**Market Basket Analysis on Edinburg Bakery Data : Data Pre-processing, Data Exploration and Visualizations and Association Rules Mining**

**Introduction**

**1. Market Basket Analysis**

Market Basket Analysis is one of the key techniques used to uncover associations between items. It works by looking for combinations of items that occur together frequently in transactions. Market basket analysis aims to understand customer behaviour and preferences by identifying which products are often purchased together. This information can be used for various purposes, such as product recommendations, store layout optimization, and targeted marketing campaigns**.**

**Project Objective :**

In this project, we aimed to uncover patterns and associations between items purchased together by applying Association Rule Mining techniques, and then used the insights to build classification models to predict customer behaviour.

First, we performed Market Basket Analysis using the Apriori Algorithm to discover frequent item sets and generate strong association rules. Key metrics such as Support, Confidence, and Lift were used to evaluate and filter the rules:

* Support Threshold: 1% (approx. 95 transactions out of 9465)
* Minimum Lift: 1.2 to ensure that only strong associations are considered.
* Rules were sorted by Confidence to prioritize the most reliable relationships.

After identifying important items through association rules, we developed several classification models to predict whether a transaction happened during the weekend. The selected features included the top items identified from Market Basket Analysis along with other engineered features.

The machine learning algorithms used for classification included:

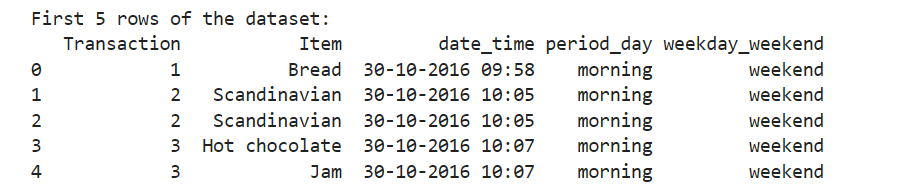
* Decision Tree Classifier
* Random Forest Classifier
* Support Vector Machine (SVM)
* XG Boost Classifier (if available)

We applied SMOTE (Synthetic Minority Over-sampling Technique) to balance the data during model training, and used 5-Fold Cross-Validation along with Grid Search to optimize hyperparameters.

Finally, we compared the models based on Accuracy, Precision, Recall, and F1-Score to select the best-performing model.

**About the Dataset**

The dataset belongs to "The Bread Basket" a bakery located in Edinburgh *(publicly available on*[*Kaggle*](https://www.kaggle.com/datasets/ytgangster/online-sales-in-usa)*)*. This dataset includes 20507 entries and 4 columns, recording over 9000 transactions of customers who ordered different items from this bakery online from 2016-01-11 to 2017-12-03**.**



**Problem Statement:**

The goal of this project is to analyse customer purchasing behaviour at "The Bread Basket" bakery by uncovering associations between items that are frequently bought together and to use these insights to predict customer behaviour. Specifically, we aim to:

