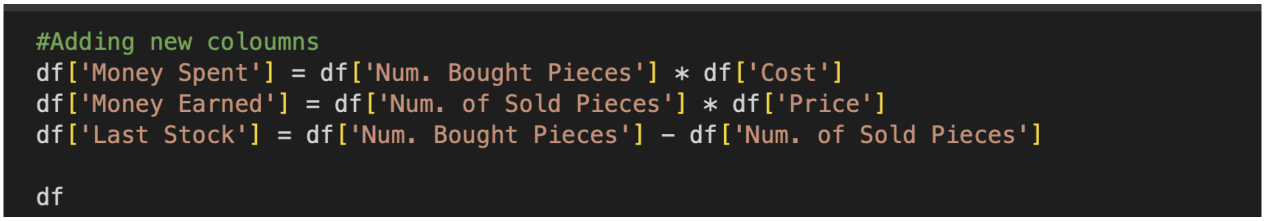


```python
#Data Arrangment
df["Cost"] = df["Cost"].str.replace(',', '').astype(int)
df["Cost"] = df["Cost"] / 100
df['Num. Bought Pieces'] = df['Num. Bought Pieces'].astype(int)
df['Price'] = df['Price'].astype(int)
df['Num. of Sold Pieces'] = df['Num. of Sold Pieces'].astype(int)

df.head()
```

```
  Product Code                 Product Name  Num. Bought Pieces  Cost  Num. of Sold Pieces  Price
1         0001            Gümüş Yıldız Küpe                  15  35.0                   10    125
2         0002   Gümüş Üçlü İç İçe Halka Küpe                  15  35.0                    6    125
3         0003    Altın Üçlü İç İçe Halka Küpe                  15  35.0                    5    125
4         0004           Gümüş Üçlü Halk Küpe                  15  35.0                    9    125
5         0005           Gümüş Üçlü Halk Küpe                  15  35.0                    9    125
```

After the changes, some important information that was planned to use in the further steps was not represented in the dataset. The necessary columns and data “Money Spent”, “Money Earned” and “Last Stock” are calculated like in Figure 6.

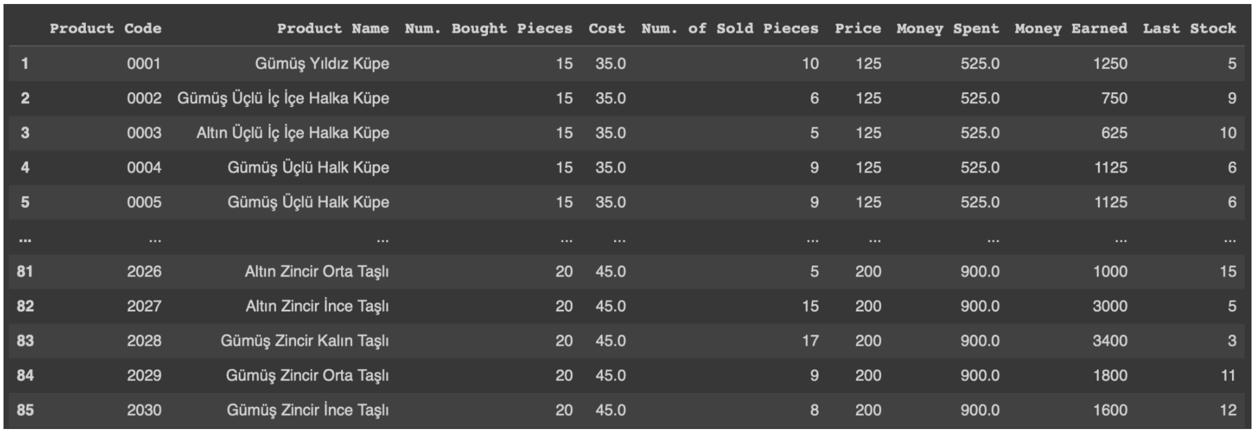
Figure 6. *Adding new columns*



```python
#Adding new coloumns
df['Money Spent'] = df['Num. Bought Pieces'] * df['Cost']
df['Money Earned'] = df['Num. of Sold Pieces'] * df['Price']
df['Last Stock'] = df['Num. Bought Pieces'] - df['Num. of Sold Pieces']

df
```

After the essential computations, the data set was printed in Figure 7.

Figure 7. *Last version of dataset*



	Product Code	Product Name	Num. Bought Pieces	Cost	Num. of Sold Pieces	Price	Money Spent	Money Earned	Last Stock
1	0001	Gümüş Yıldız Küpe	15	35.0	10	125	525.0	1250	5
2	0002	Gümüş Üçlü İç İçe Halka Küpe	15	35.0	6	125	525.0	750	9
3	0003	Altın Üçlü İç İçe Halka Küpe	15	35.0	5	125	525.0	625	10
4	0004	Gümüş Üçlü Halk Küpe	15	35.0	9	125	525.0	1125	6
5	0005	Gümüş Üçlü Halk Küpe	15	35.0	9	125	525.0	1125	6
...	...	...	...	...	...	...	...	...	...
81	2026	Altın Zincir Orta Taşlı	20	45.0	5	200	900.0	1000	15
82	2027	Altın Zincir İnce Taşlı	20	45.0	15	200	900.0	3000	5
83	2028	Gümüş Zincir Kalın Taşlı	20	45.0	17	200	900.0	3400	3
84	2029	Gümüş Zincir Orta Taşlı	20	45.0	9	200	900.0	1800	11
85	2030	Gümüş Zincir İnce Taşlı	20	45.0	8	200	900.0	1600	12

