Integrated and Individual CA

Capstone Project Proposal
Author: Eduardo Matos
Lecturer: Sam Weiss / David McQuaid
Subject: Machine Learning for Business / Data Visualisation Techniques
CCT College Dublin

CCT College Dublin

Assessment Cover Page

Module Title:	**Machine Learning for Business** **Data Visualisation Techniques**
Assessment Title:	*Integrated and Individual CA*
Lecturer Name:	**David McQuaid** **Sam Weiss**
Student Full Name:	**Eduardo Matos**
Student Number: **Assessment Due Date:**	**Sba23298 - Eduardo Matos** **28/07/2024**
Date of Submission:	**28/07/2024**

Declaration

By submitting this assessment, I confirm that I have read the CCT policy on Academic Misconduct and understand the implications of submitting work that is not my own or does not appropriately reference material taken from a third party or other source. I declare it to be my own work and that all material from third parties has been appropriately referenced. I further confirm that this work has not previously been submitted for assessment by myself or someone else in CCT College Dublin or any other higher education institution.

Index

Question 1 (1788 words)

- Briefly describe the importance of time series forecasting models for business applications that can be used in a particular scenario and explain this by providing at least two examples.
- Your goal is to forecast based on the use of the experimental data provided at the above-mentioned link to create time series machine learning models for forecasting the appliances energy usage in a low energy building.
- Use an appropriate train/test split to develop a model, and determine its forecast errors. Evaluate the performance in the context of the dataset.

Briefly describe the importance of time series forecasting models for business applications that can be used in a particular scenario and explain this by providing at least two examples.

Time series forecasting models are of significant importance in business for a number of reasons.

The application of demand forecasting models enables businesses to anticipate future product demand, thereby facilitating optimal inventory management and supply chain optimization. Such models assist businesses in forecasting future product demand, thereby facilitating the management of inventory and the optimisation of supply chains.

Financial planning is also facilitated by the use of time series forecasting models. Such models assist in the prediction of revenue, expenses and cash flows, which is essential for the formulation of budgets and financial plans.

The following example scenarios illustrate the potential applications of time series forecasting in various business contexts.

The retail sector employs time series forecasting to predict daily sales, thereby optimising stock levels and reducing the likelihood of overstock or stockouts.

The energy sector can also benefit from the use of time series forecasting. An energy provider utilises electricity demand forecasting in order to efficiently manage grid loads and schedule maintenance.

Your goal is to forecast based on the use of the experimental data provided at the above-mentioned link to create time series machine learning models for forecasting the appliances energy usage in a low energy building.

<u>Exploratory data analysis</u>

Given that the target variable in the dataset is the electricity consumption of household appliances (hereafter referred to as 'Appliances'), it is plausible that other features, such as temperature, may exert an influence upon it. However, it can be reasonably assumed that there is a direct correlation between electricity consumption and time. For instance, it is reasonable to posit that electricity usage in the evening will be higher than during the day, given the typical use of lighting and the prevalence of other household activities, such as watching television. Conversely, the target variable should be at its lowest from midnight to early morning, as most people are asleep, resulting in minimal electricity usage.

Accordingly, in order to more accurately observe the distribution of the target variable over time, the following variables have been introduced.

Overview the entire dataset.

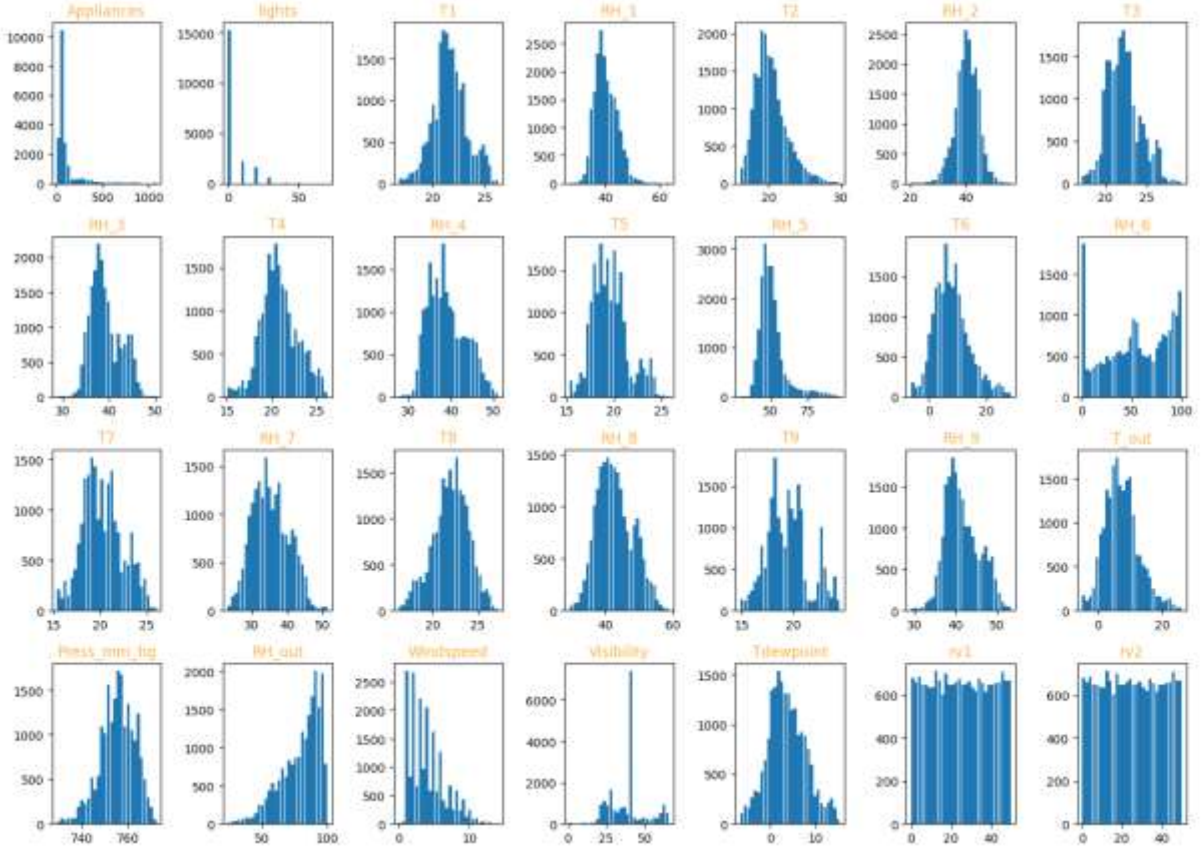


Image 1. Overview the entire dataset.

