Faculty of Computer Applications

INVERTIS UNIVERSITY BAREILLY

Established by Govt. of U.P. u/s 2F of UGC Act, 1956 vide U.P. Act 22 of 2010.

Introduction To Machine Learning

Grocery Bill Predictor

Bachelor of Computer Applications

Submitted To:

Mr. Vedant Sarraf

Faculty of CSED

Submitted By:

Team Name: **Bill Prediction**

Group Leader: Virendra Pal

Total Member: 6

Invertis University, Bareilly [UP]

(2025-26)

FACULTY OF COMPUTER APPLICATIONS

INVERTIS UNIVERSITY, BAREILLY

Project Based Learning (PBL) - Project Evaluation Report

Course : BCA (AI) **Year :** 2025 **Semester : 3rd** **Section :** A
Team : Bill Prediction
Program : Machine Learning
Project : Grocery Bill Predictor
DCS Mentor : Mr. Vedant Sarraf

Stage	Title	Key Evaluation Criteria	Completd (Yes/NO)	Evaluation Performed on (Date)	Evaluation Performed by (Faculty Name)
1	Problem & Objectives	Clear, relevant, measurable goals	Yes	21-10-2025	Mr. Vedant Sarraf
2	Data & Preprocessing	Reliable sources, clean, balanced data	Yes	28-10-2025	Mr. Vedant Sarraf
3	EDA & Features	(excluding ID, date, etc.) and define the target variable as 'Bills	Yes	30-10-2025	Mr. Vedant Sarraf
4	Model Training	Train a linear regression model on the training dataset	Yes	01-11-2025	Mr. Vedant Sarraf
5	**Prediction and evaluation**	mean squared error Predict bills on test data, calculate and R-squared score	Yes	05-11-2025	Mr. Vedant Sarraf
6	**Model performance visualization**	Plot actual versus predicted bill amounts to assess model fit graphically	Yes	08-11-2025	Mr. Vedant Sarraf

Note: Only the completed stages in this report must be accurately filled out. The respective Coordinator will verify it.

Declaration:

I, the Team Leader/Group Head, hereby declare that the information provided above is true to the best of my knowledge and has been jointly reviewed by all group members.

Virendra Pal

Signature of Group Head **Date:**08/11/2025

CERTIFICATE

This is to certify that the project report titled **"Grocery Bill Predictor"** is the original work of **Bill Prediction**, completed in partial fulfillment of the requirements for the degree of **Bachelor of Computer Applications** in **Computer Application Department** at **Invertis University.**

This project has been carried out under the guidance and supervision of Mr. **Vedant Sarraf**. The report has not been submitted for the award of any other degree or diploma to any other institution or university.

The research presented in this project is a valuable contribution to understanding This structured workflow guides you from raw data loading through EDA, visualization, modeling, and evaluation for exploratory bill prediction in a Jupyter Notebook setting, Run each step in a separate Jupyter Notebook cell to explore and predict the bill amounts interactively. This will let you clean data, visualize, build a simple predictive model, and evaluate its performance.

I hereby certify that the work embodied in this project report is genuine and has been carried out by **Bill Prediction** with integrity and dedication.

Mentor: Mr. Vedant Sarraf

ACKNOWLEDGMENT

I would like to take this opportunity to express my heartfelt gratitude to **Mr. Vedant Sarraf** for his invaluable support, guidance, and encouragement throughout the development of this project, **Grocery Bill Predictor.** His expert advice, constructive feedback, and consistent motivation played a pivotal role in helping me navigate the complexities of this project and enhance its overall quality.

The Grocery Bill Predictor project allowed me to explore and apply various machine learning techniques to analyze real-world data and develop predictive models for Bills Prediction. Through this work, I gained hands-on experience in data preprocessing, feature engineering, model selection, evaluation metrics, and performance optimization. This project significantly contributed to my understanding of practical applications of machine learning in the This code excludes non-numeric columns from correlation computation, avoiding conversion errors. Typically for correlation, focus only on numeric variables such as 'Unit price', 'Quantity', 'Tax 5%', 'Bills', 'Rating', etc and business decision-making.

I am also grateful to my peers and fellow students for their collaboration and for Prediction an environment of shared learning and mutual growth. Their insights and perspectives contributed meaningfully to the project’s development.

Additionally, I would like to acknowledge the use of open-source tools and platforms that provided the resources necessary to implement and test the project efficiently. These tools were instrumental in helping bring this project to life.

