**Mini Project Report Portfolio**

**Irfanullah Shinwary**

**Batch 3**

# **Mini Project (1): Simple Chatbot**

* **Name:** Irfanullah Shinway
* **Supervisor:** Zainullah Ziarmal
* **Course Title:** AI and Data Science
* **Submission Date:**

## **Objective**

The aim of this project is to create a basic chatbot that responds to predefined user inputs. It serves as a hands-on exercise to practice essential programming concepts, such as working with dictionaries, loops, and handling user interactions.

## **Skills and Concepts Highlighted**

* **Python Basics:** Showcases the use of loops, conditionals, and data structures like dictionaries.
* **Natural Language Processing (NLP):** Introduces a simple logic to enable text-based interaction.
* **User Interaction:** Improves understanding of processing user inputs and delivering relevant responses.

## **Project Workflow**

**1. Predefined Questions and Responses**

* A dictionary is used to store predefined questions as keys and their corresponding responses as values, demonstrating Python’s ability to handle structured data efficiently.

**2. Conversation Loop**

* The chatbot operates in a loop, continuously interacting with the user until they decide to exit.

**3. Randomized Responses**

* For each input, the chatbot selects a response randomly from predefined options to add variety.

**4. Input Validation**

* The program checks if the user’s input matches any of the predefined keys. If it doesn’t, a friendly message prompts the user to try again or provides generic responses.

## **Real-World Applications**

This basic chatbot demonstrates the first steps toward building conversational AI systems. These systems can be applied in:

* Customer support to handle common inquiries.
* Personal assistant tools for quick tasks.
* Educational tools for engaging learning experiences.

## **Features**

1. **Interactive Interface**
   * Users are guided with a list of predefined questions, making the interaction smooth and straightforward.
2. **Graceful Error Handling**
   * If the user enters an unrecognized question, the chatbot politely requests a valid input or provides a general response.
3. **Simple Expandability**
   * New questions and responses can be added easily to increase the chatbot’s scope.

**Code HighlightsA screenshot of a computer program

Description automatically generated**

**A screenshot of a computer program

Description automatically generated**

**Sample Interaction:**

Welcome to ChatBot!

You can chat with me. Try asking:

- Hi

- How are you?

- Are you working?

- What is your name?

- What did you do yesterday?

- Quit

You: hi

ChatBot: Hey! Nice to chat with you!

You: what is your name?

ChatBot: I’m ChatBot, your virtual assistant!

You: how are you?

ChatBot: Doing well, thank you! What about you?

You: what

ChatBot: Hmm, I haven't heard that before. What does it mean?

You: quit

ChatBot: See you later! Take care!

**Results**

The chatbot effectively:

* Engages users in simple, predefined conversations.
* Provides accurate responses to the listed questions.
* Ends the session when the user opts to exit.

# **Mini Project (2): Titanic Dataset Analysis**

* **Name:** **Irfanullah Shinway**
* **ID:**
* **Supervisor:** Zainullah Ziarmal
* **Course Title:** AI and Data Science
* **Submission Date:**

**Objective**

The aim of this project is to analyze the Titanic dataset to extract meaningful insights, such as survival patterns and relationships between features. This project demonstrates essential skills in data preprocessing, exploratory data analysis, and visualization.

**Skills and Concepts Highlighted**

* **Data Cleaning:** Handling missing data effectively using pandas.
* **Feature Engineering:** Creating meaningful new features to enhance the dataset.
* **Data Visualization:** Using Matplotlib to visualize trends and survival rates.
* **Exploratory Data Analysis (EDA):** Identifying patterns and relationships within the dataset.

**Project Workflow**

**1. Loading and Initial Exploration**

* Imported the Titanic dataset using pandas and displayed initial summary statistics to understand its structure.

**2. Handling Missing Values**

* **'Age':** Filled missing values with the median age to retain numerical integrity.
* **'Embarked':** Filled missing values with the most common embarkation point.
* **'Cabin':** Dropped due to a large proportion of missing values (687 out of 891 rows).