It is possible to observe the shape of the distribution for each feature. For instance, certain features, such as temperature and humidity, may exhibit a normal distribution, whereas others may display a skewed or multimodal distribution.

Furthermore, it can be observed that the values RV1 and RV2 are merely random variables that have been added and could be disregarded if data other than that pertaining to 'Appliances' were to be employed.

Energy Consumption by Hour

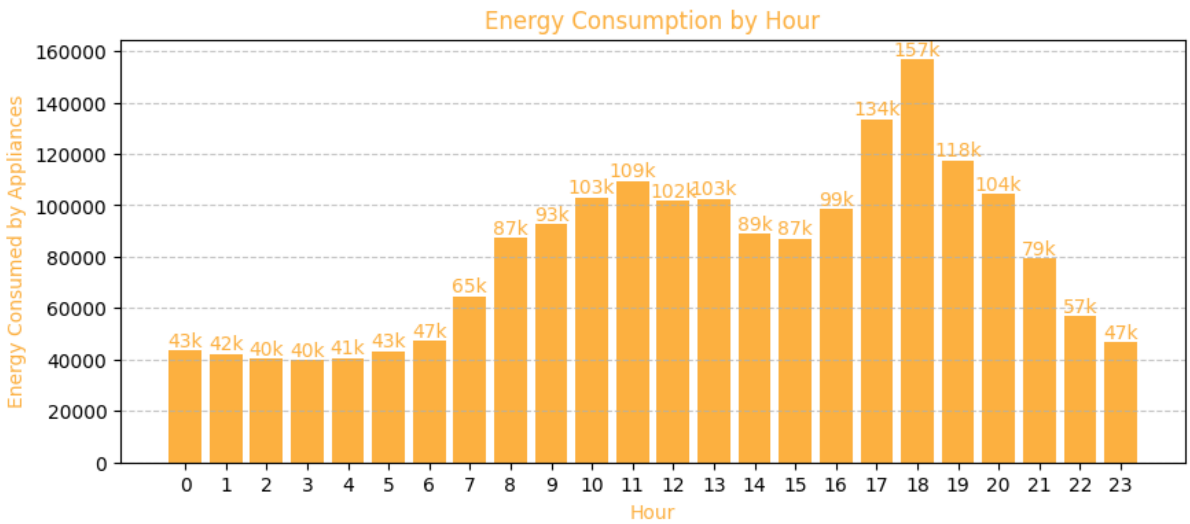


Image 2. Energy Consumption by Hour.

The graph above clearly shows that electricity consumption is highest between 17:00 (5pm) and 20:00 (8pm). This peak time is probably when most people are at home after work or school. During these hours, households tend to use more electricity for activities such as cooking dinner, using household appliances such as washing machines and dishwashers, and for entertainment activities such as watching TV or using computers. This pattern suggests that household electricity use peaks in the evening hours.

Conversely, the graph shows that electricity consumption is lowest between 23:00 (11:00 pm) and 06:00 (6:00 am) the following day. This off-peak period coincides with the time when most people are asleep and therefore demand for electricity is minimal. During these hours, the use of household appliances, lighting and other electrical equipment is significantly reduced. The drop in consumption during these hours is consistent with the expected behaviour of reduced activity and lower energy demand overnight.

Our initial hypothesis was that electricity consumption would be highest in the evening hours when people are at home and active, and lowest in the late night and early morning hours when people are asleep. The observed data from the graph fits this hypothesis perfectly. The clear peaks in the evening and troughs in the late night and early morning provide strong evidence to support our initial hypothesis. This consistency in the data across different days suggests that the pattern is reliable and not an anomaly.

The recurring patterns of peak and off-peak electricity consumption reinforce the reliability of the data. If the data were inconsistent or showed random fluctuations with no discernible pattern, it would raise questions about its accuracy. However, the clear, repetitive daily cycles observed here confirm that the data is reliable. Understanding these consumption patterns is crucial for planning and managing energy resources. It can help utilities with load forecasting, demand-side management and optimising energy distribution to ensure a stable and efficient supply.

Energy Consumption by Day of the Week

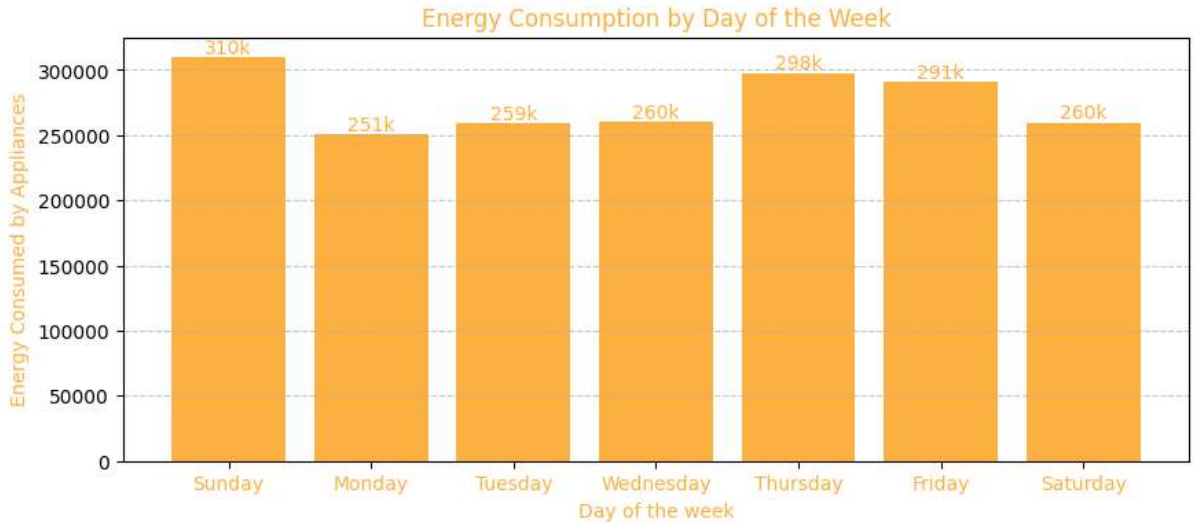


Image 3. Energy Consumption by Day of the Week

Relationship between Date and Appliances with simple moving average

As with the analysis of the hours within a single day, it is possible that a pattern may emerge when the data from an entire week is considered. However, when the data set is split by the day of the week, no significant features emerge in the distribution of the target variable. It is noteworthy that electricity consumption on Sundays is approximately 10% higher than on other days.

