BAHRIA UNIVERSITY,
(Karachi Campus)

Department of Software Engineering

PROJECT REPORT

Course Title: Computer Communication & Networking **Course Code**: CSL-495
Course Instructor: Dr. Hina Shakir **Class**: BSE-5(B)
Lab Instructor: Farheen Faisal Siddiqui

PROJECT TITLE:

"Grocery Price Comparison & Trend Analysis for Karachi"

S.NO	Enrollment	Name
1	02-131232-017	SYED MUGHEES ALI
2	02-131232-051	SAFWAN AHMED ISHAQ
3	02-131232-059	HAMZA KHALIQ

Teacher Signature: _____ **Remarks:** _____

Contents

PROJECT OVERVIEW

This project focuses on analyzing grocery price variations across multiple online retail stores operating in Karachi. Grocery prices frequently vary between stores and change over time, making it difficult for consumers to identify cost-effective purchasing options. The project applies fundamental data science techniques to collect, clean, analyze, and visualize grocery price data in order to extract meaningful insights.

A structured dataset was created using snapshot prices of commonly purchased grocery items collected from publicly available online grocery platforms such as Imtiaz, Carrefour, Metro, Naheed, and Chase Up. Using Python and Pandas, the dataset was cleaned and transformed to ensure consistency and comparability across stores and time periods. Exploratory Data Analysis (EDA) was performed to compare prices across stores and categories, followed by a simple machine learning model using Linear Regression to observe short-term price trends.

The project demonstrates the complete data science workflow, including data preprocessing, exploratory analysis, modeling, evaluation, and interpretation, while remaining within the scope of techniques covered in the lab sessions.

PROBLEM STATEMENT & OBJECTIVE

Grocery prices in Karachi differ significantly across retail stores and fluctuate over time due to market conditions, supply chain factors, and store-level pricing strategies. Consumers generally lack transparent and consolidated information to compare prices across stores, which makes informed purchasing decisions difficult. Additionally, short-term price trends are not easily observable without systematic data analysis.

Objectives

The main objectives of this project are:

- To collect and organize grocery price data from multiple online stores in Karachi.
- To clean and preprocess the data for consistency and fair comparison.
- To perform Exploratory Data Analysis (EDA) to compare prices across stores and categories.
- To apply a simple Linear Regression model to observe short-term price trends of selected grocery items.
- To interpret analytical results and provide practical insights based on the findings.

SYSTEM DESIGN

The system follows a clear and logical data pipeline design consisting of the following stages:

1. **Data Collection**
   Grocery price data was manually collected from publicly available online grocery websites. The data represents snapshot prices observed over a limited time period.
2. **Data Storage**
   The collected data was stored in a structured CSV file containing attributes such as date, store name, category, item name, quantity, price, and normalized price per kilogram.
3. **Data Preprocessing & Cleaning**
   The dataset was cleaned using Python and Pandas by:

   - Converting date fields into datetime format
   - Ensuring numeric consistency in price-related columns
   - Removing duplicate records
   - Normalizing prices to price-per-kilogram for fair comparison

4. **Exploratory Data Analysis (EDA)**
   EDA techniques such as grouping, aggregation, and visualization were used to analyze store-wise, category-wise, and time-based price patterns.
5. **Model Development & Evaluation**
   A Linear Regression model was developed to analyze short-term price trends. Model

performance was evaluated using regression-appropriate metrics such as Mean Squared Error (MSE) and R-squared values.

6. **Visualization & Interpretation**
   Graphs and plots were generated to visually communicate insights and model behavior.