**3. Feature Engineering**

* **FamilySize:** Created a new feature combining 'SibSp' (siblings/spouses) and 'Parch' (parents/children) to represent total family size onboard.
* **AgeCategory:** Categorized passengers into groups: *Child, Teen, Adult, Middle Age, Senior*.

**4. Filtering and Grouping**

* Filtered data to extract subsets:
  + Female passengers in first class.
  + Adult passengers based on the 'AgeCategory' column.
  + Passengers who survived.

**5. Visualization**

* Visualized survival rates by gender, age categories, and age distribution among passengers.

**Real-World Applications**

This analysis approach can be applied to:

* **Data-Driven Decision Making:** Understanding patterns in survival to improve safety procedures.
* **Training Predictive Models:** Developing machine learning models using cleaned and engineered features.
* **EDA Techniques:** Building a foundation for more advanced analytics in other datasets.

**Features**

1. **Data Cleaning and Imputation**
   * Addressed missing data for 'Age' and 'Embarked' while maintaining data integrity.
2. **New Feature Creation**
   * Introduced 'FamilySize' and 'AgeCategory' to enhance dataset usability.
3. **Effective Visualization**
   * Showcased trends and survival patterns visually for better interpretability.

**Code Highlights**A screenshot of a computer program

Description automatically generated

A screen shot of a computer

Description automatically generatedA screenshot of a computer screen

Description automatically generatedA screenshot of a computer screen

Description automatically generated

**Visualizations:**

1. **Survival Rate by Gender**
   * Bar chart showing higher survival rates for females.
2. **Age Distribution**
   * Histogram illustrating passenger age distribution.
3. **Survival Rate by Age Category**
   * Bar chart highlighting survival rates across different age groups.

**Results**

* Missing values in 'Age' and 'Embarked' columns were successfully handled.
* New features 'FamilySize' and 'AgeCategory' were introduced to enhance insights.
* Survival analysis revealed significant patterns based on gender and age.

**Future Work**

1. **Predictive Modeling:**
   * Develop a machine learning model to predict passenger survival based on the processed dataset.
2. **Advanced Visualization:**
   * Explore interactive visualizations using libraries like Seaborn or Plotly.
3. **Feature Enhancement:**
   * Create additional features such as 'FareCategory' or analyze ticket types for more insights.
4. **Incorporate External Data:**
   * Use supplementary datasets to enrich the analysis, such as passenger demographics or ship conditions.

# **Mini Project (3): Real-Time Stock Price Dashboard**

* **Name:** **Irfanullah Shinway**
* **ID:**
* **Supervisor:** Zainullah Ziarmal
* **Course Title:** AI and Data Science
* **Submission Date:**

**Objective**

The goal of this project is to create a real-time dashboard for visualizing stock price trends. The dashboard updates dynamically and allows users to select between popular stocks such as Apple, Microsoft, and Amazon.

**Skills and Concepts Highlighted**

* **Interactive Dashboards:** Utilized Dash framework to build interactive web applications.
* **Data Visualization:** Integrated Plotly for generating live and dynamic plots.
* **API Integration:** Used Yahoo Finance API to fetch stock data in real-time.
* **Event Handling:** Managed updates using Dash’s callback mechanism and interval components.

**Project Workflow**

**1. Framework Setup**

* Dash framework was used for building the dashboard, integrating HTML elements with interactivity and Plotly visualizations.

**2. Data Fetching**

* Implemented a function to fetch real-time stock data using the yfinance library, querying the last day's data at 1-minute intervals.

**3. Dynamic Plot Updates**

* Designed the dashboard to refresh every 10 seconds using Dash's dcc.Interval component.
* Used Dash callbacks to bind user selection and refresh intervals to the live graph.

**4. User Interactivity**

* Added a dropdown menu for users to select between stock options (Apple, Microsoft, Amazon).