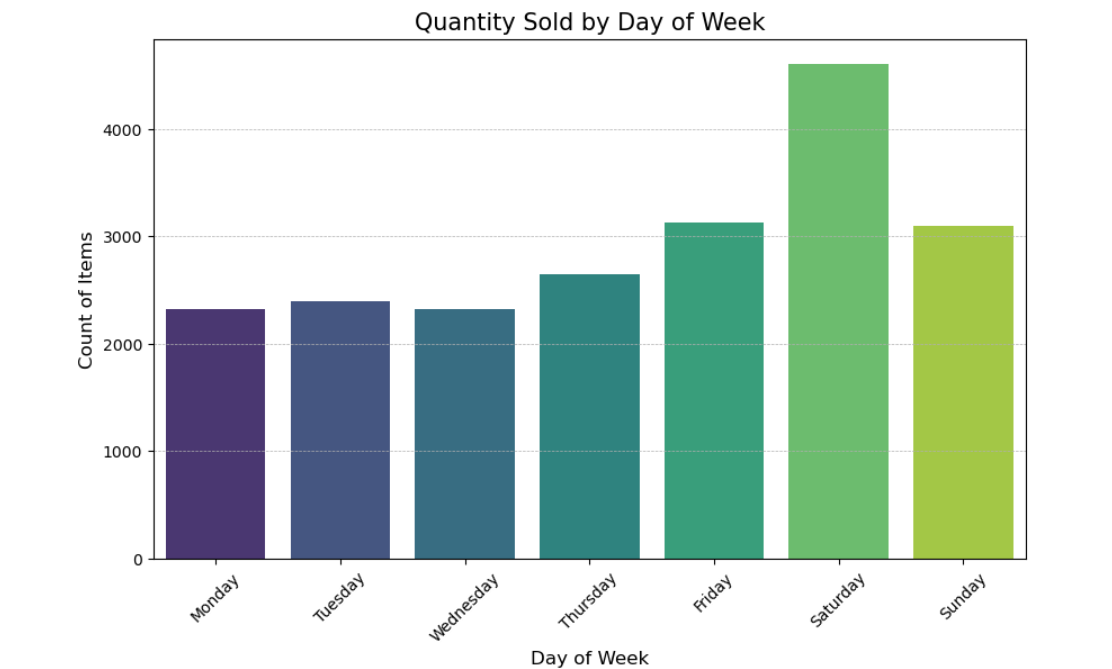
* Identify frequent item sets and generate strong association rules through Market Basket Analysis to better understand product affinities.
* Develop machine learning classification models to predict whether a customer transaction occurred during the weekend based on purchase patterns.
* Use predictive insights to potentially support marketing strategies, inventory management, and sales optimization for the bakery.

**Data Preprocessing**

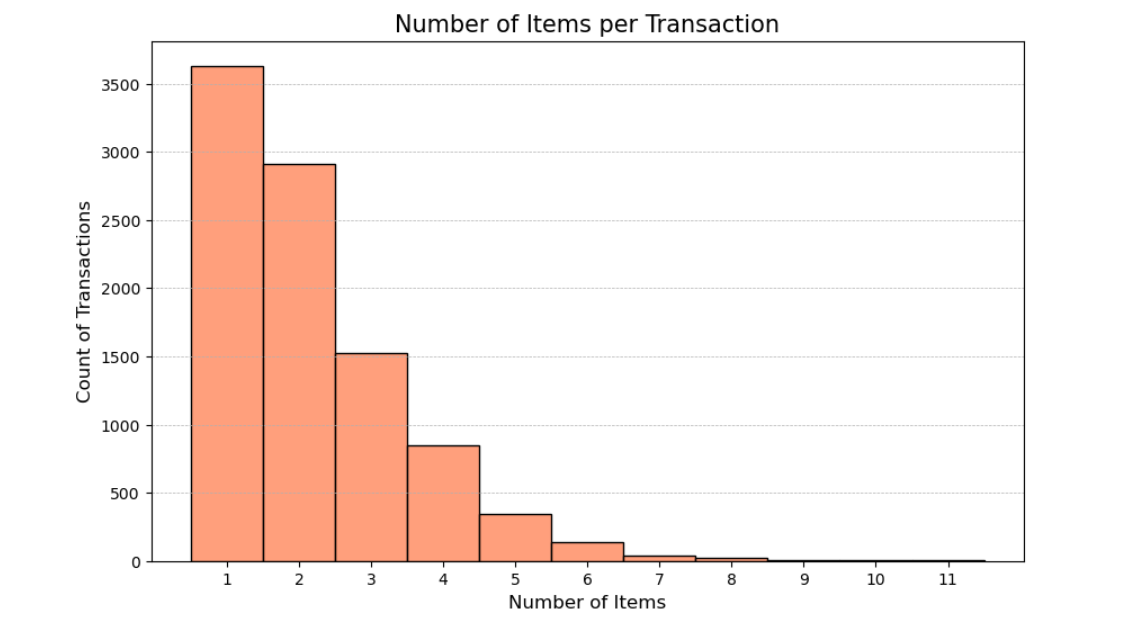
1. **Missing Values Check**:
   * Checked for missing values in the dataset using df.isnull().sum().
   * Confirmed that **no missing values** were present, so no imputation or deletion was necessary.
2. **Datetime Processing**:
   * Converted the date time column to **datetime format** using pd.to\_datetime.
   * Extracted additional time-based features:
     + **date** (only the date part),
     + **hour** (hour of the day),
     + **month** (year-month in "YYYY-MM" format),
     + **weekday** (day of the week, like Monday, Tuesday, etc.).
   * Dropped the original date\_time column as it was no longer needed.
3. **Text Cleaning**:
   * Cleaned the Item column:
     + Stripped leading and trailing spaces,
     + Converted all item names to **lowercase** to standardize entries.
4. **Encoding Categorical Variables**:
   * Applied **One-Hot Encoding** to the period\_day column using OneHotEncoder, dropping the first category to avoid multicollinearity.
   * Created new columns representing the encoded values for different periods of the day.