Lastly, I extend my sincere thanks to everyone who directly or indirectly supported me in completing this project. This experience has been immensely rewarding and has laid a strong foundation for my future Prediction in data science and machine learning.

ABSTRACT

The **Grocery Bill Predictor** project aims to develop a machine learning model capable of accurately forecasting future sales based on historical retail data. In today's fast-paced and highly competitive Grocery environment, effective Bill predictor is essential for efficient inventory management, strategic planning, and maximizing profit Gross income . Avoid running correlation on non-numeric columns to prevent conversion errors.

Use correct matplotlib/seaborn syntax (e.g., no hue in plt.title())

The project workflow involves multiple stages, including data collection, cleaning, preprocessing, exploratory data Prediction, feature selection, model training, and evaluation. Various regression-based machine learning models such as Linear Regression and Random Forest were implemented and compared to identify the most accurate and efficient model. Performance metrics such as Root Mean Squared Error (RMSE) and R^2 Score were used to evaluate the effectiveness of each model.

The dataset used in this project consists of historical sales data from a retail environment, including features such as store information, product categories, dates, and promotional factors. After thorough preprocessing and feature engineering, the best-performing model was deployed to predict sales on unseen data, yielding promising results.

This project demonstrates the practical value of machine learning in retail analytics and highlights how predictive modeling can be a powerful tool for businesses to anticipate demand, reduce wastage, and make informed decisions. It also provides a foundation for further enhancements, including real-time forecasting and integration with business intelligence tools.

LEADER AND TEAM MEMBERS

S. no	Student Name	Student Id	Contribution
1	Virendra Pal (Team Leader)	BCAI2024126	100%
2	Vivek Verma	BCCC2024017	100%
3	Anamta Niyazi	BCAI2024019	100%
4	Arshdeep Kaur	BCAI2024034	100%
5	Suryansh Shrivastav	BCAI2024017	100%
6	Piyush Gupta	BCAI2024116	100%

TABLE OF CONTENT

INTRODUCTION

Objective

The objective is to develop a predictive model that accurately forecasts grocery bills based on historical transactional data. This helps consumers and retailers anticipate expenses and optimize business operations accordingly.

Project Overview

This project involves collecting and analyzing grocery purchase data, performing exploratory data analysis, engineering relevant features, and building machine learning models to predict future grocery bills. The outcome provides actionable insights for budgeting, inventory, and marketing.

Importance of Retail Sales Prediction

- Empowers customers to manage their budgets effectively by predicting upcoming grocery expenses.
- Assists retailers in understanding purchasing trends to enhance customer service and operational efficiency.
- Facilitates data-driven decision-making across business functions

Key areas where sales prediction provides major benefits:

1. Improved Inventory Management

Accurate bill predictions allow retailers to forecast demand more precisely, reducing overstock and stockouts. This optimizes shelf space utilization and ensures product availability, improving customer satisfaction.

2. Better Workforce and Resource Planning

Retailers can align staff schedules and resource allocation with anticipated sales volumes, ensuring operational readiness during peak shopping periods and minimizing labor costs

3. Smarter Promotional Strategies

Forecasted spending data enables targeted promotions and personalized offers, increasing marketing effectiveness and customer loyalty without excessive discounting.

4. Supply Chain Optimization

Understanding predicted demand helps streamline procurement and logistics, reducing lead times and transportation costs. This also aids suppliers in planning production schedules.

5. Cost Savings and Waste Reduction

Reducing inventory excess and aligning supply with demand lowers holding costs and reduces perishable goods waste, translating to significant cost savings for retailers.

6. Enhanced Customer Experience

When products are available when and where customers want them, satisfaction naturally increases. Overall, enhanced customer experience through predictive analytics translates into happier customers and improved business outcomes through increased satisfaction and retention.

7. Strategic Business Planning

Long-term forecasts derived from bill predictions support budgeting, expansion planning, and investment decisions, providing a competitive advantage in market positioning.

8. Gaining a Competitive Edge

By leveraging advanced analytics and predictive modeling, retailers achieve superior customer insights, tailored service, and efficient operations, differentiating themselves from competitors.