**Real-World Applications**

1. **Financial Analysis:**
   * Enables investors to track stock performance in real time.
2. **Dynamic Reporting:**
   * Offers an interactive way to monitor and report market trends.
3. **Adaptable Framework:**
   * The design can be extended to include multiple stocks, indices, or additional financial metrics.

**Features**

1. **Real-Time Data Refresh**
   * Fetches and visualizes stock prices every 10 seconds.
2. **Dynamic User Interaction**
   * Dropdown menu for selecting stocks dynamically updates the chart.
3. **Scalable Design**
   * Modular structure allows easy integration of additional stocks and features.

**Code Highlights**

A screenshot of a computer program

Description automatically generatedA screenshot of a computer

Description automatically generated

**Output**

1. **Dashboard Layout**
   * Shows a dropdown for stock selection and a real-time updating graph.
2. **Live Graph**
   * Visualizes stock price trends dynamically based on user selection.

**Results**

* Successfully developed a real-time stock price dashboard.
* Incorporated interactive features for stock selection and live data updates.
* Enhanced understanding of Dash, Plotly, and API integration.

**Future Work**

1. **Expanded Stock Options:**
   * Add more stock options and categories for a broader scope of analysis.
2. **Multi-Stock Comparison:**
   * Allow users to visualize and compare multiple stocks simultaneously.
3. **Advanced Analytics:**
   * Incorporate additional financial metrics like moving averages, volume analysis, and Bollinger Bands.
4. **Deploy to Web:**
   * Host the application on a cloud platform such as Heroku or AWS for public access.
5. **Historical Data Analysis:**
   * Enable users to view historical trends alongside real-time updates.

# **Mini Project (4): Loan Approval Prediction**

* **Name:** Irfanullah Shinway
* **ID:**
* **Supervisor:** Zainullah Ziarmal
* **Course Title:** AI and Data Science
* **Submission Date:**

**Objective**

The goal of this project is to predict loan approval status based on applicant information using machine learning models. The project involves preprocessing, feature engineering, and evaluating multiple classifiers to identify the best performing model.

**Skills and Concepts Highlighted**

* **Data Cleaning and Preprocessing:** Handled missing values and encoded categorical features.
* **Feature Selection and Engineering:** Identified key features relevant for loan status prediction.
* **Supervised Learning Models:** Trained and evaluated Logistic Regression, Decision Tree, and Random Forest models.
* **Model Evaluation:** Compared models using metrics such as accuracy, precision, recall, and F1-score.
* **Visualization:** Used heatmaps and bar charts to analyze performance and confusion matrices.

**Project Workflow**

**1. Data Exploration**

* Loaded and inspected the dataset, noting missing values, data types, and distribution of key features.
* The dataset consisted of 614 records with 13 attributes.

**2. Data Cleaning**

* Filled missing values using appropriate methods:
  + **Mode** for categorical columns (e.g., Gender, Married).
  + **Median** for numerical columns (e.g., LoanAmount).

**3. Data Encoding**

* Used LabelEncoder to convert categorical variables into numerical format for model compatibility.

**4. Feature and Target Selection**

* Defined the features (X) by excluding Loan\_ID and Loan\_Status.
* Set Loan\_Status as the target variable (y).

**5. Model Implementation and Training**

* Split data into training (80%) and testing (20%) sets.
* Trained the following models:
  + **Logistic Regression**
  + **Decision Tree Classifier**
  + **Random Forest Classifier**

**6. Model Evaluation**

* Evaluated models on the test set using metrics like accuracy, precision, recall, and F1-score.
* Compared model performances and visualized results.

**7. Visualization**

* Created bar plots for model accuracy comparison.
* Visualized confusion matrices for detailed performance analysis.

**Real-World Applications**

1. **Loan Eligibility Prediction:**
   * Helps banks and financial institutions automate the loan approval process.
2. **Risk Assessment:**
   * Assists in identifying high-risk applicants based on historical data.
3. **Decision Support Systems:**
   * Provides insights for decision-making in financial services.