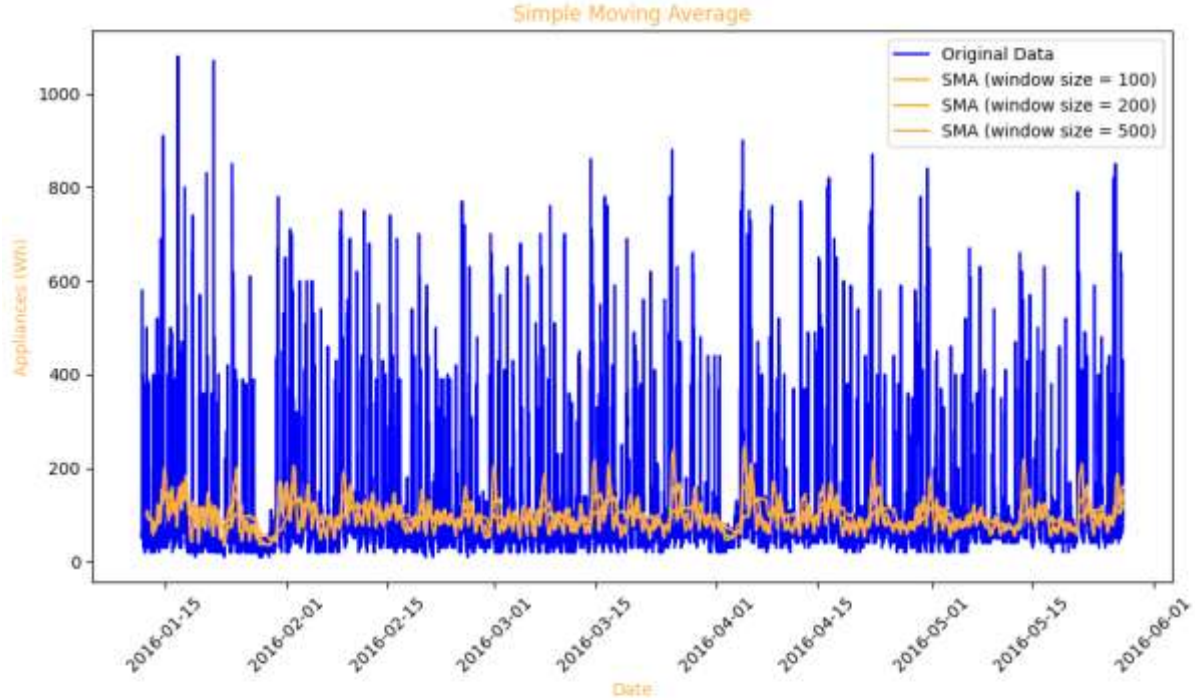


Image 4. Relationship between Date and Appliances with simple moving average.

There is no clear trend in the SMA plot. Appliance usage is stable with minor fluctuations.

Seasonality

The data does not show strong seasonal patterns. There are no obvious cycles or repeating patterns by time of year, month or week.

There are no unusual spikes or drops in appliance usage. The fluctuations are normal.

Pairwise Correlation Heatmap

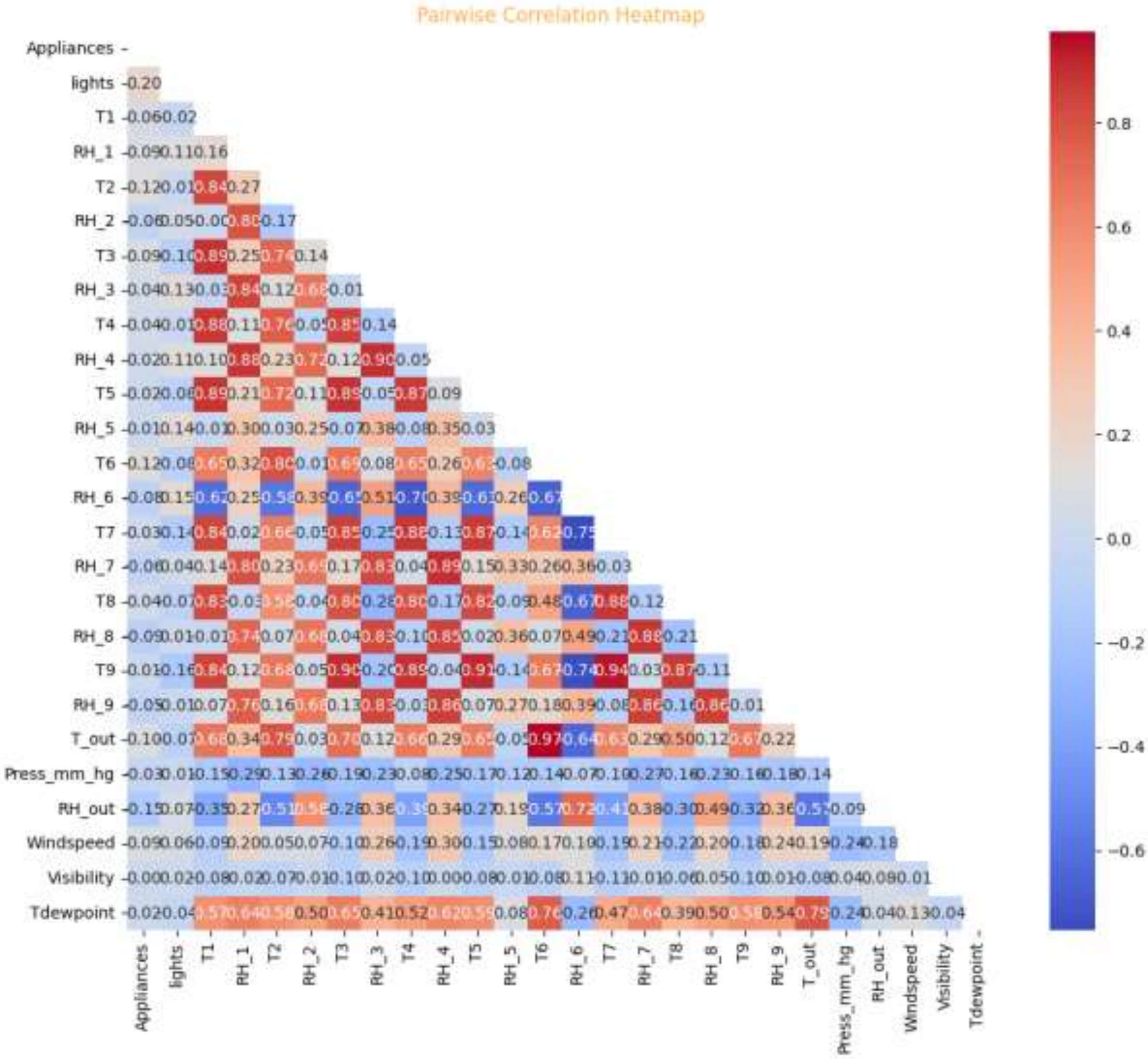


Image 5. Pairwise Correlation Heatmap

The objective of this visual representation is to present the correlations among different features using a heatmap. The data demonstrate a robust positive correlation between temperature and humidity levels. It is notable that RH_6 (outdoor humidity) exhibits a strong negative correlation with temperatures, indicating that as temperatures increase, humidity levels decrease.