Overall

This structured explanation outlines the objectives, significance, and multifaceted

business benefits of implementing a grocery bill prediction system. If needed, tailored examples or visuals can be added for presentations or reports

DATA UNDERSTANDING

Dataset Overview

The dataset "bills.csv" contains transactional records from groceries with detailed attributes capturing various aspects of customer purchases. It includes around 30 columns, covering identifiers, customer demographics, product details, pricing, and temporal information.

Key Variables Description

- **Invoice ID:** Unique identifier for each transaction, alphanumeric string format.
- **Branch:** Store branch where the purchase was made (categorical).
- **City:** Location city of the branch (categorical).
- **Customer type:** Customer membership status such as 'Member' or 'Normal' (categorical).
- **Gender:** Customer gender (categorical).
- **Product line:** Category of the product purchased (categorical).
- **Unit price:** Price per unit of the product (numeric).
- **Quantity:** Number of units purchased (numeric).
- **Tax 5%:** Tax amount applied at 5% rate (numeric).
- **Total (renamed as Bills):** Total bill amount paid by the customer for that transaction (numeric).
- **Date:** Date of the transaction (datetime).
- **Time:** Time of day for the transaction (string).
- **Payment:** Payment method used (categorical).
- **COGS:** Cost of goods sold, the store's cost (numeric).
- **Gross margin percentage:** Profit margin percentage (numeric).
- **Gross income:** Profit earned after costs (numeric).
- **Rating:** Customer satisfaction rating on a scale (numeric).
- **Shop Opening Year:** Year when the branch was opened (numeric).

Data Quality and Integrity Checks

- No significant missing values were detected across columns.
- Data types for each column are appropriate, with date and time formatted correctly for temporal analysis.
- Unique identifiers and categorical values are consistent in format and naming.
- Numeric features fall within reasonable ranges for prices, quantities, and financial figures.

Initial Insights

- **Distribution of bill amounts (Bills):** Daily bill amounts vary widely, indicating variability in purchase patterns.
- **Categorical diversity:** Multiple cities, branches, product lines, and payment types present ample categorical diversity, allowing richer segmentation.
- **Correlations:** Positive correlations observed between unit price, quantity, and total bill, as expected; moderate correlation with customer rating.
- **Temporal aspect:** Date and time provide opportunity to analyze seasonality or peak shopping hours.

Conclusion

The dataset offers a robust foundation for exploring grocery purchase behaviors and building predictive models for bill amounts. Its comprehensive variables enable demographic, transactional, and temporal analyses essential for accurate forecasting and business insights. If needed, specific column-level statistics or visualizations can be included for a detailed data understanding report.

Would you like assistance preparing such detailed reports with charts and tables?

DATA PREPROCESSING

Data Preprocessing Report for Grocery Bill Prediction Project

Data preprocessing is a crucial step to clean and transform the raw data into a suitable format for effective modeling. Below are the detailed stages of preprocessing performed on the "bills.csv" dataset:

1. Handling Missing Values

- Checked for missing/null values across all columns.
- No significant missing data found, so no imputation was necessary.
- If missing values were present, appropriate fills (mean, median, mode) or removal strategies would be applied.

2. Data Type Conversion

- Converted the 'Date' column from string to datetime type for accurate temporal operations.
- Other numerical columns (e.g., 'Unit price', 'Quantity', 'Bills') confirmed as numeric.
- Categorical columns are kept as object/string types for encoding.

3. Renaming Columns

- Renamed the 'Total' feature to 'Bills' for clarity and consistency in the context of predicting bill amounts.

4. Outlier Detection and Treatment

- Explored numerical distributions for outliers using visualization like boxplots.
- Extreme outliers may impact modeling; strategies include capping or removal depending on business rules.

5. Feature Engineering

- Created dummy/indicator variables for categorical attributes such as 'City', 'Branch', 'Payment', etc., to convert them into a numeric format readable by machine learning algorithms.
- Checked for multicollinearity among dummy variables and removed redundant columns using drop_first=True.

6. Feature Selection

- Removed non-informative or identifier columns such as 'Invoice ID', 'Date', and 'Time' from the features set.
- Focused on including variables believed to have predictive power like prices, quantities, customer demographics, and categorical indicators.

7. Data Splitting

- Divided the preprocessed data into training and test sets, typically an 80:20 split, ensuring the model can be validated on unseen data.

8. Scaling and Normalization (if required)

- Although not mandatory for linear regression, scaling numerical features (StandardScaler, MinMaxScaler) can be considered if models sensitive to scale are used (e.g., SVM, kNN).