**Features**

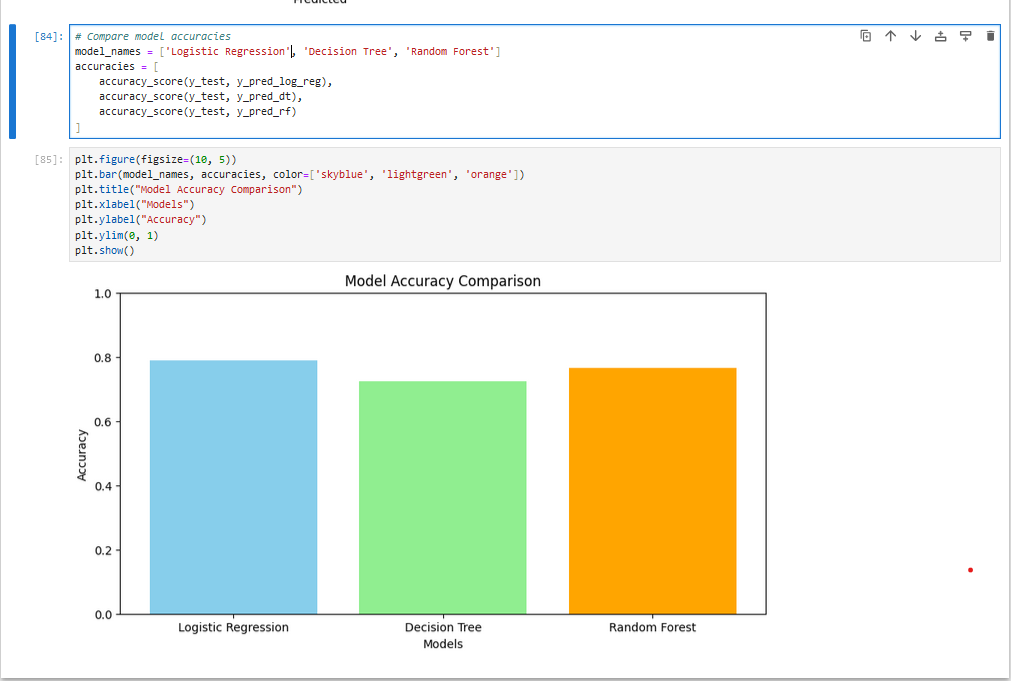
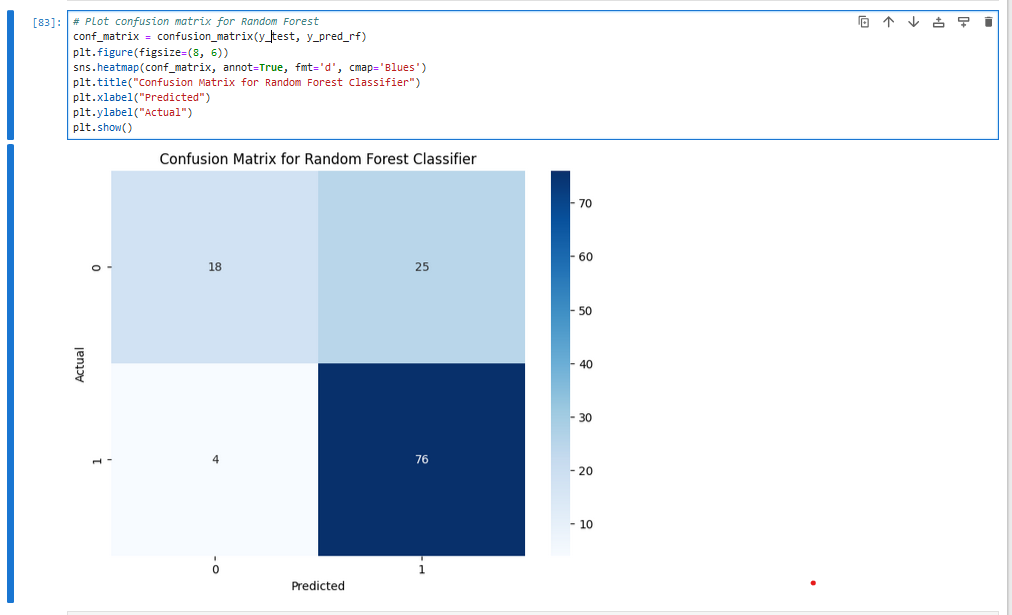
1. **Data Cleaning and Handling Missing Values**
   * Ensured the dataset was ready for modeling by addressing missing values effectively.
2. **Multiple Classifier Evaluation**
   * Compared Logistic Regression, Decision Tree, and Random Forest to identify the most accurate model.
3. **Comprehensive Visualization**
   * Used heatmaps and bar plots to convey model performance and confusion matrices.