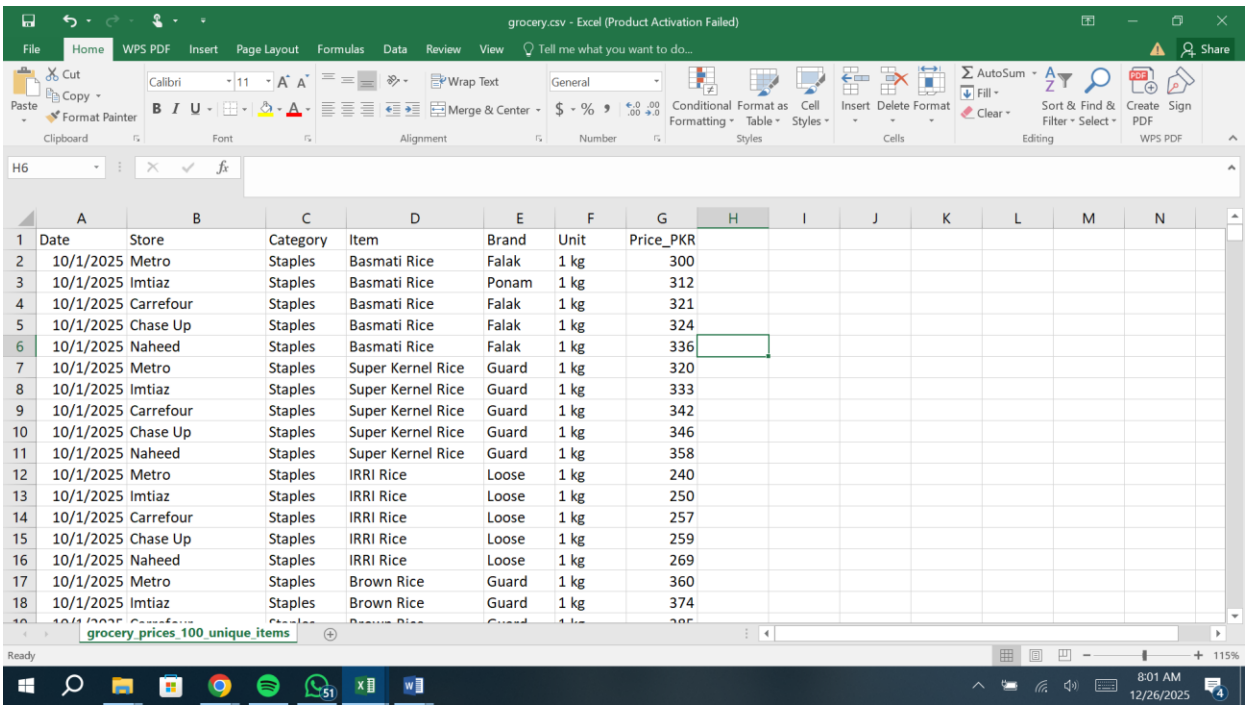
CODE EXPLANATION

The project code is implemented in Python using Google Colab. Key libraries include Pandas for data handling, NumPy for numerical operations, Matplotlib for visualization, and Scikit-learn for machine learning.

- The dataset is loaded from a CSV file using `pd.read_csv()`.
- Data preprocessing includes date conversion, numeric type enforcement, duplicate removal, and validation checks.
- Exploratory analysis is performed using Pandas `groupby()` operations to calculate average prices and statistical summaries.
- Visualizations such as bar charts, line plots, and scatter plots are created using Matplotlib to illustrate price comparisons and trends.
- A Linear Regression model is trained using date values converted into numeric format.
- Model evaluation is performed using Mean Squared Error (MSE) and R-squared metrics.
- Validation checks are included to ensure data correctness and robustness.

Each section of the code is modular and follows a logical execution flow, making it easy to understand and reproduce.

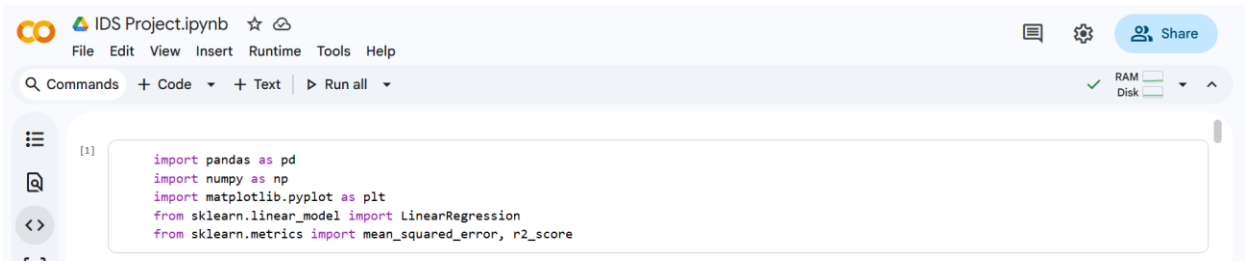
TESTING AND RESULTS (SCREENSHOTS / SAMPLE INPUT & OUTPUT)