Data Visualization:

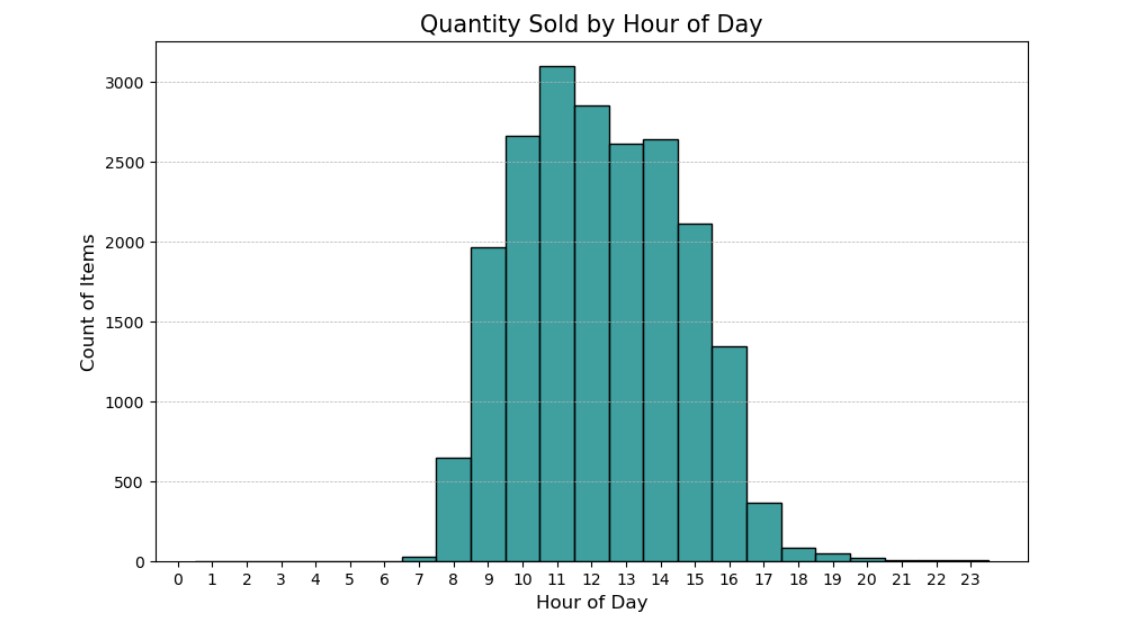
**Quantity Sold by Day of Week (First Diagram)**: This histogram illustrates the distribution of items sold across the days of the week, revealing Saturday as the peak sales day with approximately 4500 items sold, followed by Friday at around 3500 items. Sales remain relatively stable from Monday to Thursday, fluctuating between 2200 and 2500 items, with Tuesday marking the lowest at about 2200 items. Sunday sees a moderate decline to around 3000 items, indicating a clear trend of higher sales towards the end of the week, particularly on weekends, which businesses can target for increased inventory or promotions.