**Code Highlights**

**A screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generated**

**Results**

1. **Model Performance Comparison**
   * **Logistic Regression:** Accuracy = 78.86%
   * **Decision Tree:** Accuracy = 72.36%
   * **Random Forest:** Accuracy = 76.42%
2. **Insights**
   * Logistic Regression showed the highest accuracy and balanced performance.
   * Random Forest outperformed Decision Tree due to ensemble averaging, reducing overfitting.

**Future Work**

1. **Hyperparameter Tuning:**
   * Optimize model parameters for better accuracy.
2. **Feature Engineering:**
   * Explore additional features and transformations for improved predictive power.
3. **Model Deployment:**
   * Deploy the model using Flask or Django for real-world usability.
4. **Integration with APIs:**
   * Enable real-time prediction through API integration.

# **Mini Project (5): Customer Segmentation Using K-Means Clustering**

* **Name:** **Irfanullah Shinway**
* **ID:**
* **Supervisor:** Zainullah Ziarmal
* **Course Title:** AI and Data Science
* **Submission Date:**

**Objective**

The objective of this project is to perform customer segmentation based on purchasing patterns using the K-Means clustering algorithm. The analysis helps in understanding customer behaviors and identifying actionable insights for targeted marketing.

**Skills and Concepts Highlighted**

* **Data Preprocessing and Cleaning:** Handled duplicates, missing values, and normalized data.
* **Feature Engineering:** Selected relevant features (Quantity, UnitPrice) for clustering.
* **Unsupervised Learning:** Applied K-Means clustering and evaluated the optimal number of clusters using the Elbow method.
* **Data Visualization:** Used scatter plots and bar charts to represent clustering outcomes.

**Project Workflow**

**1. Data Exploration**

* Loaded the dataset from an Excel file, converted it into a CSV format for easier handling.
* The dataset contained **541,909 rows and 8 columns**, including transaction and customer details.

**2. Data Cleaning**

* **Removed duplicates:** Eliminated 5,268 duplicate rows.
* **Handled missing values:**
  + Dropped rows with missing Description values (1,454 rows).
  + Retained rows without CustomerID to focus on transactional patterns.

**3. Data Transformation**

* Converted InvoiceDate to datetime format for potential time-series analysis.
* Selected Quantity and UnitPrice as clustering features and scaled them using **MinMaxScaler** to normalize the data.

**4. Clustering Analysis**

* **Elbow Method:** Determined the optimal number of clusters using the Elbow curve by plotting inertia (sum of squared distances) for k values from 1 to 10.
* **K-Means Clustering:** Applied K-Means with k=3 clusters (optimal value) to the normalized data.

**5. Visualization**

* **Cluster Visualization:** Created a scatter plot of data points, colored by cluster labels.
* **Cluster Centers:** Visualized the cluster centers in a bar chart to understand feature significance.