In contrast to the observed correlations among temperatures and humidities, the correlation between our target variable, 'Appliances', and these variables appears to be insignificant.

Use an appropriate train/test split to develop a model and determine its forecast errors. Evaluate the performance in the context of the dataset.

ARIMA model implementing

- ARIMA is generally insensitive to feature scaling, but LSTM models are sensitive to feature scaling because they are a type of RNN. Our dataset has no missing values, so imputation is not needed. This makes the modelling process smoother.
- 'Appliances', 'date', 'Windspeed', 'Visibility', 'Tdewpoint', 'rv1', 'rv2 are removed from the training set for the following reasons:
- Appliances: This is the target variable, so it should be separated from the features.
- Date: It is time-related.
- rv1 and rv2: These are random variables introduced for testing purposes, not actual features of interest.
- Windspeed, Visibility, Tdewpoint: These columns are not relevant for this analysis.

Seasonal decomposition

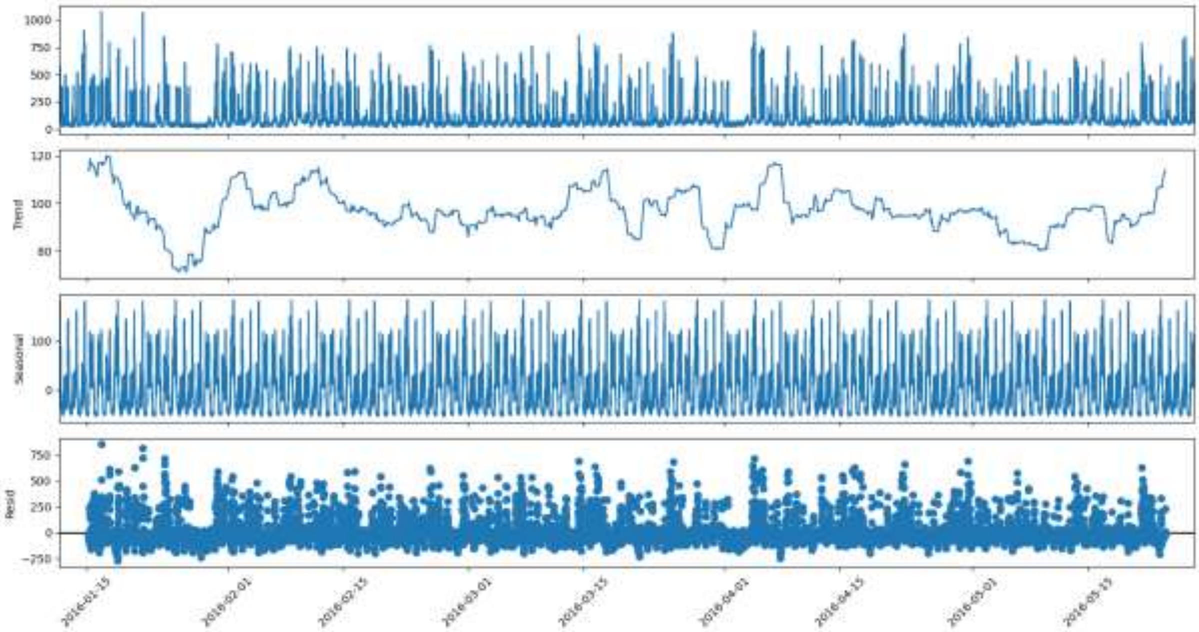
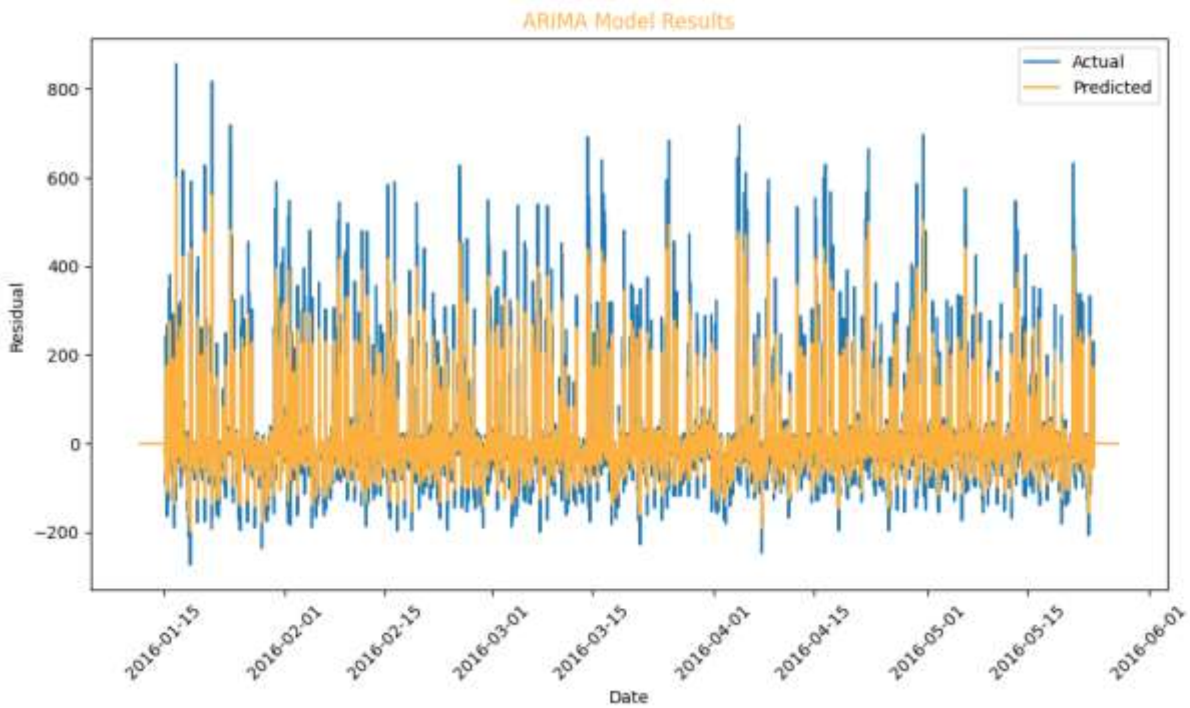


Image 6. Seasonal decomposition

ARIMA works well with linear trends and seasonality. However, the data seems to be stationary.