**Number of Items per Transaction**: The histogram displays the frequency of transactions based on the number of items purchased, showing a right-skewed distribution where single-item transactions dominate with around 3500 occurrences. Transactions involving 2 items drop sharply to about 1500, and those with 3 items further decrease to roughly 500, with a steep decline thereafter—transactions with 4 or more items are increasingly rare, and those with 8 or more items are nearly negligible. This suggests that most customers prefer purchasing single items, which could inform strategies for product bundling or promotions to encourage larger purchases.



**Quantity Sold by Hour of Day (Third Diagram)**: This histogram tracks the number of items sold per hour across a 24-hour period, highlighting a peak in sales between 9 AM and 12 PM, with the highest volume around 11 AM at approximately 3000 items. Sales begin to rise after 6 AM, decline gradually after 2 PM, and drop significantly after 5 PM, with minimal activity from midnight to 5 AM and after 8 PM, where counts approach 0. This pattern indicates that midday hours, particularly late morning, are the busiest, suggesting businesses should focus resources during these times while exploring ways to stimulate sales during off-peak hours.



**Feature Engineering:**

**1. Bag-of-Words (BoW) Representation:**

* A sparse matrix was created where each row represents a transaction and each column represents an item.
* The matrix contains binary values: 1 if the item is present in the transaction, and 0 if it is absent.
* This BoW matrix was used as input for further feature creation and mutual information analysis to understand item relationships within transactions.

**2. TF-IDF Features:**

* Applied TF-IDF (Term Frequency-Inverse Document Frequency) transformation on the list of items per transaction.
* Extracted the top 10 most informative TF-IDF features, capturing the relative importance of each item within the transactions.
* These features helped prioritize items that are distinctive and informative across transactions, offering insight into customer purchasing behavior.

**3. Standardization of Numerical Features:**

* The code applies standardization to a selected set of numerical features in the dataset, such as hour, transaction\_count, and items\_per\_transaction. Using StandardScaler, it scales these features to have a mean of 0 and a standard deviation of 1, ensuring that all numerical features are on a similar scale. The scaled features are then concatenated back into the original DataFrame with a \_scaled suffix to distinguish them. This step helps improve the performance of machine learning models by preventing features with larger numeric ranges from dominating the model's learning process.

**Model Building :**

In the model-building phase of this project, two key models were developed: the **Apriori Model** for association rule mining and several **Classification Models** to predict whether a transaction occurred on a weekend.