**Real-World Applications**

1. **Customer Segmentation:**
   * Classifies customers into distinct groups for targeted marketing and promotions.
2. **Product Insights:**
   * Helps in identifying patterns in purchase volume and pricing preferences.
3. **Business Strategy:**
   * Assists in designing personalized loyalty programs and inventory planning.

**Features**

1. **Data Cleaning and Preprocessing:**
   * Removed duplicates and normalized the dataset for clustering.
2. **K-Means Algorithm Implementation:**
   * Determined the optimal number of clusters and assigned cluster labels to data points.
3. **Insightful Visualizations:**
   * Scatter plots for cluster patterns and bar charts for cluster center comparison.

**Code HighlightsA screen shot of a computer

Description automatically generated**

**A screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generated**

## **Results**

1. **Optimal Clusters:**
   * Determined 3 clusters using the Elbow Method.
2. **Cluster Characteristics:**
   * **Cluster 0:** High quantity, low unit price (bulk buyers).
   * **Cluster 1:** Low quantity, moderate unit price (individual buyers).
   * **Cluster 2:** Moderate quantity, high unit price (premium customers).
3. **Visualization Outcomes:**
   * Scatter plot revealed distinct clustering patterns.
   * Bar chart of cluster centers highlighted feature significance.

## **Future Work**

1. **Feature Expansion:**
   * Include additional features such as purchase time or product categories.
2. **Dynamic Clustering:**
   * Explore hierarchical clustering or DBSCAN for non-linear data patterns.
3. **Integration:**
   * Develop a dashboard to visualize clustering results interactively.
4. **Customer Recommendations:**
   * Build a recommendation system based on segmented clusters.

# **Mini Project (6): Handwritten Digit Recognition using Neural Networks**

* **Name:** Irfanullah Shinway
* **ID:**
* **Supervisor:** Zainullah Ziarmal
* **Course Title:** AI and Data Science
* **Submission Date:**

## **Objective**

The objective of this project is to develop a machine learning model capable of recognizing handwritten digits using the MNIST dataset. The project employs both a Feedforward Neural Network (MLP) and a Convolutional Neural Network (CNN) to compare their performance on digit classification.

## **Skills and Concepts Highlighted**

* **Deep Learning Models:** Developed and trained both MLP and CNN models using TensorFlow/Keras.
* **Image Preprocessing:** Normalized pixel values and reshaped data for CNN input compatibility.
* **Model Evaluation:** Analyzed accuracy and loss metrics during training and testing phases.
* **Visualization:** Displayed model performance and predictions for qualitative assessment.

## **Project Workflow**

**1. Dataset Overview**

* **Dataset:** MNIST dataset containing 70,000 grayscale images of handwritten digits (28x28 pixels).
* **Training Data:** 60,000 images and labels.
* **Testing Data:** 10,000 images and labels.

**2. Data Preprocessing**

* Normalized pixel values to a range of 0 to 1 for faster convergence during training.
* Converted labels to one-hot encoding for categorical cross-entropy loss computation.
* Reshaped images to include a channel dimension (required for CNN).

**3. Model Development**

* **Feedforward Neural Network (MLP):**
  + Flattened input images to 1D arrays.
  + Designed a sequential model with two hidden layers of 128 and 64 neurons using ReLU activation.
  + Output layer with 10 neurons (softmax activation) for multi-class classification.
* **Convolutional Neural Network (CNN):**
  + Added two convolutional layers with 32 and 64 filters, followed by max-pooling layers.
  + Flattened output and passed it through dense layers for classification.
  + Output layer with 10 neurons for digit classification.

**4. Model Training**

* Optimized models using the Adam optimizer.
* Loss function: Categorical Cross-Entropy.
* Metrics: Accuracy.
* Trained models for 10 epochs with a batch size of 32, monitoring validation performance.

**5. Model Evaluation**

* Evaluated test accuracy and loss for both models.
* Visualized training and validation accuracy/loss over epochs.

**6. Prediction Visualization**

* Predicted digits on test images and compared with true labels.
* Visualized a subset of predictions alongside the corresponding images.