ARIMA Model Results



The autoregressive (AR) model relies on past observations to predict future values, so I have carefully chosen the appropriate start and end times.

Arima Model Scores

Mean Absolute Error (MAE): 30.430244453539146
Root Mean Squared Error (RMSE): 62.33467043548941
Mean Absolute Percentage Error (MAPE): 31.594001370590114

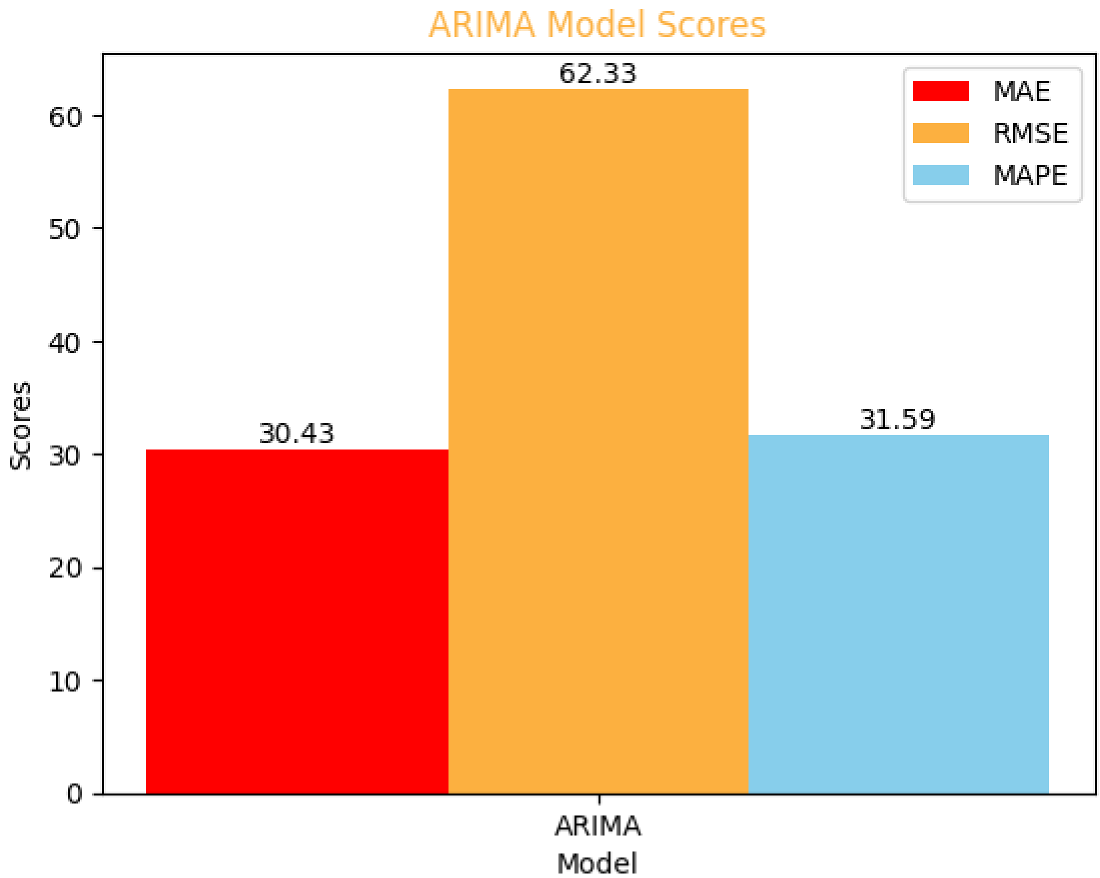


Image 7. Arima Model Scores.

- Mean Absolute Error (MAE): 30.43

The model's predictions are, on average, 30.43 units off. This is a simple way to measure how accurate a prediction is.

- Root Mean Squared Error (RMSE): 62.33

This value suggests that the model makes larger errors sometimes.

- Mean Absolute Percentage Error (MAPE): 31.59%

This means that the prediction errors are about 31.59% of the actual values. The model is moderately accurate.
The ARIMA model provides a reasonable prediction, but there's room for improvement.

Question 3 for Question 1

Energy Consumption Over Time

Image 8. Energy Consumption Over Time.

Code explanation

```python
# Initialize the Dash app
app = dash.Dash(__name__)
```

This creates a Dash application instance, which serves as the backbone of the dashboard.

```python
# Define the layout
app.layout = html.Div(style={'fontFamily': 'Arial', 'padding': '20px'}, children=[
    html.H1('Energy Consumption Dashboard', style={'textAlign': 'center', 'color': '#FCB040'}),

    html.Label('Select Data Range:', style={'fontSize': '20px', 'color': '#FCB040'}),
    dcc.DatePickerRange(
        id='date-picker-range',
        start_date=df['date'].min(),
        end_date=df['date'].max(),
        display_format='YYYY-MM-DD',
        style={'fontSize': '18px'}
    ),

    html.Label('Energy Consumption Over Time', style={'fontSize': '20px', 'color': '#FCB040'}),
    dcc.Graph(id='time-series-chart'),

    html.Label('Appliances Energy Consumption Distribution', style={'fontSize': '20px', 'color': '#FCB040'
    dcc.Graph(id='histogram-chart')
])
```

This part defines the structure and style of the dashboard:

HTML Div and Header: The html.Div component acts as a container for all other components, with html.H1 setting the main title of the dashboard, centred and coloured in #FCB040.
Date Picker: The dcc.DatePickerRange component allows the user to select a date range. It initialises with the minimum and maximum dates from the dataset, formatted as 'YYYY-MM-DD' and styled for readability.
Graphs: Two dcc.Graph components are defined, one to display the time series graph of energy consumption over time, and the other to display a histogram of the distribution of energy consumption of the appliances. Labels and titles are styled to match the overall theme.

```
@app.callback(
    Output('time-series-chart', 'figure'),
    [Input('date-picker-range', 'start_date'),
     Input('date-picker-range', 'end_date')]
)
def update_time_series(start_date, end_date):
    filtered_df = df[(df['date'] >= start_date) & (df['date'] <= end_date)]
    fig = px.line(filtered_df, x='date', y='Appliances', title='Energy Consumption Over Time')
    fig.update_layout(title_font_size=24, xaxis_title='Date', yaxis_title='Appliances (Wh)',
                      title_font_color='#FCB040', font=dict(size=18), plot_bgcolor='white')
    return fig

@app.callback(
    Output('histogram-chart', 'figure'),
    [Input('date-picker-range', 'start_date'),
     Input('date-picker-range', 'end_date')]
)
def update_histogram(start_date, end_date):
    filtered_df = df[(df['date'] >= start_date) & (df['date'] <= end_date)]
    fig = px.histogram(filtered_df, x='Appliances', title='Distribution of Energy Consumption')
    fig.update_layout(title_font_size=24, xaxis_title='Appliances (Wh)', yaxis_title='Count',
                      title_font_color='#FCB040', font=dict(size=18), plot_bgcolor='white')
    return fig
```

This function filters the data set based on the selected date range and creates a histogram of the energy consumption of the devices.

```
if __name__ == '__main__':
    app.run_server(debug=True)
```

This line starts the Dash server in debug mode, allowing live updates and easy troubleshooting during development.

