Product Price Prediction Using Machine Learning

Dataset: Tamimi Supermarket (Saudi Arabia)

Presented by: Nouf Almojel

Course: CS465 – Machine Learning

Instructor: Dr. Abrar Wafa

Agenda

Topics Covered

Phase 1: Introduction

- **Objective**: Predict product prices using ML
- **Business Context:** Optimize pricing, detect high-value products, improve inventory and marketing
- **Dataset Source:** Tamimi Markets – 1,220 real-world products
- **Focus:** Price patterns, unit standardization, feature extraction, model performance comparison
- **Tech Stack:** Python, Pandas, Seaborn, Scikit-learn

Why This Project?

- Saudi-based dataset = high relevance
- Real-world application in retail, economics, and logistics
- Demonstrates complete ML lifecycle: preprocessing → feature engineering → modeling
- Can help supermarkets price competitively while maintaining profitability
- Provides insight into pricing strategy per product category

Tools & Libraries –

- Python (Google Colab for cloud-based notebooks)
- Pandas/NumPy: Data cleaning and transformation
- Seaborn/Matplotlib: Data visualization and plotting
- Scikit-learn: ML algorithms (regression, tree, SVM, NN)
- MLPRegressor: Simple neural network implementation for tabular data

Phase 2- Data Preparation

Key Steps:

- Imported tamimimarkets.csv
- No missing values or duplicate records
- Standardized product sizes into grams/milliliters
- Fixed manual exception for 1 product size
- Split into:
  - 976 products → training
  - 244 products → testing
- Target variable: Price

Code Snippet:

```
df = pd.read_csv("tamimimarkets.csv")
df.drop_duplicates(inplace=True)
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2)
```

Phase 3- Exploratory Data Analysis (EDA)

Key Observations:

- Price Range: 0.3 SAR to 123 SAR
- Median Price: 9.5 SAR
- Mean Price: 12.3 SAR (affected by outliers)
- Std Dev: 11.85 SAR (high variability)
- Outlier: Volvic water at 123 SAR
- Cheapest: Tamimi branded water at 0.3 SAR

Code Snippet:

```
plt.hist(df["Price"], bins=30)
sns.heatmap(df.corr(), annot=True)
```

Phase 4- Feature Engineering

Features Created:

- Price_per_unit: Price / Standardized Size
- Is_Premium: Flagged if price > 75th percentile (16.95 SAR)
- Product Category: Derived from name (e.g., dairy, meat, grains)
- Size_Standardized: Unified measure (g/ml)
- Dropped non-informative or ID columns

Code Snippet:

```
df["Price_per_unit"] = df["Price"] /
df["Size_Standardized"]
df["Is_Premium"] = df["Price"] >
df["Price"].quantile(0.75)
```

Feature Importance Chart

Top Features by Correlation:

- Is_Premium: 0.735
- Size_Standardized: 0.400
- Price_per_unit: 0.206
- Size_Value: 0.051

“Premium" label is the strongest indicator of price.

Phase 5- Model Building

Models Trained:

1. Linear Regression – Simple, interpretable
2. Decision Tree – Rule-based model
3. SVM – Margin-based regression
4. Neural Network – Captures non-linearity

Linear Regression

- Quick & interpretable
- Captures linear relationships
- Struggles with non-linearities
- Baseline $R^2 \approx 0.46$

Code Snippet:

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

Decision Tree

- Captures non-linear decision boundaries
- Easy to visualize + explain
- Initially overfit (training $R^2 = 1.0$)
- After tuning:
  - $R^2 = 0.816$
  - RMSE = 7.79 SAR

SVM & Neural Network

Support Vector Machine:

- Finds optimal hyperplane
- Handles small feature sets well

Neural Network (MLP):

- 2 hidden layers: 64 neurons each
- Captures complex nonlinear interactions
- Slightly higher training time

Phase 6- Model Evaluation

Metrics:

- R^2 Score: Measures variance explained
- RMSE: Root Mean Square Error in SAR
- Best Model: Decision Tree
  - R^2 = 0.816
  - RMSE = 7.79 SAR

Phase 7- Insights & Reporting

Key Findings:

- Premium flag = top feature
- Larger products tend to be pricier
- Grains/Bakery → Most expensive category
- Dairy → Least expensive
- Average unit price: 0.39 SAR
- Volvic’s 123 SAR bottle = outlier

What Could Be Improved?