## **Real-World Applications**

1. **Handwritten Digit Recognition:**
   * Used in postal systems for automated zip code reading.
2. **Digit Classification in Finance:**
   * Automates data entry tasks, such as check processing in banking systems.
3. **Pattern Recognition and Image Analysis:**
   * Forms a foundation for more complex computer vision tasks like optical character recognition (OCR).

## **Features**

1. **Deep Learning Model Implementation:**
   * Compared MLP and CNN architectures to identify the superior model for image classification.
2. **Data Preprocessing:**
   * Prepared raw MNIST data for model training and evaluation.
3. **Visualization:**
   * Analyzed performance through epoch-wise accuracy/loss plots and predicted image outputs.

## **Code Highlights**

**A screenshot of a computer

Description automatically generatedA screen shot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generatedA screen shot of a graph

Description automatically generatedA screen shot of a graph

Description automatically generatedA screenshot of a computer

Description automatically generatedA white number on a black background

Description automatically generated**

## **Results**

1. **Test Accuracy:**
   * CNN achieved **99.16% accuracy** on the test set.
2. **Training Insights:**
   * Validation accuracy consistently improved over epochs, indicating effective learning.
   * CNN outperformed MLP, demonstrating the effectiveness of convolutional layers in image-based tasks.

## **Future Work**

1. **Data Augmentation:**
   * Enhance generalization by augmenting training data with transformations like rotation, scaling, and flipping.
2. **Hyperparameter Tuning:**
   * Experiment with filter sizes, layer configurations, and learning rates to optimize performance.
3. **Transfer Learning:**
   * Utilize pre-trained models like VGG16 or ResNet for better feature extraction.
4. **Deployment:**
   * Deploy the trained model as a web service for real-time digit recognition.

# **Mini Project (7): Time Series Forecasting for Retail Sales using ARIMA**

* **Name:** Irfanullah Shinway
* **ID:**
* **Supervisor:** Zainullah Ziarmal
* **Course Title:** AI and Data Science
* **Submission Date:**

## **Objective**

This project focuses on time series forecasting for retail sales using historical data of various categories from the US retail sector. The goal is to apply ARIMA (AutoRegressive Integrated Moving Average) models to predict future sales, focusing on the "Clothing" category.

## **Skills and Concepts Highlighted**

* **Time Series Forecasting:** Utilized ARIMA model for time series forecasting.
* **Data Preprocessing:** Processed and visualized retail sales data for model training.
* **Model Evaluation:** Evaluated the model using RMSE (Root Mean Squared Error) and MAE (Mean Absolute Error).
* **Data Visualization:** Plotted sales data and forecast to compare actual vs. predicted values.

## **Project Workflow**

**1. Dataset Overview**

The dataset us-retail-sales.csv contains monthly sales data for six retail categories from January 1992 to December 2008. The columns include:

* **Month:** Date of sale.
* **Clothing, Appliances, FoodAndBeverage, Automobiles, GeneralMerchandise, BuildingMaterials:** Monthly sales for each category.

**2. Data Exploration and Preprocessing**

* Loaded the dataset using pandas and examined its structure.
* Converted the 'Month' column to a datetime format and set it as the index for time series analysis.
* Described the dataset to understand sales distribution and trends.

**3. Visualization**

* Visualized one of the time series (Clothing) to understand its trends over time.

**4. Model Development**

* **ARIMA Model:**
  + Used the "Clothing" sales data for training the model.
  + Split the dataset into training and testing subsets (80% training, 20% testing).
  + Applied ARIMA with a configuration of (5, 1, 0) (5 autoregressive terms, 1 differencing, and 0 moving average terms).

**5. Forecasting and Evaluation**

* Forecasted the future sales for the testing period.
* Evaluated the model using RMSE and MAE metrics.
* Visualized the actual vs. predicted sales.