Question 4 for Question 1

Explain how your dashboard is designed with the above-mentioned demographic in mind and your rationale for the types of the visualisations that you have developed in Question 3.

The interactive dashboard is designed with the needs of older adults (60+) in mind, prioritising simplicity, clarity and ease of use.

Firstly, the text and layout are tailored to ensure readability. The dashboard uses large fonts (18px or larger) in Arial, a simple and legible font. This choice helps older adults who may have visual impairments. High contrast colours, such as #FCB040 for titles, are used against a white background to improve visibility. The overall layout is straightforward and uncluttered, presenting information in a clear and organised manner. This minimises cognitive load and makes navigation easier for non-technical users.

Secondly, the interactive elements are designed to be intuitive and accessible. The date range selector allows users to filter data by selecting a range of dates. This element is large and easy to click on to accommodate users with limited dexterity. When users select a date range, the dashboard provides immediate visual feedback by updating the relevant charts. This feature keeps users engaged and helps them understand the impact of their actions.

The dashboard includes two main types of visualisations: a time series chart and a histogram.

The time series chart shows energy consumption over time. Line charts are simple and effective for showing trends over time, helping users easily understand how energy consumption changes within the selected time period. The x-axis represents the date, and the y-axis represents the energy consumed. The graph is updated based on the selected date range, allowing users to see trends over different time periods.

The histogram shows the distribution of energy levels. Histograms are a simple and effective way to show the distribution of data. They help users understand the frequency of different power levels. The x-axis represents the energy consumption levels and the y-axis represents the number of occurrences. This visualisation helps users understand how often certain levels of energy consumption occur. The graph is dynamically updated based on the selected date range, providing relevant and timely information.

Question 2 (874 words)

Perform Market Basket Analysis on the chosen dataset by using Apriori and FP growth algorithms. Can you express 3 similarities between these models? Address the following questions for both algorithms as mentioned below

- Determine the top 10 most frequently purchased items based on the chosen dataset.
- Use the Apriori algorithm to find frequent itemsets with a minimum support of 0.01.
- Extract the association rules with a minimum confidence of 0.5.
- Identify the top 5 association rules based on metric, lift or leverage.
- Explain the meaning of these rules in the context of the dataset.
- Provide an explanation of the top 5 association rules.

Compare the machine learning results obtained based on both algorithms and show the exact time used for the evaluation of the number of rules generated in both cases. Use any dataset that has not been used in the class, tutorials and previous assignment.

3 similarities between these models?

- The algorithms find frequent item sets and extract association rules.

- They can both identify strong relationships between items in the dataset.

- You need to set a minimum support threshold to filter item sets.

Determine the top 10 most frequently purchased items based on the chosen dataset.

Items Distribution

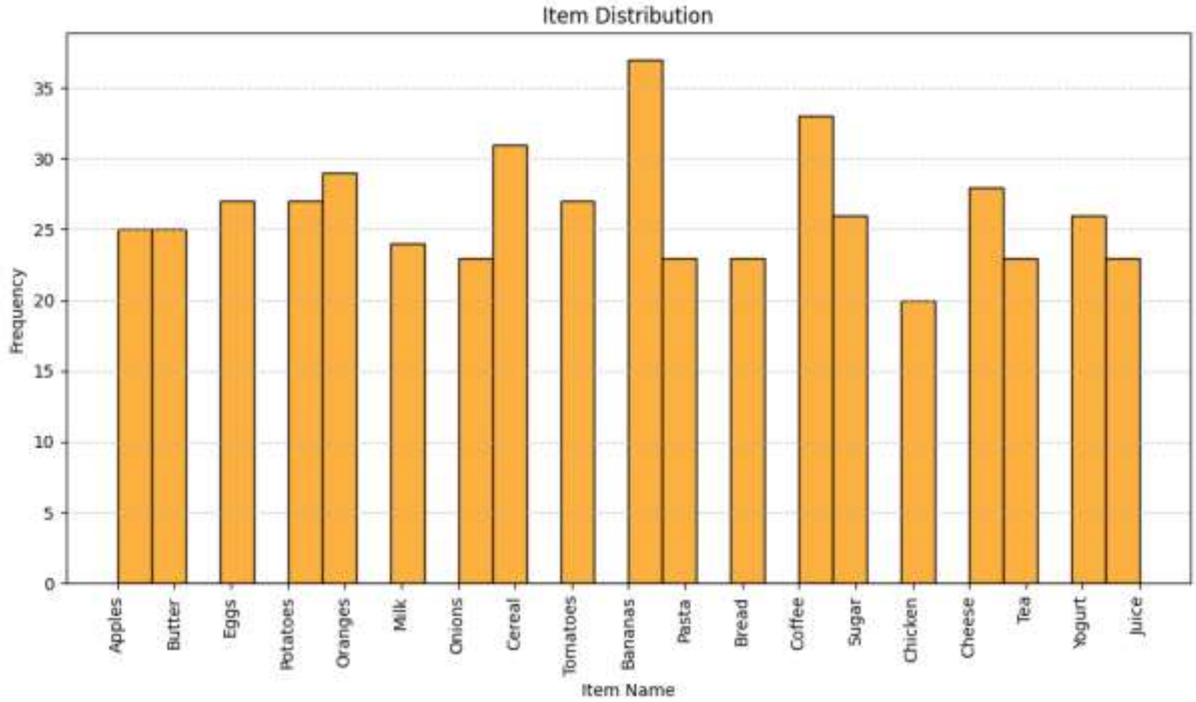
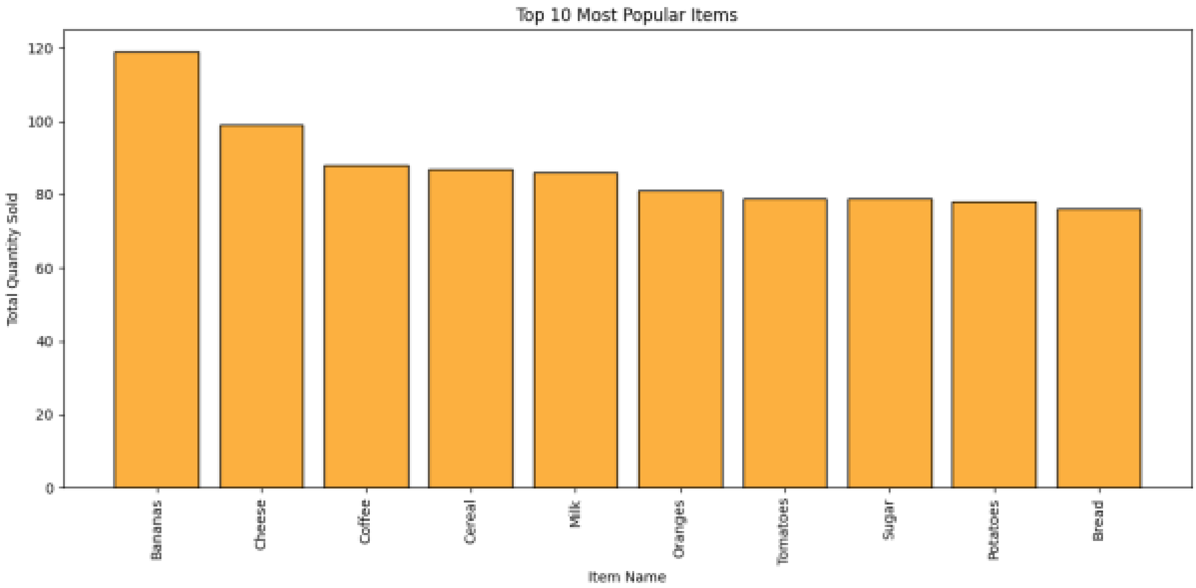