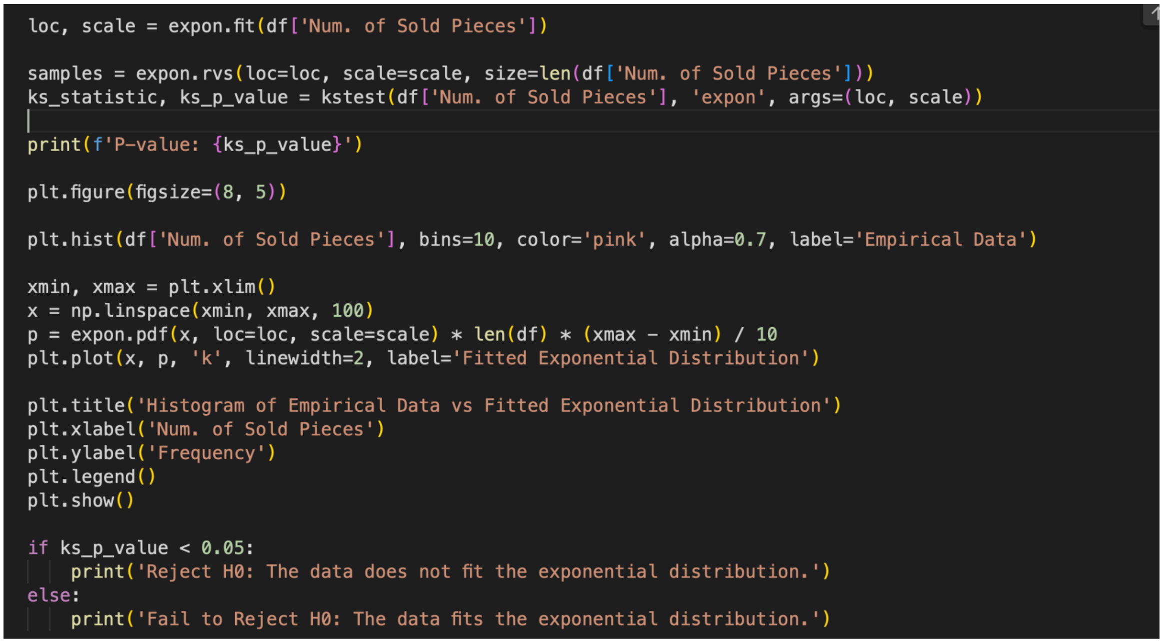
2.3 Data Visualization

The first data to visualize is the histogram of the number of products sold until that time. When the dataset was piloted with histogram data, the dataset seems to look like Exponential Distribution. Based on the observation of histogram, the null and alternative hypothesis was created:

Ho: The data follows exponential distribution
H1: The data does not follow exponential distribution

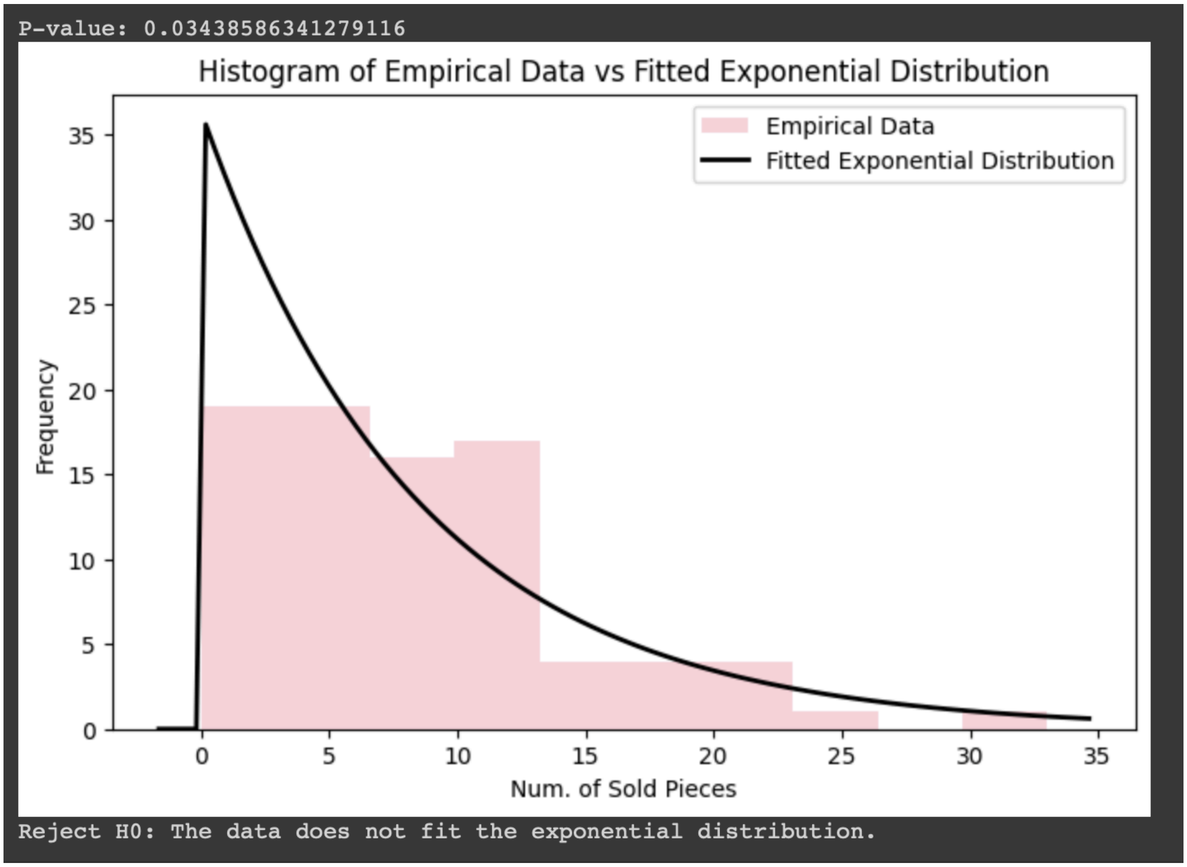
For the hypothesis testing the P-value is calculated, the histogram is drawn and to understand the relationship between data and exponential distribution the trendline of exponential distribution is added. Since P-value is calculated as 0.05 and it is smaller than 0.05, the null hypothesis is rejected. The code is shown in Figure 8, and the histogram graph shown in Figure 9.