grocery.csv - Excel (Product Activation Failed)

	A	B	C	D	E	F	G
1	Date	Store	Category	Item	Brand	Unit	Price_PKR
2	10/1/2025	Metro	Staples	Basmati Rice	Falak	1 kg	300
3	10/1/2025	Imtiaz	Staples	Basmati Rice	Ponam	1 kg	312
4	10/1/2025	Carrefour	Staples	Basmati Rice	Falak	1 kg	321
5	10/1/2025	Chase Up	Staples	Basmati Rice	Falak	1 kg	324
6	10/1/2025	Naheed	Staples	Basmati Rice	Falak	1 kg	336
7	10/1/2025	Metro	Staples	Super Kernel Rice	Guard	1 kg	320
8	10/1/2025	Imtiaz	Staples	Super Kernel Rice	Guard	1 kg	333
9	10/1/2025	Carrefour	Staples	Super Kernel Rice	Guard	1 kg	342
10	10/1/2025	Chase Up	Staples	Super Kernel Rice	Guard	1 kg	346
11	10/1/2025	Naheed	Staples	Super Kernel Rice	Guard	1 kg	358
12	10/1/2025	Metro	Staples	IRRI Rice	Loose	1 kg	240
13	10/1/2025	Imtiaz	Staples	IRRI Rice	Loose	1 kg	250
14	10/1/2025	Carrefour	Staples	IRRI Rice	Loose	1 kg	257
15	10/1/2025	Chase Up	Staples	IRRI Rice	Loose	1 kg	259
16	10/1/2025	Naheed	Staples	IRRI Rice	Loose	1 kg	269
17	10/1/2025	Metro	Staples	Brown Rice	Guard	1 kg	360
18	10/1/2025	Imtiaz	Staples	Brown Rice	Guard	1 kg	374

Dataset of 1100+ ROWS



IDS Project.ipynb

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, r2_score
```

[2]
```
df = pd.read_csv("grocery.csv")
```

[4]
```
df.head()
df.describe()
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1125 entries, 0 to 1124
Data columns (total 7 columns):
 #   Column     Non-Null Count  Dtype 
---  ------     --------------  ----- 
 0   Date       1125 non-null   object
 1   Store      1125 non-null   object
 2   Category   1125 non-null   object
 3   Item       1125 non-null   object
 4   Brand      1125 non-null   object
 5   Unit       1125 non-null   object
 6   Price_PKR  1125 non-null   int64 
dtypes: int64(1), object(6)
memory usage: 61.7+ KB
```

```
import re
def convert_to_grams(unit):
    unit_lower = unit.lower().strip()

    unit_patterns = {
        'kg': 1000,
        'ml': 1,
        'g': 1,
        'l': 1000,
        'm': 1
    }

    # Corrected regex to extract numerical part
    match = re.search(r'(\d+\.?\d*)', unit_lower)
    if not match:
        return np.nan

    try:
        value = float(match.group(1))
    except ValueError:
        return np.nan

    if 'kg' in unit_lower:
        return value * unit_patterns['kg']
    elif 'ml' in unit_lower:
        return value * unit_patterns['ml']
    elif 'g' in unit_lower:
        return value * unit_patterns['g']
    elif 'l' in unit_lower:
        return value * unit_patterns['l']
    elif 'm' in unit_lower:
        return value * unit_patterns['m']
    else:
        return np.nan

df['quantity_grams'] = df['Unit'].apply(convert_to_grams)

