EXPERNETIC

Machine Learning Internship– Assignment

AURELIA PERERA

CONTENTS

1.0 INTRODUCTION

Task 01

This project explores supermarket transaction data to uncover meaningful business insights through machine learning techniques. Using Python and relevant data science libraries, we clean, prepare, and analyze sales and promotion data from multiple supermarket locations to develop predictive models that aid strategic planning.

Task 02

This project demonstrates how an artificial intelligence (AI) agent can learn to navigate a maze using Q-Learning, a type of reinforcement learning algorithm. The goal is for the agent to autonomously find the optimal path from a starting point to a goal, learning from its environment through rewards and penalties.

2.0 Task 01 – Using Machine Learning models to Implement Business insights.

2.1 OBJECTIVES

Data Cleaning and Preparation: Ensure all datasets are free of missing values, inconsistencies, and irrelevant entries, and prepare them for analysis.

Exploratory Data Analysis: Gain initial insights into the dataset structure, data distributions, and relationships between variables.

Machine Learning Applications:

- Forecast weekly revenue using a time-series **ARIMA model**.
- Evaluate the impact of promotions and displays on product sales using **classification and regression techniques**.

2.2 Methodology

1. **Data Loading and Cleaning**

- Datasets used: Sales, Promotion, Supermarkets, and Items.
- Duplicates and null values were identified and removed.
- Standardized and transformed features (e.g., product size, time format).
- Extracted temporal features such as day of the week, weekend flags, and month.
- Merged datasets based on shared keys such as code, supermarket, and week.

2. Feature Engineering

Created new variables:

- Revenue = amount × units
- Binary flags: Is_Featured and Is_Endcap for promotion characteristics.
- Time Feature

Cleaned and standardized item sizes (e.g., converting irregular formats to oz, lb).

3. Time Series Forecasting (ARIMA Model)

- Aggregated revenue by week.
- Performed stationarity testing using Augmented Dickey-Fuller (ADF).
- Applied differencing where necessary and used the ARIMA model to forecast future weekly revenue.

4. Promotion Impact Analysis

Combined promotional features with sales data.

Built a machine learning model to predict sales volume based on promotional attributes.

Evaluated multiple models and selected the best-performing one.

2.3 BUSINESS INSIGHTS

1. Revenue Forecasting

By applying the ARIMA model to weekly revenue data, the supermarket can predict future income trends. This enables better budgeting, inventory planning, and staffing schedules for high-demand periods.

2. Promotion Effectiveness

The predictive model revealed that both being featured and displayed on an end cap significantly increase the likelihood of higher sales volume. Businesses can leverage this insight to maximize the ROI of promotional efforts by strategically placing high-margin or seasonal products in these positions.

2.4 CONCLUSION

This project demonstrates the value of integrating data cleaning, feature engineering, and machine learning techniques to generate actionable business insights. Forecasting revenue and evaluating promotional impact provide supermarkets with a robust foundation for making strategic marketing and inventory decisions.

3.0 Task 02 - Navigating a maze using Reinforcement learning

3.1 OBJECTIVES

- **Design a Maze Environment**: Create a virtual 2D maze using a grid format where the agent can navigate. This environment includes walkable paths and obstacles (walls).
- **Define Action Space and Movement Rules**: Specify the directions the agent can move — up, down, left, and right — and apply constraints to prevent it from walking into walls or outside the maze boundaries.
- **Implement Q-Learning Algorithm**:
  - **Reward Structure**: Assign positive rewards for reaching the goal and penalties for invalid moves or hitting walls.
  - **Epsilon-Greedy Strategy**: Use a method that allows the agent to randomly explore new paths (exploration) while also choosing the best-known actions (exploitation).
  - **Q-Value Updates**: Update the Q-table using the Bellman equation, allowing the agent to learn the expected value of taking a specific action from a given state.
- **Train the Agent**: Allow the agent to run through the maze repeatedly, learning from its experiences by adjusting Q-values based on the outcomes of its actions.
- **Visualize and Evaluate Performance**: Graphically show how the agent improves over time and analyze the final path it takes to determine if it has learned an optimal strategy.

3.2 METHODOLOGY

- **Environment Setup**: A 2D maze is represented using a NumPy array, where 0 denotes a walkable path and 1 a wall.

- **Actions**: The agent can move in four directions — up, down, left, and right.
- **Q-Learning**:
    - The agent starts with a Q-table initialized with zeros.
    - It explores the environment using an **epsilon-greedy** policy — balancing exploration and exploitation.
    - Q-values are updated using the Bellman equation after each action.
- **Training**: The agent undergoes multiple episodes, progressively learning better routes based on accumulated experience.
- **Visualization**: Training progress and the final optimal path are visualized using matplotlib to assess learning effectiveness.

3.3 CONCLUTION

Through this project, we demonstrate the power of reinforcement learning in solving complex pathfinding problems. The AI agent successfully learns to traverse a maze, building an optimal policy through trial and error. This experiment provides foundational insight into the practical use of Q-Learning for autonomous decision-making systems and robotics.