Figure 8. *Hypothesis testing code*



```
loc, scale = expon.fit(df['Num. of Sold Pieces'])

samples = expon.rvs(loc=loc, scale=scale, size=len(df['Num. of Sold Pieces']))
ks_statistic, ks_p_value = kstest(df['Num. of Sold Pieces'], 'expon', args=(loc, scale))

print(f'P-value: {ks_p_value}')

plt.figure(figsize=(8, 5))

plt.hist(df['Num. of Sold Pieces'], bins=10, color='pink', alpha=0.7, label='Empirical Data')

xmin, xmax = plt.xlim()
x = np.linspace(xmin, xmax, 100)
p = expon.pdf(x, loc=loc, scale=scale) * len(df) * (xmax - xmin) / 10
plt.plot(x, p, 'k', linewidth=2, label='Fitted Exponential Distribution')

plt.title('Histogram of Empirical Data vs Fitted Exponential Distribution')
plt.xlabel('Num. of Sold Pieces')
plt.ylabel('Frequency')
plt.legend()
plt.show()

if ks_p_value < 0.05:
    print('Reject H0: The data does not fit the exponential distribution.')
else:
    print('Fail to Reject H0: The data fits the exponential distribution.')
```

Figure 9. *Histogram of number of sold pieces*



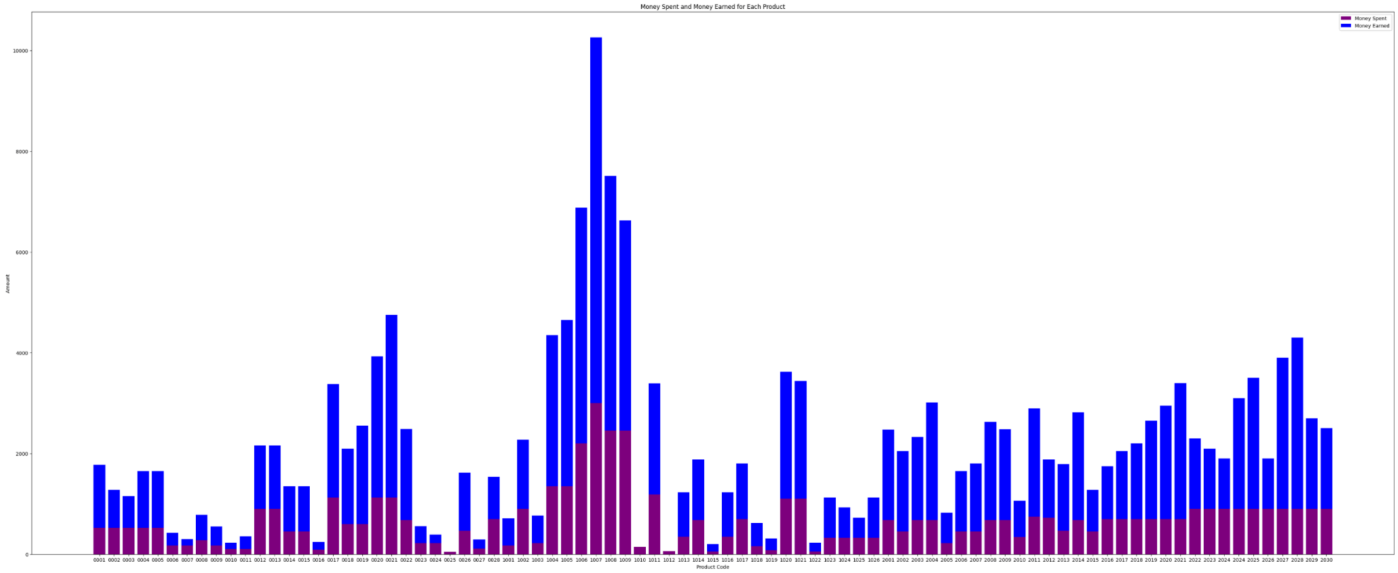
```
P-value: 0.03438586341279116
```

```
Reject H0: The data does not fit the exponential distribution.
```

The first analysis that helps to understand about the sales and revenue is if there is any relationship between exponential distribution and sales histogram. It is observed that there is a sales distribution in the range of (0, 10) that shows us most of the products sales between this range.

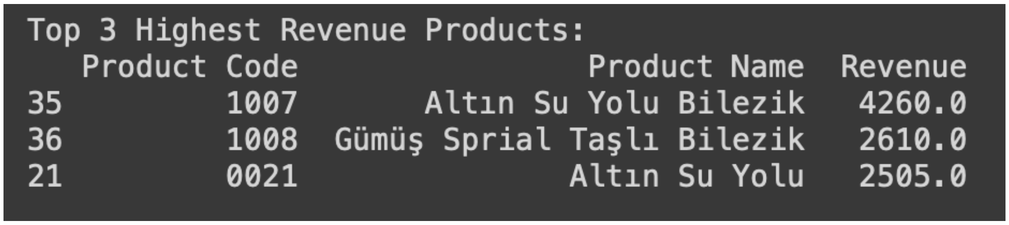
Another visualization that is aimed to represent revenue. After the tests of different pilot types, the most suitable one was found as a bar chart. In a single bar graph, money spent on products represented in purple and money earned represented in blue for each product in Figure 10.