- Add brand reputation as a feature
- Incorporate seasonality or time-of-year
- Track sales or promotions
- Use product pairings or bundling info
- Add customer reviews or demand estimates

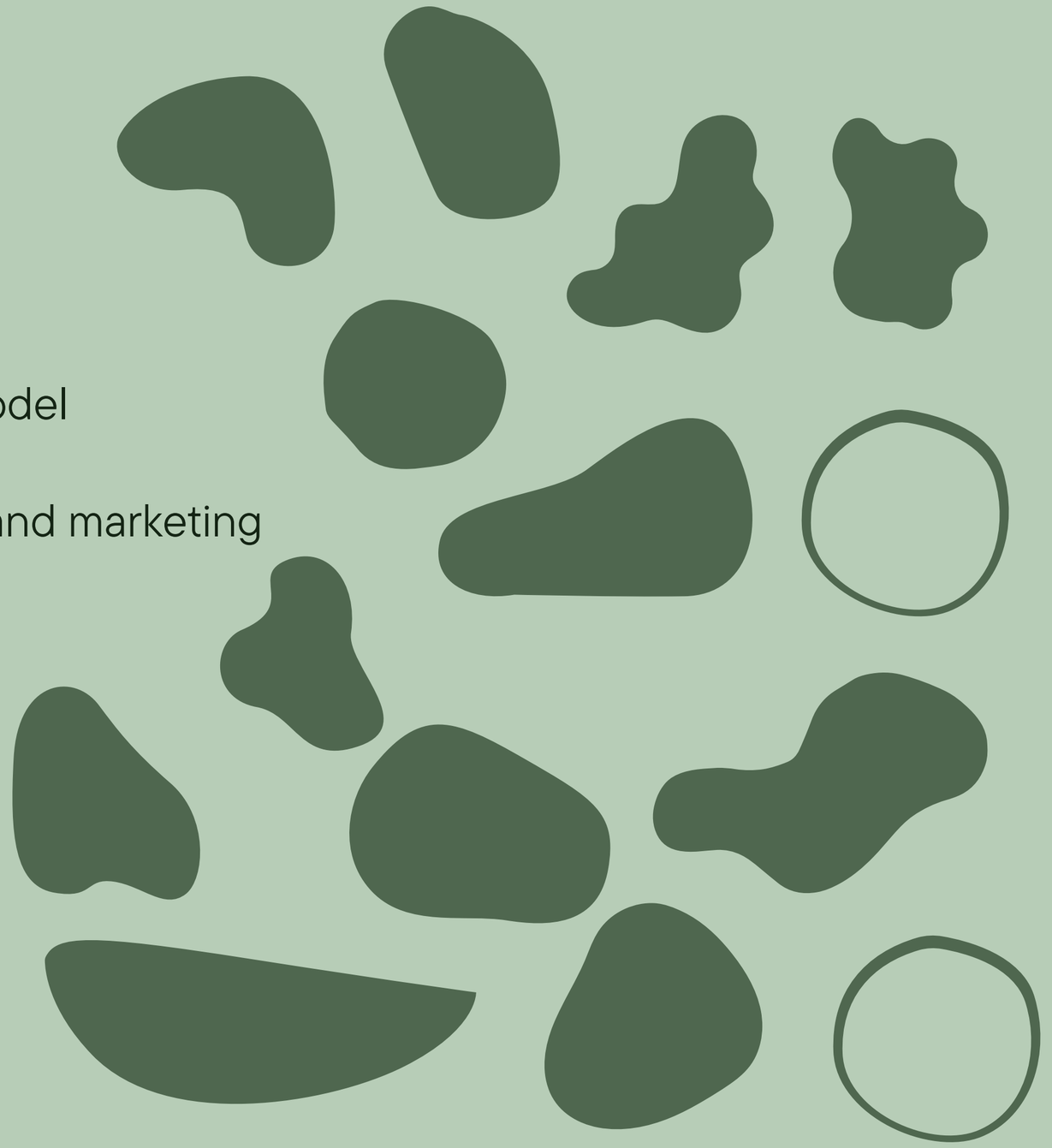
Extra Resources-

Code Link

Report Link

Conclusion

- Successfully built a working ML pipeline
- Decision Tree = most balanced & powerful model
- Feature engineering drove performance
- Real-world applications in pricing, inventory, and marketing
- Room for improvement with richer features



Thank you!

Any Questions

Email: 222410007@psu.edu.sa