**1. Apriori Model:**

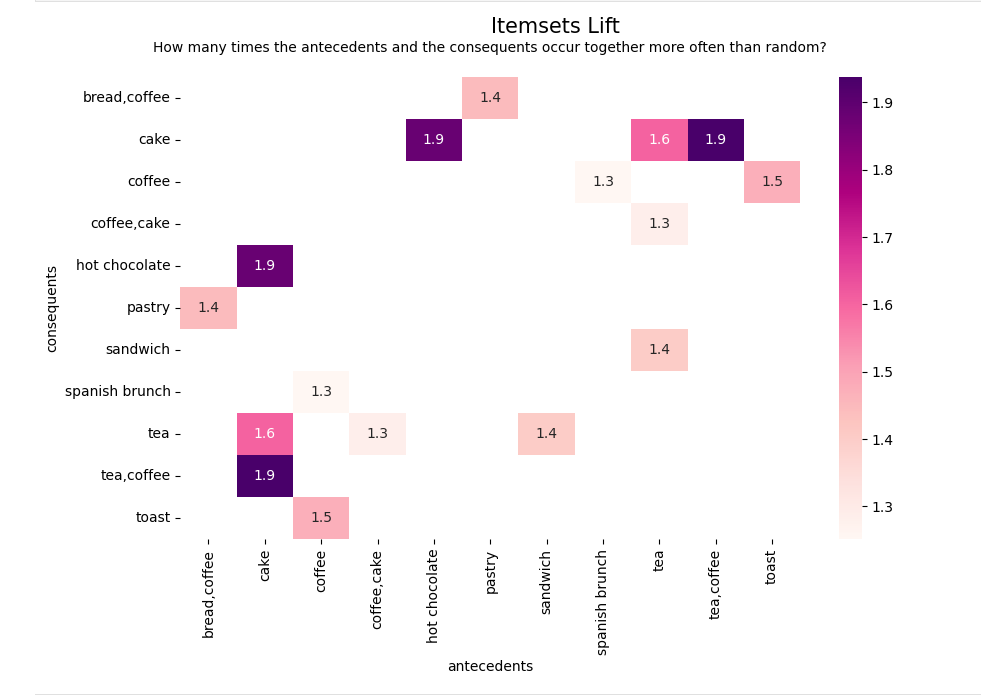
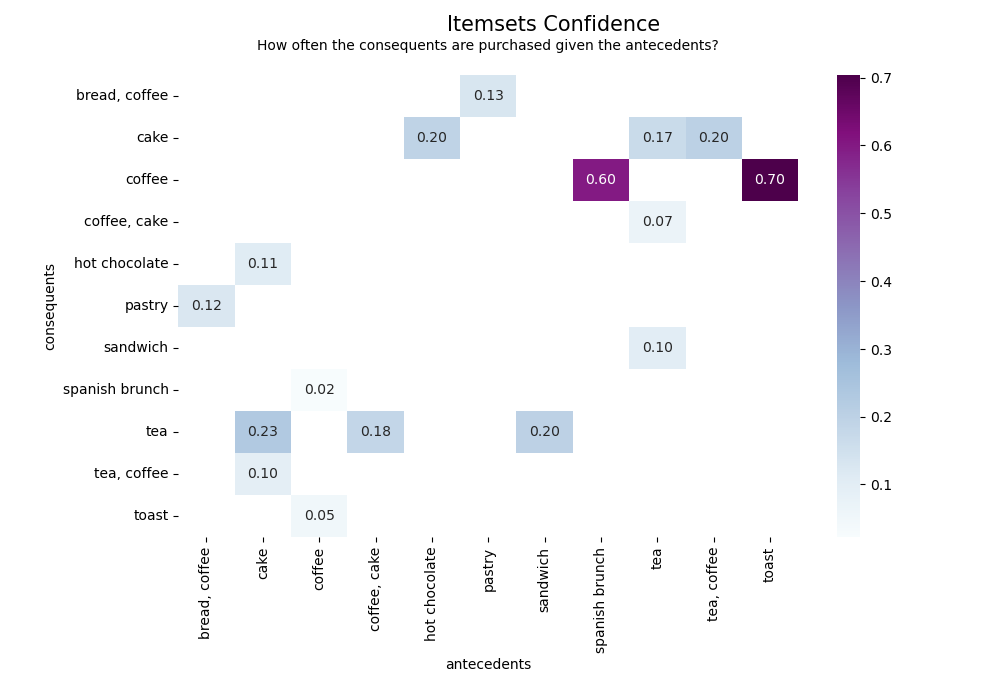
* **Transaction Counts**: The model first aggregates the data to count the number of transactions for each item.
* **Binary Basket**: A binary matrix was created for the items in each transaction, with 1 indicating the presence of an item and 0 indicating its absence.
* **Frequent Itemsets**: The Apriori algorithm was applied to discover frequent itemsets with a minimum support of 1%.
* **Association Rules**: Association rules were generated using metrics such as lift and confidence. The rules were sorted based on confidence to identify the strongest relationships.

**2. Classification Models:**

* **Features**: A comprehensive set of features was engineered for the classification task, including numerical features (e.g., hour, transaction count), top item presence indicators, TF-IDF features, and interaction features.
* **Target Variable**: The target variable was is\_weekend, indicating whether a transaction occurred on a weekend.
* **Model Selection**: Several classification models were tested, including:
  + Decision Tree
  + Random Forest
  + Support Vector Machine (SVM)
  + XGBoost (if available)
* **Cross-Validation**: A 5-fold cross-validation approach was used to assess model performance. For each model, hyperparameter tuning was done using grid search.
* **SMOTE**: Synthetic Minority Over-sampling Technique (SMOTE) was employed to address class imbalance by generating synthetic examples for the minority class.