Figure 10. *Money earned and money spent graph*



Respecting the received information in that bar chart, the products with highest values are found as follows in Figure 11.

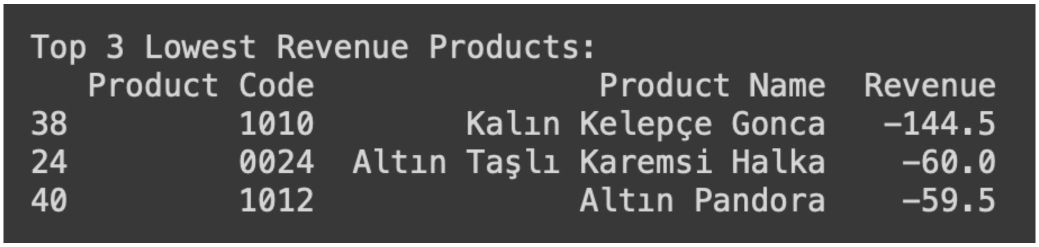
Figure 11. *Highest revenue product*



```
Top 3 Highest Revenue Products:
   Product Code                Product Name  Revenue
35         1007       Altın Su Yolu Bilezik   4260.0
36         1008  Gümüş Sprial Taşlı Bilezik   2610.0
21         0021               Altın Su Yolu   2505.0
```

Other than the products with highest value, some products' revenue are smaller than 0, these are shown in Figure 12.

Figure 12. *Lowest revenue products*



```
Top 3 Lowest Revenue Products:
   Product Code                Product Name  Revenue
38         1010         Kalın Kelepçe Gonca   -144.5
24         0024    Altın Taşlı Karemsi Halka    -60.0
40         1012               Altın Pandora    -59.5
```

Based on the lowest and highest revenue products, their stock graphs are also piloted to see how many stock we have and whether the restock is necessary. For the highest revenue, the last stocks are higher in Figure 13. And for the lowest revenue, the last stocks are lower which shows that they are not restocking:

Figure 13. *Stocks of highest revenue products*

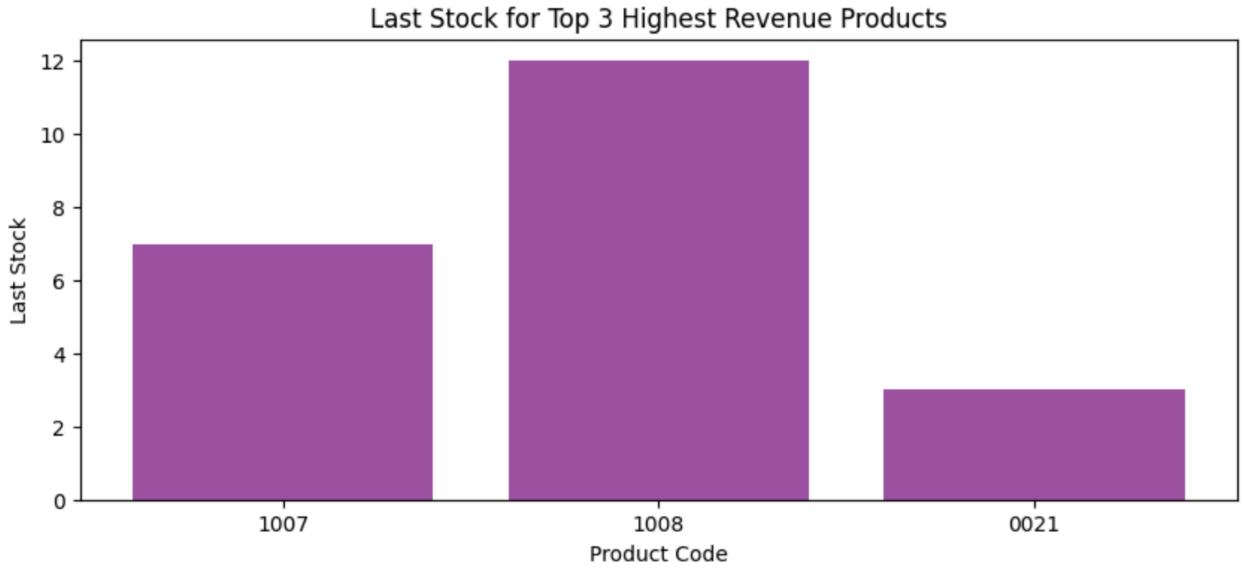
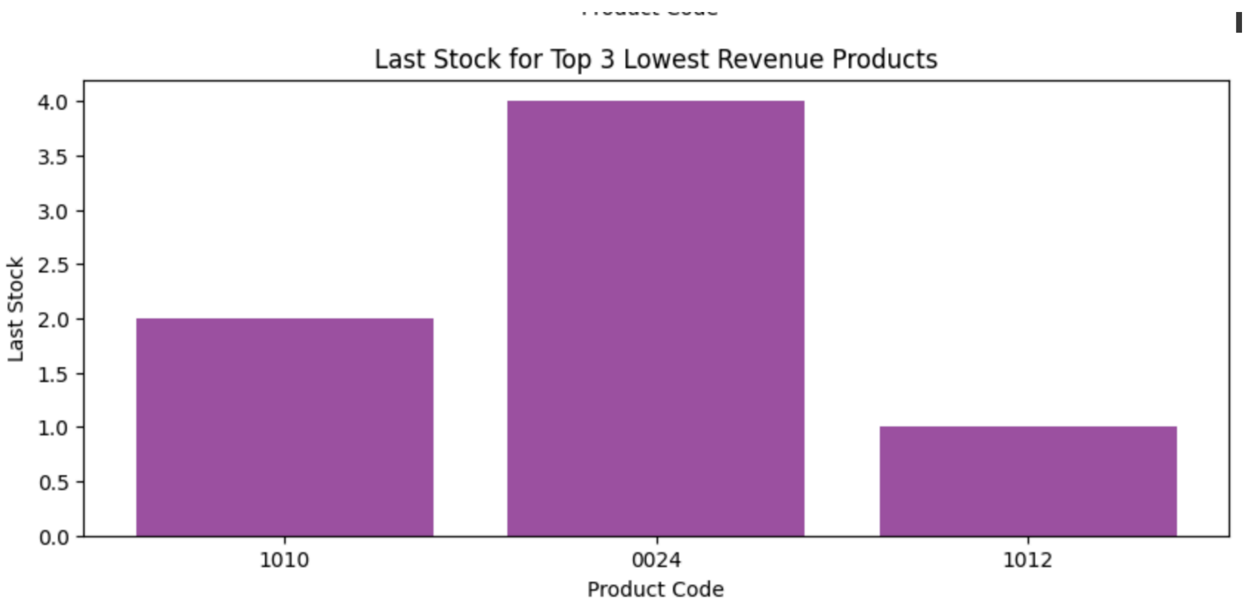
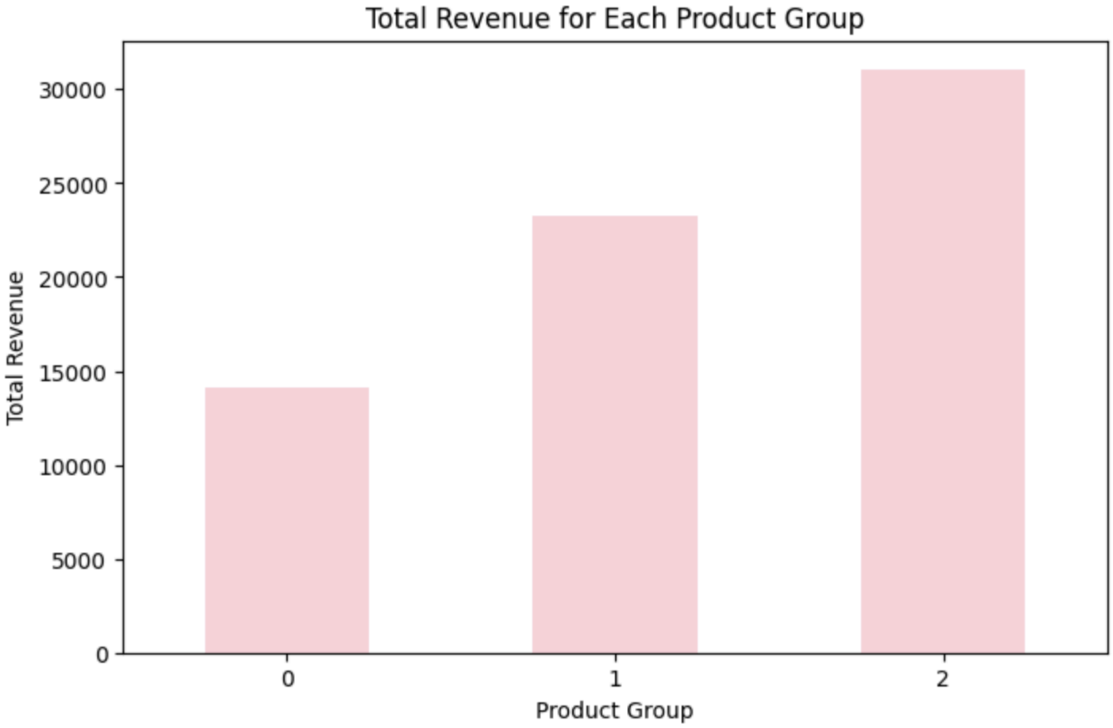


Figure 14. *Stocks of lowest revenue products*



The other analysis that is useful in the project was which product type has the highest revenue. In my data, the type of products are separated based on their initial number: 0 for earrings, 1 for bracelet and 2 for necklace. In the analysis, the products are grouped according to their initials and the revenue of each group is summed. The outcome was in Figure 15. Based on the outcome, necklaces have the highest revenue and earrings are the lowest. Also total revenue was found as 68,434.25 TL.

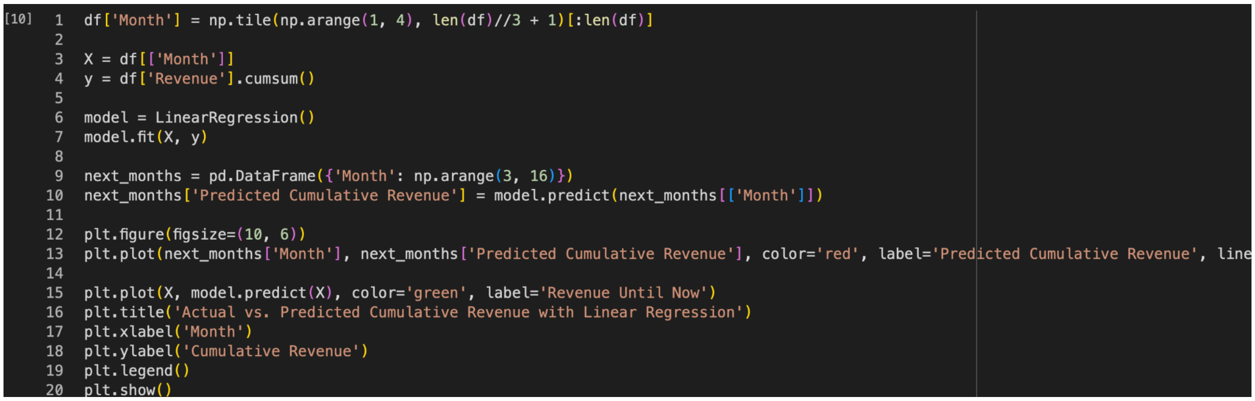
Figure 15. *Bar chart of revenue of each product group*