display(df.head())
```

	Date	Store	Category	Item	Brand	Unit	Price_PKR	quantity_grams
0	10/1/2025	Metro	Staples	Basmati Rice	Falak	1 kg	300	1000.0
1	10/1/2025	Imtiaz	Staples	Basmati Rice	Ponam	1 kg	312	1000.0
2	10/1/2025	Carrefour	Staples	Basmati Rice	Falak	1 kg	321	1000.0
3	10/1/2025	Chase Up	Staples	Basmati Rice	Falak	1 kg	324	1000.0
4	10/1/2025	Naheed	Staples	Basmati Rice	Falak	1 kg	336	1000.0

[14]
```
df['price_per_kg'] = (df['Price_PKR'] / df['quantity_grams']) * 1000
```

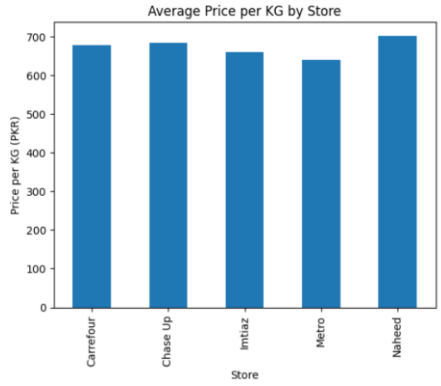
[18]
```
normalized_view = df.sort_values(by='price_per_kg')
display(normalized_view)
```

	Date	Store	Category	Item	Brand	Unit	Price_PKR	quantity_grams	price_per_kg
210	10/1/2025	Metro	Beverages	Mineral Water	Nestle	1.5 L	90	1500.0	60.000000
585	11/1/2025	Metro	Beverages	Mineral Water	Nestle	1.5 L	92	1500.0	61.333333
211	10/1/2025	Imtiaz	Beverages	Mineral Water	Nestle	1.5 L	92	1500.0	61.333333
212	10/1/2025	Carrefour	Beverages	Mineral Water	Nestle	1.5 L	93	1500.0	62.000000
213	10/1/2025	Chase Up	Beverages	Mineral Water	Nestle	1.5 L	93	1500.0	62.000000
...	...	...	...	...	...	...	...	...	...
239	10/1/2025	Naheed	Packaged	Coffee	Nescafe	200 g	1012	200.0	5060.000000
987	12/1/2025	Carrefour	Packaged	Coffee	Nescafe	200 g	1014	200.0	5070.000000
988	12/1/2025	Chase Up	Packaged	Coffee	Nescafe	200 g	1024	200.0	5120.000000
614	11/1/2025	Naheed	Packaged	Coffee	Nescafe	200 g	1032	200.0	5160.000000
989	12/1/2025	Naheed	Packaged	Coffee	Nescafe	200 g	1052	200.0	5260.000000

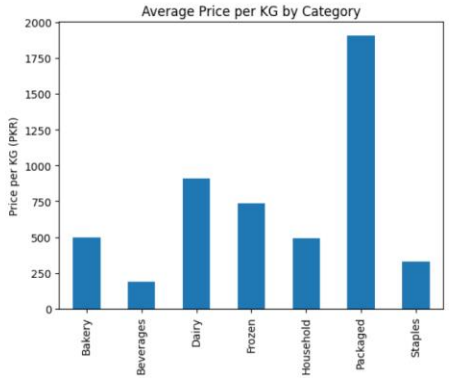
975 rows × 9 columns

Next steps: Generate code with normalized_view | New interactive sheet

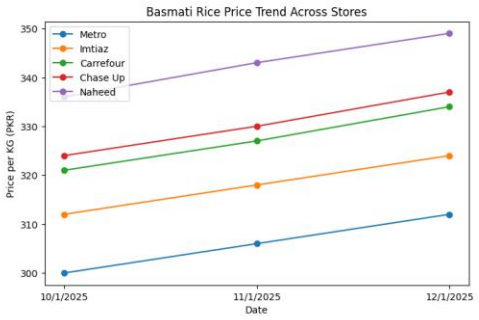
```
store_avg = df.groupby('Store')['price_per_kg'].mean()
store_avg.plot(kind='bar', title='Average Price per KG by Store')
plt.ylabel("Price per KG (PKR)")
plt.xlabel("Store")
plt.show()
```



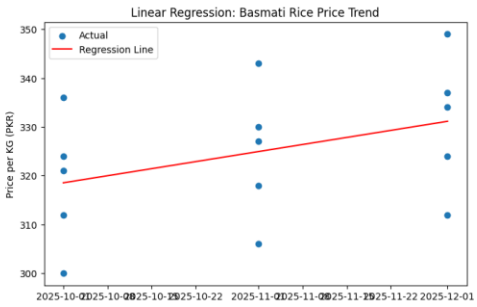