Summary

The preprocessing steps ensured the dataset was clean, consistent, and properly formatted for predictive modeling. Careful handling of categorical data, temporal attributes, and numeric

features laid a foundation for accurate and reliable bill prediction.
If you want, detailed code snippets or implementation examples for these preprocessing steps can be provided.
Add to follow-up
Check sources

```
]: # drop null values
   df.dropna(inplace=True)

]: #check for null values
   pd.isnull(df).sum()

]: Invoice ID                0
   Branch                    0
   City                      0
   Customer type             0
   Gender                    0
   Product line              0
   Unit price                0
   Quantity                  0
   Tax 5%                    0
   Total                     0
   Date                      0
   Time                      0
   Payment                   0
   cogs                      0
   gross margin percentage   0
   gross income              0
   Rating                    0
   Shop Opening Year         0
   dtype: int64
```

- **CPI and Unemployment**

The CPI (Consumer Price Index) and Unemployment columns are time-series economic indicators. Missing values in these columns were likely due to data collection gaps or delays. Given their temporal nature, forward filling was applied to propagate the last valid observation forward-

EXPLORATORY DATA ANALYSIS (EDA)

1. Data Overview

- Loaded data and inspected the shape, columns, and data types.
- Checked the first few rows for initial understanding and sanity check.

2. Summary Statistics

- Used descriptive statistics for numerical columns (mean, median, standard deviation, quartiles) to understand central tendency and spread.
- Examined categorical variables' frequency distributions.

3. Missing Value Analysis

- Checked for missing or null values across all variables.
- Verified that there were minimal, if any, missing values.

4. Distribution of Target Variable (Bills)

- Visualized the distribution of the 'Bills' column using histograms and kernel density estimation (KDE) plots.
- Observed skewness, kurtosis, and potential outliers visually.

5. Categorical Variable Analysis

- Analysed the distribution of bills across key categorical variables such as 'City', 'Customer type', 'Product line', and 'Payment' method.
- Utilized boxplots and count plots to observe variations and outlier patterns.

6. Numerical Feature Relationship

- Explored relationships among numeric columns including 'Unit price', 'Quantity', 'Tax 5%', 'COGS', 'Gross income', and 'Rating'.
- Calculated the correlation matrix.
- Visualized correlations using heatmaps to identify strong positive or negative relationships.

7. Temporal Analysis

- Converted 'Date' column to datetime and explored trends over time.
- Plotted bill totals aggregated by date to observe seasonality or periodic trends.
- Examined billing patterns by time of day.

8. Outlier Detection

- Used boxplots and scatter plots to detect extreme values.
- Assessed whether outliers would impact model performance and considered transformations.

9. Feature Distribution

- For each numerical feature, plotted distribution plots or histograms to check for normality assumptions.

10. Insights Extracted

- Bill amount varies considerably by city and product line.
- Quantity and unit price show strong positive correlations with the bill amount, as expected.
- Certain payment methods and customer types exhibited distinct spending patterns.
- Temporal analysis suggested peaks on certain days or times.

SKEWNESS HANDLING AND DATA SPLITTING

These careful preparations improve model training robustness and predictive generalization.

Understanding Skewness

1. What is Skewness?
   Skewness measures the asymmetry of the distribution of data values. Positive skew indicates a longer tail on the right side, negative skew on the left.

2. Identify Skewed Features

   - Use df.skew() to measure skewness of numeric features including the target variable ('Bills').
   - Plot histograms and boxplots to visually assess skewness.

3. Why Handle Skewness?
   Many machine learning models assume normally distributed input features or are affected by skewed data which can reduce model accuracy.

4. Common Techniques to Handle Skewness:

   - Log Transformation: Apply log(x + 1) to reduce right skewness (use +1 to handle zeros).
   - Square Root Transformation: Useful for moderate positive skewness.
   - Box-Cox Transformation: More flexible power transform requiring positive data.
   - Yeo-Johnson Transformation: Handles zero and negative values.

5. Apply Transformations:
   After identification, apply appropriate transformation on skewed numeric columns to make their distribution more normal-like.

6. Post-transformation Checks:
   Re-examine skewness and distributions to confirm improvement.

Data Splitting

1. Purpose
   Splitting data into training and testing subsets validates model performance on unseen data to avoid overfitting.

2. Common Splitting Ratios

   - Typically, 70%-80% data for training, 20%-30% for testing.
   - A validation set can be further separated or obtained via cross-validation.