2.4 Machine Learning

For analyzing the data, there is also machine learning technique used. The aim is to use machine learning to train the revenue dataset to make assumptions about future revenues. Since the revenue for each month is not certain, the months are assigned to the dataset. Then, X is defined as an independent variable for Months and Y is defined as a dependent variable of total revenue. The trained dataset is used to show a linear regression graph of revenue. The code is shown in Figure 16.

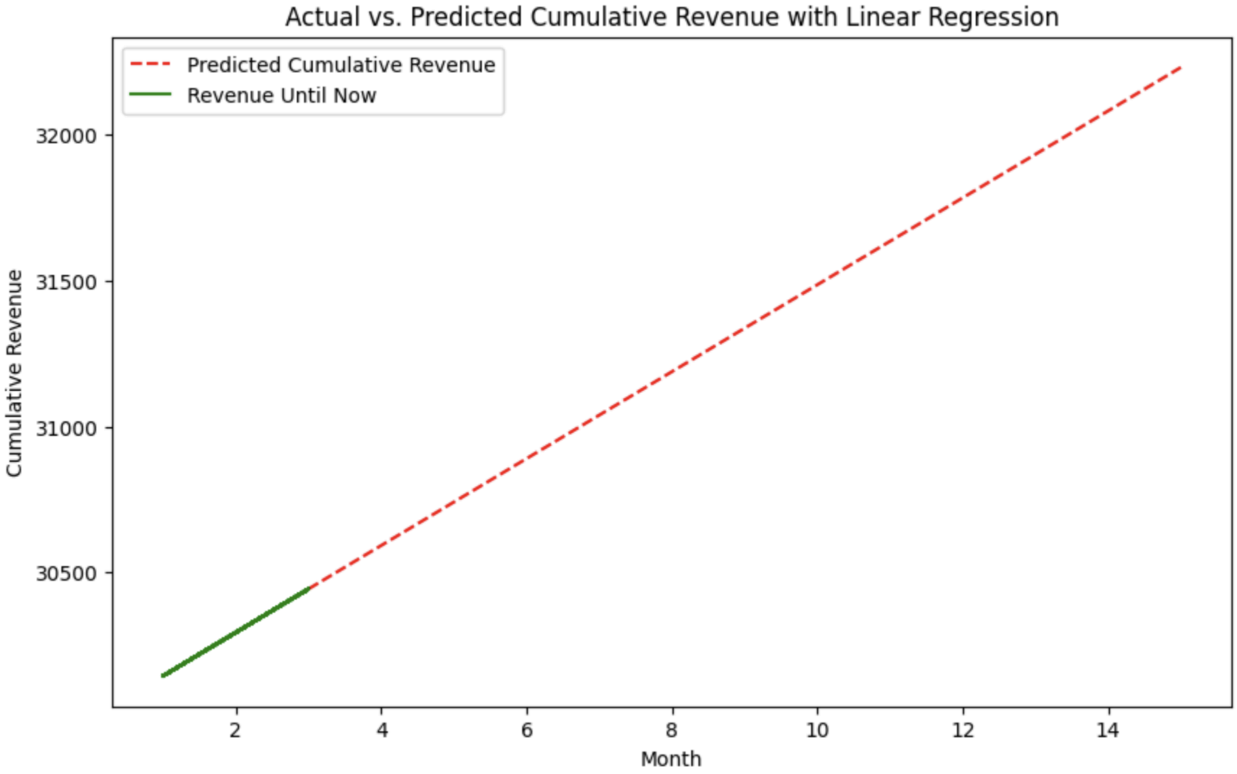
Figure 16. *Code for machine learning*



```
df['Month'] = np.tile(np.arange(1, 4), len(df)//3 + 1)[:len(df)]

X = df[['Month']]
y = df['Revenue'].cumsum()

model = LinearRegression()
model.fit(X, y)

next_months = pd.DataFrame({'Month': np.arange(3, 16)})
next_months['Predicted Cumulative Revenue'] = model.predict(next_months[['Month']])

plt.figure(figsize=(10, 6))
plt.plot(next_months['Month'], next_months['Predicted Cumulative Revenue'], color='red', label='Predicted Cumulative Revenue', line

plt.plot(X, model.predict(X), color='green', label='Revenue Until Now')
plt.title('Actual vs. Predicted Cumulative Revenue with Linear Regression')
plt.xlabel('Month')
plt.ylabel('Cumulative Revenue')
plt.legend()
plt.show()
```

