**DEPI**

**IBM-Data science professional certificate**

**Final project**

**Shirt sales forecasting and optimization**

**Submitted by: Mohamed Adel Hamzawi**

**Peter Ramzy**

**Ahmed Ashraf**

**Mina Samy**

**Ahmed Saied**

**Submitted to: Eng. Islam Adel**

## Project Overview

The goal of this project is to build a sales forecasting model using historical sales data to identify trends, seasonality, and patterns, and ultimately provide future forecasts that can be used for optimization. The project is divided into multiple phases, from data exploration to model deployment.

**Data Collection and Exploration**

**Tasks Completed:**

1. **Data Collection:**
   * The dataset comprises three files:
     + **Orders**: Information about each shirt order, including the date, quantity, and store location.
     + **Customers**: Details about customers, including customer ID and demographics.
     + **Products**: Information on shirt products, such as product ID, type, and price.
   * These files were merged into a single dataset using customer ID and product ID as key fields.
2. **Exploratory Data Analysis (EDA):**
   * Conducted an initial exploration of the dataset to understand its structure and assess data quality.
   * Identified key trends in shirt sales, including:
     + Seasonal peaks during holidays.
     + Variation in sales across different shirt types and customer demographics.
   * Investigated anomalies and missing values that needed cleaning.
3. **Tools Used:**
   * **Libraries**: NumPy and Pandas were used for data manipulation, while Matplotlib was used for visualizing trends.

**Deliverables:**

* **Merged Dataset**: A single, clean dataset combining orders, customers, and products.
* **EDA Report**: Detailed initial findings on sales patterns, customer behavior, and product performance.
* **EDA Notebook**: Includes visualizations (sales trends, customer demographics) and basic statistical analysis.

**Data Cleaning and Visualization**

**Tasks Completed:**

1. **Data Cleaning:**
   * Handled missing values by applying imputation where necessary (e.g., forward fill for missing sales data).
   * Detected and removed outliers in the dataset using statistical methods (e.g., z-scores).
   * Reformatted the date column and ensured the dataset was prepared for time series modeling.
2. **Data Analysis:**
   * Analyzed sales seasonality, customer demographics, and product types.
   * Key insights:
     + Higher sales observed during specific promotions.
     + Different shirt styles had varying levels of demand depending on customer groups.
   * Performed the **Augmented Dickey-Fuller (ADF)** test to check stationarity and determined that the dataset required differencing.
3. **Data Visualization:**
   * Created several visualizations to explore sales trends and patterns:
     + Line plots to illustrate time-based sales trends.
     + Bar charts showing the popularity of different shirt types.
     + Customer segmentation analysis based on demographics and purchase behavior.
4. **Tools Used:**
   * **Libraries**: NumPy, Pandas, Matplotlib, and Seaborn for data cleaning and visualization.

**Deliverables:**

* **Cleaned Dataset**: Preprocessed and ready for time series analysis.
* **Data Analysis Report**: Analyzed sales patterns, seasonality, and customer preferences.
* **Visualizations**: Graphs depicting sales trends, product popularity, and customer segmentation.

**Forecasting Model Development**

**Tasks Completed:**

1. **Model Selection:**
   * Decided to use the **ARIMA (AutoRegressive Integrated Moving Average)** model to forecast sales for the next two months based on the past data.
   * The ARIMA model was chosen for its ability to handle non-stationary time series and account for sales trends and seasonality.
2. **Model Training:**
   * Trained an ARIMA model with parameters p=1p = 1p=1, d=1d = 1d=1, and q=1q = 1q=1 after conducting grid search and analyzing the dataset's stationarity.
   * The model was trained on historical shirt sales data, and the results showed promising accuracy in predicting short-term future sales.
   * **Metrics** such as Mean Squared Error (MSE) were logged to assess performance.
3. **Model Optimization:**
   * Fine-tuned the ARIMA model by testing different parameter configurations, optimizing for better predictive performance.
   * Logged the model's **R-squared** and **MSE** metrics to track how well the model fits the data.
4. **MLflow Integration:**
   * Used **MLflow** to log important parameters such as ppp, ddd, and qqq, as well as model performance metrics (e.g., MSE).
   * Tracked the experiment details and model artifacts using MLflow, making it easier to compare different models and versions.
5. **Tools Used:**
   * **Libraries**: Statsmodels for ARIMA model training, MLflow for tracking model parameters and performance, and Matplotlib for visualizations.

**Deliverables:**

* **Forecasting Model Report**: Documented the ARIMA model performance and provided insights into the upcoming two months of shirt sales forecasts.
* **Python Code**: Shared code for ARIMA model training, evaluation, and parameter logging using MLflow.
* **MLflow Tracking**: Logged parameters, metrics, and model artifacts for tracking and comparison purposes.

**MLOps, Deployment, and Final Presentation**

**Tasks Completed:**

1. **MLOps Implementation:**
   * Integrated **MLflow** to track and manage multiple versions of the forecasting model.
   * Logged model parameters, metrics, and artifacts for version control and easy access for future comparisons.
   * Utilized **MLflow’s tracking server** to monitor experiment results over time and ensure reproducibility.
2. **Model Deployment:**
   * Deployed the ARIMA forecasting model using **Streamlit**, a lightweight web application framework, allowing users to interact with the model.
   * Built an interface where users can input new sales data (such as recent shirt orders) and generate real-time sales forecasts.
   * Incorporated a feature where forecasts are visualized, making it user-friendly for stakeholders to understand and act upon predictions.
   * Deployed the application locally with plans to push it to the cloud (e.g., AWS or Heroku) for broader access.
3. **Final Report and Presentation:**
   * Compiled a final project report summarizing the workflow:
     + Data collection and merging.
     + Data exploration and analysis.
     + ARIMA model selection and training.
     + Forecasting results for the next two months.
     + Model performance metrics (e.g., MSE, R-squared) and visualization of results.
   * Created a **Streamlit dashboard demo** as part of the final presentation, showcasing:
     + The forecasting model’s predictions.
     + Interactive visualizations and parameter adjustment options for users to experiment with.
4. **Tools Used:**
   * **MLflow**: Used for managing experiment tracking and logging model parameters and metrics.
   * **Streamlit**: Created a web application to deploy the forecasting model.
   * **Python Libraries**: Statsmodels for ARIMA, Pandas and NumPy for data manipulation, Matplotlib for visualizations.

**Conclusion**

Over the course of this project, we successfully implemented a comprehensive sales forecasting system for a shirt orders dataset, utilizing a structured approach to data exploration, model development, and deployment. Starting with the collection and merging of orders, customers, and products data, we conducted in-depth exploratory data analysis to uncover key sales trends and customer behavior patterns. The dataset was thoroughly cleaned and visualized, providing a solid foundation for model development.

For forecasting, we selected and optimized the **ARIMA** model to predict future shirt sales. Through model tuning and validation, we ensured accurate forecasts, which were tracked and managed using **MLflow** to log all parameters, metrics, and results. The project culminated in the deployment of a user-friendly web application using **Streamlit**, allowing real-time sales predictions based on newly input data.

The deliverables include a cleaned dataset, insightful visualizations, an optimized forecasting model, and an interactive dashboard for stakeholders to make informed, data-driven decisions. This project demonstrates the power of time series forecasting in helping businesses optimize inventory, sales, and marketing strategies by providing accurate predictions of future demand.

With this forecasting model in place, the business can proactively plan for future sales trends, ensuring they are well-prepared for fluctuations in demand, seasonality, and customer preferences. Overall, the project’s success highlights the importance of data-driven decision-making in driving operational efficiency and improving overall business performance.