Image 9. Items Distribution.

Top 10 Most Popular Items



The top 10 items are:

- Bananas, cheese, coffee, cereal, milk, Oranges, tomatoes, sugar, potatoes, bread, and bananas are the most popular items sold at the store.

Use the Apriori algorithm to find frequent item sets with a minimum support of 0.01 | Extract the association rules with a minimum confidence of 0.5 | Identify the top 5 association rules based on metric, lift or leverage.

Top 5 rules from Apriori:

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction	zhangs_metric
4353	(Oranges, Butter, Yogurt)	(Bread, Eggs)	0.013072	0.013072	0.013072	1.0	76.5	0.012901	inf	1.0
4360	(Bread, Eggs)	(Oranges, Butter, Yogurt)	0.013072	0.013072	0.013072	1.0	76.5	0.012901	inf	1.0
5596	(Pasta, Chicken, Milk)	(Onions, Apples, Eggs)	0.013072	0.013072	0.013072	1.0	76.5	0.012901	inf	1.0
5597	(Onions, Pasta, Apples)	(Chicken, Milk, Eggs)	0.013072	0.013072	0.013072	1.0	76.5	0.012901	inf	1.0
5598	(Pasta, Apples, Eggs)	(Onions, Chicken, Milk)	0.013072	0.013072	0.013072	1.0	76.5	0.012901	inf	1.0

```
Top 5 rules from FP-Growth:
                antecedents               consequents  \
1089    (Pasta, Eggs, Potatoes)      (Onions, Apples, Milk)
1090    (Pasta, Milk, Potatoes)      (Onions, Apples, Eggs)
1093        (Pasta, Milk, Eggs)  (Onions, Apples, Potatoes)
1094  (Onions, Apples, Potatoes)         (Pasta, Milk, Eggs)
1097      (Onions, Apples, Eggs)     (Pasta, Milk, Potatoes)

      antecedent support  consequent support   support  confidence  lift  \
1089            0.013072            0.013072  0.013072         1.0  76.5
1090            0.013072            0.013072  0.013072         1.0  76.5
1093            0.013072            0.013072  0.013072         1.0  76.5
1094            0.013072            0.013072  0.013072         1.0  76.5
1097            0.013072            0.013072  0.013072         1.0  76.5

      leverage  conviction  zhangs_metric
1089  0.012901         inf            1.0
1090  0.012901         inf            1.0
1093  0.012901         inf            1.0
1094  0.012901         inf            1.0
1097  0.012901         inf            1.0
```

Top 5 Rules from Apriori Algorithm

Oranges, Butter, Yogurt => Bread, Eggs:
Customers who buy Oranges, Butter, and Yogurt also buy Bread and Eggs. This rule is always true in the data set. The high lift value of 76.5 shows these items are bought together much more often than by chance.

Bread and eggs => oranges, butter and yoghurt.
Similarly, customers who buy bread and eggs also buy oranges, butter and yoghurt. It has a confidence of 1.0 and a lift of 76.5, showing a strong association.

Pasta, chicken, milk => onions, apples, eggs:
This rule shows that customers who buy pasta, chicken, and milk also buy onions, apples, and eggs. The perfect confidence value of 1.0 and a lift of 76.5 show a strong relationship between these items.

Onions, pasta, apples => chicken, milk, eggs
Onions, pasta and apples are bought with chicken, milk and eggs. This rule also has a confidence of 1.0 and a high lift, showing a consistent pattern.

Pasta, apples, eggs => onions, chicken, milk
Customers who buy pasta, apples and eggs also buy onions, chicken and milk. This rule shows a strong association.

Top 5 Rules from FP-Growth Algorithm

Pasta, Eggs, Potatoes => Onions, Apples, Milk:
Pasta, eggs and potatoes customers also buy onions, apples and milk. The rule shows a strong correlation.

Pasta, Milk, Potatoes => Onions, Apples, Eggs:
This rule shows that customers who buy pasta, milk, and potatoes also buy onions, apples, and eggs. The perfect confidence value and lift show a strong association.

Pasta, milk, eggs => onions, apples, potatoes
Pasta, milk and eggs customers also buy onions, apples and potatoes. The confidence level is 1.0 and the lift is 76.5, which suggests a consistent purchasing pattern.

Onions, apples and potatoes lead to pasta, milk and eggs. The rule shows a strong relationship between these items.

Onions, apples, eggs => pasta, milk, potatoes
Customers who buy onions, apples and eggs also buy pasta, milk and potatoes. The perfect confidence value and high lift show a strong link.

Question 3 for Question 2

Create an interactive Dashboard aimed at older adults (60+) with specific features to summarise the most important aspects of the data and communicate through your visualisation based on the results obtained from Question 1 and Question 2.

Top 10 Most Popular Items

Image 10. Top 10 Most Popular Items.