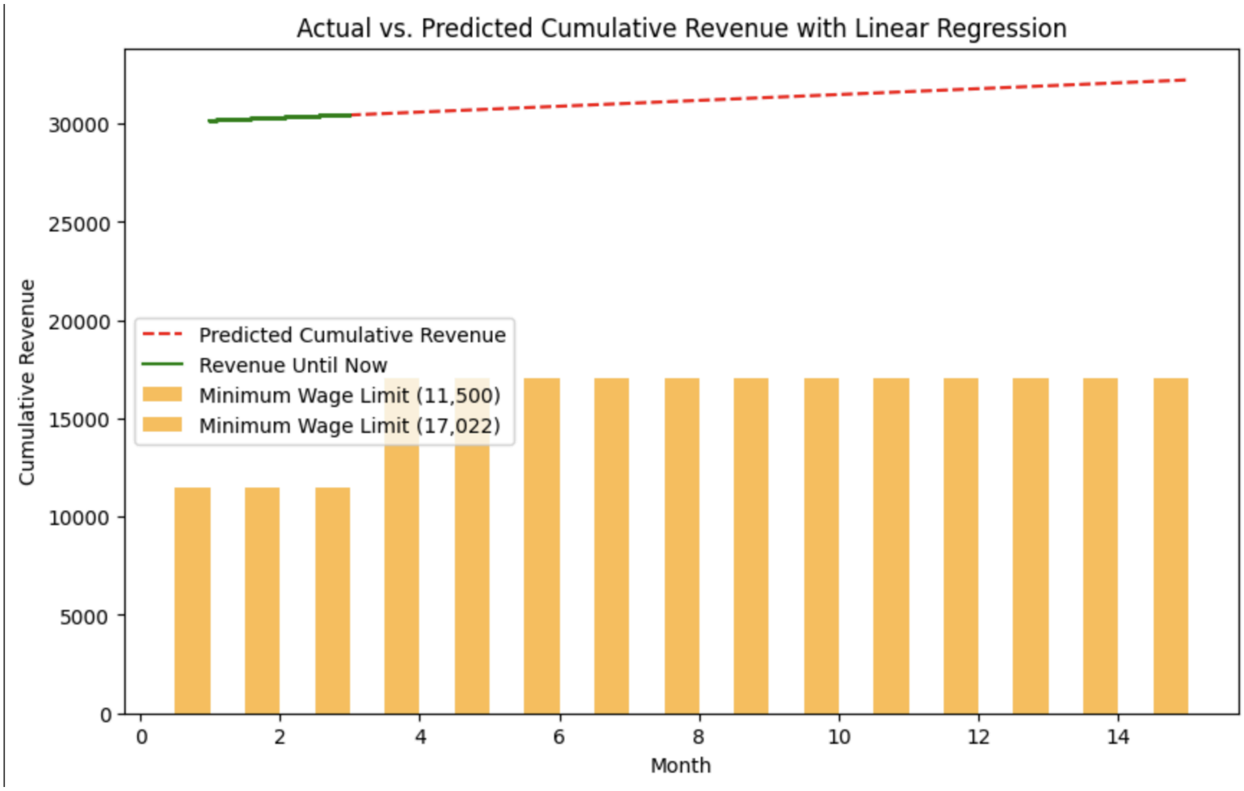
For the first 3 months, a linear regression model was represented. Based on this predicted model, the following 1 year of revenue was forecasted again using linear regression like in Figure 17. Since the distribution of each year’s revenue is not known, the machine learning technique also calculated and forecasted based on these predictions.

Figure 17. *Linear regression model of predicted revenue*



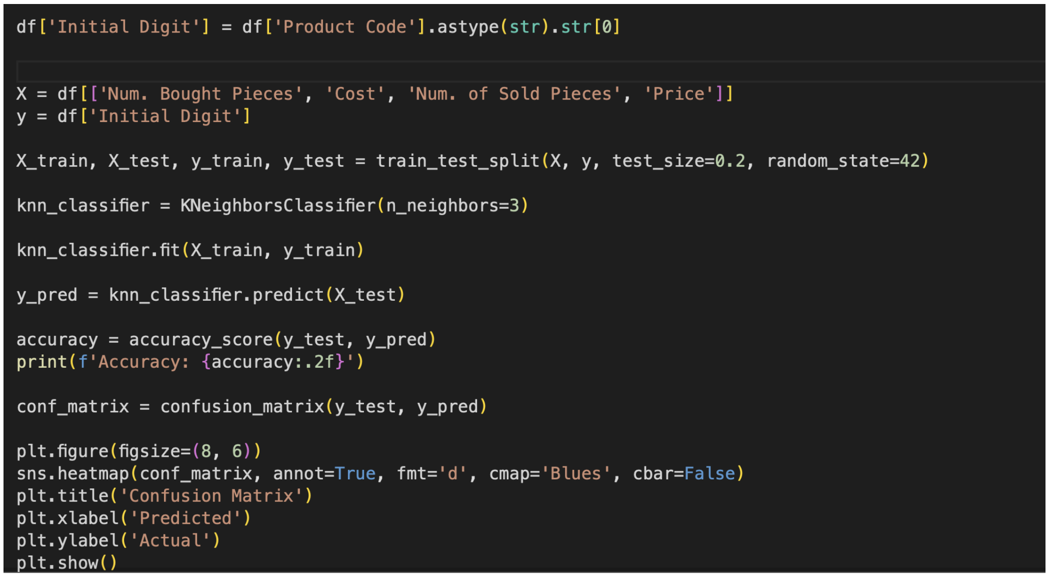
In the revenue graph, the minimum wage limit is also added to understand whether my small-business is making more or less profit than the minimum wage limit of Turkey. Since the minimum wage is increased to the 11,500 TL to 17,002 TL for the new year, last 3 months revenue is compared to the 11,500 TL and forecasted revenue with 17,002 TL