3. Random State/Seed
   Set a random_state parameter to ensure reproducibility of splits.

4. Implementation using sklearn:

python

```
from sklearn.model_selection import train_test_split



X_train, X_test, y_train, y_test = train_test_split(
    features, target, test_size=0.2, random_state=42
)
```

5. Stratified Splitting (Optional)
   When class labels or key categorical variables are imbalanced, stratified split ensures proportional representation.

TRAIN-TEST SPLIT

Train-Test Split is a fundamental step in machine learning where a dataset is divided into two subsets:

1. Training Set: This portion (usually 70-80% of the data) is used to train the machine learning model. The model learns patterns and relationships from the features and the target variable in this set.
2. Test Set: The remaining portion (typically 20-30%) is reserved for evaluating the model. This set simulates new, unseen data to assess how well the model generalizes and performs outside the training data.

Why Train-Test Split Is Important

- Prevents overfitting by ensuring that the evaluation of the model is on data it hasn't seen before.
- Gives an unbiased estimate of the model's prediction accuracy.
- Helps in tuning model parameters and validating the final model selection.

Common Train-Test Split Strategies:

- Random Splitting: Randomly shuffles and splits data, suitable for large, balanced datasets.
- Stratified Splitting: Maintains class or category proportions between train and test sets, important for imbalanced data.
- Time-based Splitting: For time series data, uses earlier data for training and later data for testing to preserve temporal order.

How to Perform Train-Test Split in Python (scikit-learn)

python

```
from sklearn.model_selection import train_test_split

# Assume X contains features and y contains target variable
X_train, X_test, y_train, y_test = train_test_split(
X, y, test_size=0.2, random_state=42
)
```

- test_size=0.2 means 20% test data, 80% training.
- random_state=42 ensures reproducibility.
- For classification with imbalanced classes, add stratify=y to preserve class balance.

Evaluation After Splitting

- Train your model on the training set.
- Predict and evaluate performance on the test set using metrics like accuracy, mean squared error, or R-squared.

MODEL SELECTION & TRAINING

Model Selection

- **Goal:** Choose the most suitable algorithm(s) for predicting grocery bill amounts.
- **Considerations:** Data size, feature types, expected relationships, interpretability, and performance.
- **Common Regression Models:**
- **Linear Regression:** Simple, interpretable, baseline for continuous target prediction.
- **Decision Trees:** Handle non-linearity and interactions between features.
- **Random Forests:** Ensemble of decision trees, reduces overfitting and

boosts accuracy.

- **Gradient Boosting:** Powerful ensemble boosting method fine-tuning errors iteratively.
- **Support Vector Regression:** Effective for high-dimensional data with kernel tricks.
- **Neural Networks:** Flexible, learn complex patterns but require more data and tuning.

2. Training Models

- **Data Preparation:** Use preprocessed and split data — train features (X_train) and train targets (y_train).
- **Fitting:** Train the selected model(s) on training data.
- **Hyperparameter Tuning:**
- Use grid search or randomized search with cross-validation to optimize hyperparameters.
- Prevent overfitting by selecting parameters that generalize well on validation folds.

Example (Linear Regression training):

python

```
from sklearn.linear_model import LinearRegression

model = LinearRegression()
model.fit(X_train, y_train)
```

3. Model Validation and Cross-Validation

- Evaluate model performance during training using K-Fold Cross-Validation for robustness.
- Prevent overfitting by checking variations between train and validation scores.

Example (Cross-Validation):

python

```
from sklearn.model_selection import cross_val_score
scores = cross_val_score(model, X_train, y_train, cv=5,
scoring='neg_mean_squared_error')
```

4. Model Comparison

- Train different models and compare metrics such as Mean Squared Error (MSE), R-squared, or MAE on validation data.
- Select the model balancing accuracy, robustness, and interpretability.

5. Final Model Training

- Retrain the chosen model on the entire training dataset after hyperparameter tuning.
- Prepare the model for final testing/prediction phase.

Summary

Effective model selection and training involve exploring multiple algorithms, systematically tuning hyperparameters, validating with cross-validation, and comparing performance metrics to ensure the best predictive results for grocery bill forecasting.

Project File Full View

GitHub Link = mailto:https://github.com/virendra-pal-official/Grocery_Bill_Predictor/blob/main/Grocery_Bill_Predictor.ipynb