**Model Evaluation**:

**Apriori Algorithm :**

**Overall Evaluation of the Apriori Algorithm:** The Apriori algorithm demonstrates effectiveness in identifying actionable association rules, as seen in high-confidence rules like coffee → toast (0.70) and high-lift rules like cake → coffee (1.9), which can guide business decisions such as bundling or targeted marketing. However, the presence of low-confidence rules (e.g., spanish brunch → coffee at 0.02) and a modest range of lift values (1.3 to 1.9) suggest that the algorithm might be capturing some weaker associations, potentially due to overly lenient support or confidence thresholds. Fine-tuning these parameters could reduce noise and focus on stronger patterns. Additionally, the algorithm’s success in this context highlights its utility for small to medium-sized itemsets in retail settings, but for larger datasets, computational efficiency might become a concern due to Apriori’s reliance on generating and pruning candidate itemsets.

**Other Models Comparison:**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | | Model | Accuracy | Precision (0) | Precision (1) | Recall (0) | Recall (1) | F1-Score (0) | F1-Score (1) | | Decision Tree | 0.8371 | 0.7722 | 0.7325 | 0.8684 | 0.7275 | 0.8606 | 0.7495 | | Random Forest | 0.8106 | 0.7247 | 0.6853 | 0.8510 | 0.6938 | 0.8327 | 0.7024 | | SVM | 0.7623 | 0.6573 | 0.7024 | 0.7923 | 0.7024 | 0.8044 | 0.6791 | | XGBoost | 0.8484 | 0.8238 | 0.7625 | 0.9013 | 0.7592 | 0.8924 | 0.7919 | |

**Best Performing Model:**

* **XGBoost** is the best-performing model with the highest accuracy (84.84%), precision for both classes, recall for class 0 (0.9013), and a balanced F1-Score of 0.7919.
* It significantly outperforms the other models in precision, recall, and F1-Score, making it the most suitable model for predicting your target variable.

**Summary:**

* **Decision Tree**: Provides a good balance between accuracy and precision/recall for both classes. It is a strong contender when interpretability is important.
* **Random Forest**: While slightly lower in performance than the Decision Tree and XGBoost, it still performs well and is robust to overfitting.
* **SVM**: While its performance is decent, it has a lower accuracy and recall compared to XGBoost and Decision Tree.

**Insights and Recommendations for Decision Making**

**Data Exploration Insights**

1. **Best Selling Items:**
   * **Coffee** leads sales with **26.7%**, followed by **bread** at **16.2%**.
   * Over **80%** of items have low sales frequency, contributing less than **1%** to total sales.
2. **Transaction Patterns:**
   * **38%** of transactions involve a single item, while **95%** have **5 or fewer items**.
   * Peak sales occur from **9 AM to 2 PM**, accounting for **78%** of total sales.
   * **Weekends** outperform weekdays, with **Saturdays** seeing **20% higher sales** and **Wednesdays** having **30% lower sales**.

**Market Basket Analysis Insights**

* **Coffee** appears in **47.8%** of transactions and is involved in **67.7%** of association rules.
* **Toast** and **Spanish brunch** show strong links to coffee, with **70%** and **60%** of buyers purchasing coffee, respectively.
* Strong associations are found between **cake and tea**, and **hot chocolate and cake**.

**Recommendations for Decision Making**

1. **Inventory Optimization:**
   * **Suspend low-sales items** (e.g., polenta, bacon) to save on storage costs.
2. **Peak Hour Resource Allocation:**
   * Increase resources during **9 AM to 2 PM** and offer promotions for **post-6 PM** orders.
3. **Weekday Sales Boost:**
   * Launch a **"Happy Wednesday"** promotion to increase sales on **Wednesdays**, the slowest day.
4. **Combo Offers:**
   * Promote bundles like **cake & juice**, **sandwich & tea**, and **pastry & coffee** to boost overall sales.

**Future Analysis**

1. **Investigate Sales Spikes:**
   * Analyze the **Nov 2016 - Mar 2017** surge to understand if marketing or external factors drove the increase.
2. **Optimize Conditions:**
   * Study the factors behind high sales periods to replicate or improve them for sustained growth.