Figure 18. *Predicted revenue compared to minimum wage*



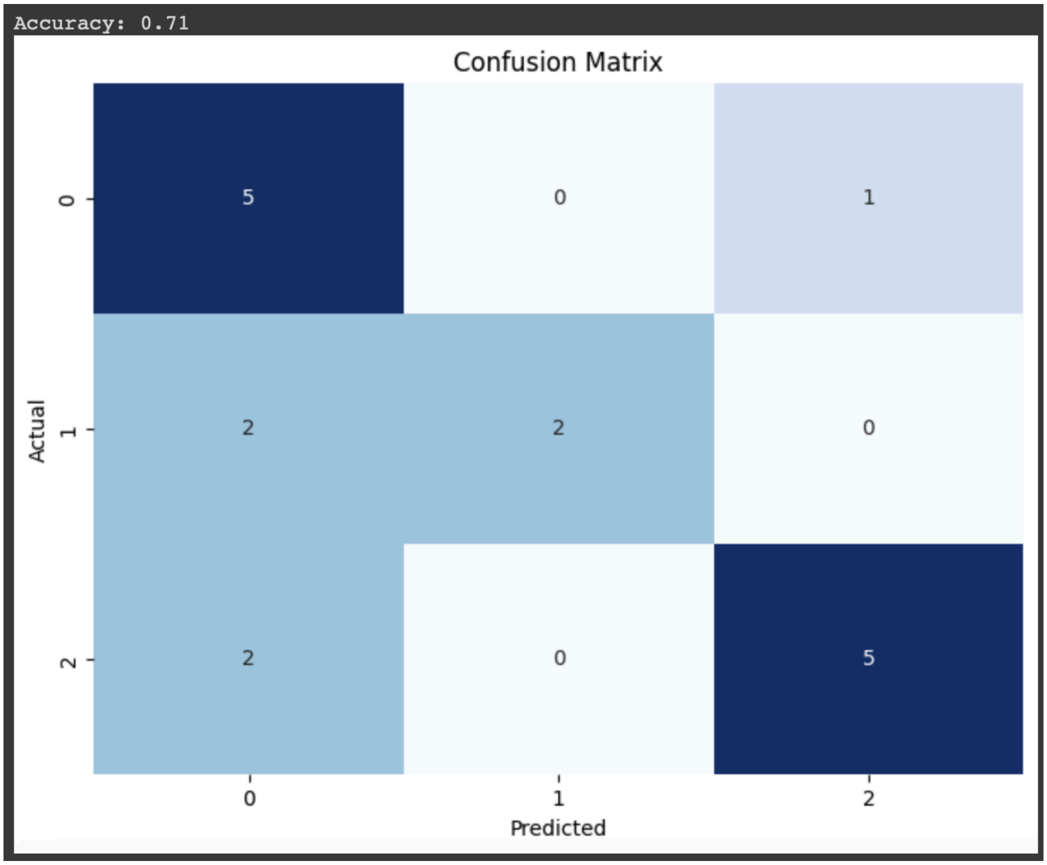
Another information that can be archived using machine learning was K-Classification. With the code in Figure 19, the classification was done for the product's type. In Figure 20, also accuracy is calculated and confession matrix was obtained. Plotting the map, we obtained which product type has a higher contribution in the number of sales.

Figure 19. *Code for Classification*



```
df['Initial Digit'] = df['Product Code'].astype(str).str[0]


X = df[['Num. Bought Pieces', 'Cost', 'Num. of Sold Pieces', 'Price']]
y = df['Initial Digit']

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

knn_classifier = KNeighborsClassifier(n_neighbors=3)

knn_classifier.fit(X_train, y_train)

y_pred = knn_classifier.predict(X_test)

accuracy = accuracy_score(y_test, y_pred)
print(f'Accuracy: {accuracy:.2f}')

conf_matrix = confusion_matrix(y_test, y_pred)

plt.figure(figsize=(8, 6))
sns.heatmap(conf_matrix, annot=True, fmt='d', cmap='Blues', cbar=False)
plt.title('Confusion Matrix')
plt.xlabel('Predicted')
plt.ylabel('Actual')
plt.show()
```

Figure 20. *Confusion matrix*



```
Accuracy: 0.71
```

Chapter 3: Conclusion

In summary, we went through important steps like bringing in data, making it understandable for the computer, and creating helpful charts. Looking at the data, we noticed a pattern in how products are sold. The money we made and spent on each product was visualized in simple bar charts, helping us see which products are doing well. We also used computer predictions to guess how much money we might make in the future. Comparing our earnings to the minimum wage helped us understand if our business is doing better or worse than a basic income. This analysis gives us a good picture of how well our business is doing and helps us plan for the future.

The first histogram analysis shows us the distribution of sales and in which range the sales are distributed. It is observed that most of the products sold are (0, 10), so the sells can be improved to pass this range. Another analysis is done to understand which products have higher revenue rate. For the purpose of this, the bar chart has been piloted, and the highest revenue rates are calculated. some products are making negative revenue, and it is also concluded they will not be restocked again. Then the revenue of product groups are calculated and it is seen that the highest revenue product group is found as necklaces. Another analysis technique to use was a machine learning method to predict the following 1 year plan. The dependent and independent variables are setted and the linear regression model has been drawn. Then the prediction regarding the revenue compared to the minimum wage.

As a conclusion, the small-business is making a sufficient profit which is more than the minimum wage in Turkey. Also based on the prediction, the future revenue will increase a lot. The base limitations about the project was some insufficient written data. In the future, the dataset can be collected more detailed, for example details like how much product sold in a certain period can help to predict the future revenue better. Or updating the dataset in python currently helps to observe which products are sold more with more profit.