```
category_avg = df.groupby('Category')['price_per_kg'].mean()

category_avg.plot(kind='bar', title='Average Price per KG by Category')
plt.ylabel("Price per KG (PKR)")
plt.xlabel("Category")
plt.show()
```




```
item_df = df[df['Item'] == 'Basmati Rice']

plt.figure(figsize=(8,5))
for store in item_df['Store'].unique():
    s_data = item_df[item_df['Store'] == store]
    plt.plot(s_data['Date'], s_data['price_per_kg'], marker='o', label=store)

plt.title("Basmati Rice Price Trend Across Stores")
plt.xlabel("Date")
plt.ylabel("Price per KG (PKR)")
plt.legend()
plt.show()
```



```
[32] item_df['predicted_price'] = model.predict(X)

     plt.figure(figsize=(8,5))
     plt.scatter(item_df['Date'], item_df['price_per_kg'], label='Actual')
     plt.plot(item_df['Date'], item_df['predicted_price'], color='red', label='Regression Line')

     plt.title("Linear Regression: Basmati Rice Price Trend")
     plt.xlabel("Date")
     plt.ylabel("Price per KG (PKR)")
     plt.legend()
     item_df = item_df.sort_values('Date')

     plt.show()
```




```
[34] from sklearn.metrics import mean_squared_error, r2_score

     y_pred = model.predict(X)

     mse = mean_squared_error(y, y_pred)
     r2 = r2_score(y, y_pred)

     print("MSE:", round(mse, 2))
     print("R² Score:", round(r2, 4))

     MSE: 151.26
     R² Score: 0.1489
```

```
[35] assert df['price_per_kg'].min() > 0
     assert df['price_per_kg'].isnull().sum() == 0
     assert df['Date'].isnull().sum() == 0

     print("All validation checks passed successfully.")

     All validation checks passed successfully.
```

Testing was conducted through multiple validation checks and result inspections to ensure correctness and reliability of the analysis.

- **Data Validation Tests:**
  Assertions were used to verify that no missing or negative price values existed in the dataset.
- **Exploratory Analysis Results:**
  Store-wise and category-wise analysis revealed noticeable price differences across retailers. Some stores consistently offered lower prices for staple items, while others were relatively expensive.
- **Model Evaluation Results:**
  The Linear Regression model showed a positive slope for selected items, indicating a gradual increase in prices over the observed period. The R-squared value indicated that time explained a portion of the price variation, which is expected given the limited dataset.

Screenshots of code execution, data summaries, and visualizations were captured from Google Colab and included in the report to demonstrate correct functionality and outputs.

CONCLUSION & REFLECTION ON LEARNING OUTCOMES

This project successfully demonstrated the application of core data science concepts learned during the lab sessions. By working with real-world grocery price data, the project highlighted the importance of data preprocessing, normalization, and exploratory analysis before applying any machine learning techniques.

The analysis showed that grocery prices vary significantly across stores and categories, and simple trend analysis can provide useful short-term insights. The Linear Regression model, while basic, effectively demonstrated how machine learning can be used to observe patterns in data without overcomplicating the problem.

Through this project, practical skills were developed in data cleaning, visualization, model evaluation, and result interpretation. The project reinforced the importance of methodological correctness, realistic assumptions, and clear communication of findings — essential skills for future data science applications.