## **Real-World Applications**

1. **Retail Sales Forecasting:**
   * Helps businesses plan inventory, staffing, and marketing strategies based on future sales predictions.
2. **Supply Chain Optimization:**
   * Provides forecasts to optimize supply chain and inventory management, reducing wastage and improving profitability.
3. **Financial Planning:**
   * Retailers can use forecasts for better budgeting, ensuring efficient resource allocation.

## **Features**

1. **ARIMA Forecasting:**
   * Applied ARIMA model to predict future sales of clothing.
2. **Data Preprocessing and Splitting:**
   * Data was cleaned, preprocessed, and split into training and test sets for model evaluation.
3. **Visualization:**
   * Generated plots to compare the actual vs. predicted sales, helping to visualize the forecasting performance.

## **Code Highlights**

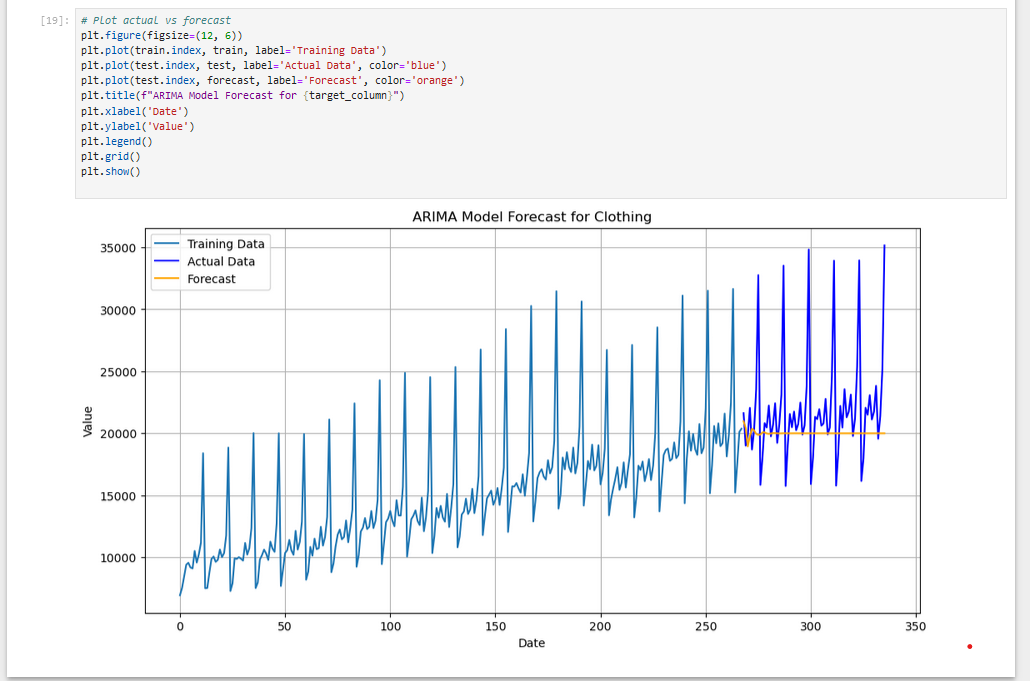
**A screenshot of a computer

Description automatically generatedA screen shot of a graph

Description automatically generatedA screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generatedA screen shot of a graph

Description automatically generated**

## **Results**

1. **Forecast Visualization:**
   * The forecast closely follows the trend of the actual sales data, but occasional mismatches highlight the challenges of forecasting volatile retail sales.

## **Future Work**

1. **Hyperparameter Tuning:**
   * Experiment with different ARIMA configurations (e.g., different orders for AR, I, MA) to optimize performance.
2. **Additional Models:**
   * Compare ARIMA with other forecasting models such as SARIMA, Prophet, or LSTM for better accuracy.
3. **Incorporating External Factors:**
   * Consider incorporating external factors like holidays, promotions, and economic indicators to improve forecasts.
4. **Automated Forecasting System:**
   * Develop a fully automated forecasting system for ongoing sales predictions, integrated with inventory management tools.