Code explanation

```
# Preprocess the data
item_popularity = df.groupby('Itemname')['Quantity'].sum().sort_values(ascending=False)
top_10_items = item_popularity.head(10)
```

The data is grouped by item name and the total quantity for each item is calculated. The items are then sorted in descending order by total quantity and the top 10 most popular items are selected.

```
# Prepare data for association rules (dummy example, replace with actual rules from Question 2)
rules_apriori = [
    {'antecedents': 'Oranges, Butter, Yogurt', 'consequents': 'Bread, Eggs'},
    {'antecedents': 'Bread, Eggs', 'consequents': 'Oranges, Butter, Yogurt'},
    {'antecedents': 'Pasta, Chicken, Milk', 'consequents': 'Onions, Apples, Eggs'},
    {'antecedents': 'Onions, Pasta, Apples', 'consequents': 'Chicken, Milk, Eggs'},
    {'antecedents': 'Pasta, Apples, Eggs', 'consequents': 'Onions, Chicken, Milk'}
]
```

This is a dummy example of association rules. Replace them with real rules generated from your analysis.

```
# Initializing the Dash app
app = dash.Dash(__name__)
```

The Dash application instance is created.

```python
# Initializing the Dash app
app = dash.Dash(__name__)

# Define the layout
app.layout = html.Div(style={'fontFamily': 'Arial', 'padding': '20px'}, children=[
    html.H1('Market Basket Analysis Dashboard', style={'textAlign': 'center', 'color': '#FCB040'}),

    html.Label('Top 10 Most Popular Items', style={'fontSize': '20px', 'color': '#FCB040'}),
    dcc.Graph(id='top-10-items-chart'),

    html.Label('Association Rules', style={'fontSize': '20px', 'color': '#FCB040'}),
    dcc.Dropdown(
        id='association-rules-dropdown',
        options=[{'label': f"{rule['antecedents']} => {rule['consequents']}", 'value': idx} for idx, rule
        value=0,
        style={'fontSize': '18px'}
    ),
    html.Div(id='association-rule-details', style={'fontSize': '18px', 'color': '#FCB040'})
])
```

The layout of the dashboard is defined using HTML and core components:

- Title: A main title is displayed at the top, centred and coloured in #FCB040.
- Top 10 Items: A label and graph component to display the top 10 most popular items.
- Association Rules: A label, a drop-down menu to select association rules and a div to display the details of the selected rule. The dropdown is populated with the dummy association rules defined earlier.

```python
@app.callback(
    Output('top-10-items-chart', 'figure'),
    Input('association-rules-dropdown', 'value')
)
def update_top_10_items_chart(selected_rule):
    fig = px.bar(top_10_items, x=top_10_items.index, y=top_10_items.values, labels={'x': 'Item', 'y': 'Quantity'},
                 title='Top 10 Most Popular Items')
    fig.update_layout(title_font_size=24, xaxis_title='Item Name', yaxis_title='Total Quantity Sold',
                      title_font_color='#FCB040', font=dict(size=18), plot_bgcolor='white')
    return fig
```

This callback updates the Top 10 Items chart. When the dropdown value changes, it generates a bar chart using Plotly Express to display the top 10 items.

```python
@app.callback(
    Output('association-rule-details', 'children'),
    Input('association-rules-dropdown', 'value')
)
def display_association_rule_details(selected_rule):
    rule = rules_apriori[selected_rule]
    return html.Div([
        html.P(f"Antecedents: {rule['antecedents']}"),
        html.P(f"Consequents: {rule['consequents']}")
    ])
```

This callback updates the association rule details. When the dropdown value changes, it displays the antecedents and consequents of the selected rule.

```python
if __name__ == '__main__':
    app.run_server(debug=True)
```

This line starts the Dash server in debug mode, allowing live updates and easy troubleshooting during development.

Question 4 for Question 2

Explain how your dashboard is designed with the above-mentioned demographic in mind and your rationale for the types of the visualisations that you have developed in Question 3.

The interactive dashboard is specifically designed to meet the needs of older adults (60+), focusing on simplicity, clarity and ease of use.

Firstly, we have ensured that the text is clear and easy to read by using large fonts and high contrast colours. Main titles and labels are set at a font size of 20px or larger in Arial, which is simple and easy to read. High contrast colours, such as #FCB040 for titles and labels on a white background, improve visibility and help users with visual impairments to easily distinguish between different sections and elements.

The layout is designed to be straightforward and uncluttered. This minimalist approach reduces cognitive load and makes it easier for users to navigate the dashboard, especially those who may not be tech-savvy.

We used a bar chart to display the top 10 most popular items. Bar charts are intuitive and allow users to quickly compare the popularity of different items. This visualisation helps users easily identify which items are most frequently purchased, making it an effective way to communicate this information.

A drop-down menu is used to present the association rules derived from the market basket analysis. This menu provides an easy way for users to select and view specific information without being overwhelmed by too much data at once. When a rule is selected, the antecedents and consequents are displayed in plain text. This textual display ensures that the information is accessible and easy to understand.

Interactive elements such as drop-down menus and clickable options are designed to be large and easy to use, accommodating users with limited dexterity. The dashboard provides immediate visual feedback as users interact with it, helping them understand the impact of their actions and keeping them engaged.

References

dash.plotly.com. (n.d.). Dash in 20 Minutes Tutorial | Dash for Python Documentation | Plotly. [online] Available at: https://dash.plotly.com/tutorial.

plotly (n.d.). Plotly Express. [online] plotly.com. Available at: https://plotly.com/python/plotly-express/.

Kharwal, A. (2023). Market Basket Analysis: Case Study. [online] Statso. Available at: https://statso.io/market-basket-analysis-case-study/ [Accessed 19 Jul. 2024].

Stack Overflow. (n.d.). Efficient algorithms to perform Market Basket Analysis. [online] Available at: https://stackoverflow.com/questions/73595640/efficient-algorithms-to-perform-market-basket-analysis [Accessed 12 Jul. 2024].

Shruthi Gurudath (2020). Market Basket Analysis & Recommendation System Using Association Rules. [online] ResearchGate. Available at: https://www.researchgate.net/publication/343484851_Market_Basket_Analysis_Recommendation_System_Using_Association_Rules

Brownlee, J. (2019). Time Series Prediction with LSTM Recurrent Neural Networks in Python with Keras. [online] Machine Learning Mastery. Available at: https://machinelearningmastery.com/time-series-prediction-lstm-recurrent-neural-networks-python-keras/.

Investopedia. (n.d.). Autoregressive Integrated Moving Average (ARIMA) Prediction Model. [online] Available at: https://www.investopedia.com/terms/a/autoregressive-integrated-moving-average-arima.asp#:%7E:text=An%20autoregressive%20integrated%20moving%20average%2C%20or%20ARIMA%2C%20is%20a%20statistical [Accessed 28 Jul. 2024].