Project Proposal

- The project aims to leverage historical sales data and advanced analytics to predict future sales trends. By implementing robust machine learning models, businesses can enhance inventory management, refine marketing strategies, and optimize overall sales performance through data-driven decision-making.

Project Plan

- A **detailed project timeline** will be developed, outlining key milestones such as **data acquisition, preprocessing, model development, validation, deployment, and continuous monitoring**. A Gantt chart will provide a visual roadmap to track progress and ensure efficient resource utilization.

Task Assignment & Roles

- Team members will be assigned distinct roles, including **Data Engineer (data preprocessing and management), Data Scientist (model training and evaluation), Machine Learning Engineer (model optimization and deployment), and Project Manager (overall coordination and risk management)**. These roles will ensure the successful execution of each project phase.

Risk Assessment & Mitigation Plan

- Potential risks include **data inconsistency, missing values, model overfitting, computational inefficiencies, and deployment scalability issues**. To mitigate these risks, strategies such as **data validation pipelines, feature engineering optimizations, hyperparameter tuning, and cloud-based deployment solutions** will be implemented.

KPIs (Key Performance Indicators)

- The success of the project will be quantified using key metrics, including:
    - **Prediction Accuracy** (e.g., RMSE, MAE, MAPE) for model evaluation.
    - **System Uptime & Response Time** to ensure a seamless user experience.

- **User Adoption Rate** to measure engagement with the forecasting system.
- **Inventory Optimization Impact** to assess improvements in stock management.

Literature Review

Feedback & Evaluation

- Lecturer evaluations will provide insights into the **technical soundness, business impact, and implementation quality** of the project.

Suggested Improvements

- Potential improvements may involve **enhanced feature selection techniques, integration of external datasets (e.g., weather, social trends), advanced deep learning approaches, and real-time forecasting capabilities**.

Final Grading Criteria

- The assessment criteria will be based on:
  - **Project Documentation (20%)** – Clarity and completeness of reports.
  - **Implementation Quality (30%)** – Model performance and system robustness.
  - **Testing & Validation (30%)** – Accuracy and reliability of results.
  - **Final Presentation (20%)** – Effectiveness in communicating findings and demonstrating the model.

Requirements Gathering

Stakeholder Analysis

- The key stakeholders include:

- **Business Executives & Managers** – Require accurate sales insights for decision-making.
- **Sales & Marketing Teams** – Need predictions to optimize campaigns.
- **Inventory & Supply Chain Managers** – Use forecasts to manage stock levels efficiently.

User Stories & Use Cases

- The system will support multiple use cases, including:
    1. *As a sales manager, I want to receive monthly sales forecasts so that I can proactively adjust inventory and marketing strategies.*
    2. *As a warehouse operator, I want to track inventory levels based on predicted sales so that I can prevent stockouts or overstocking.*
    3. *As a retailer, I want to analyze sales trends for different product categories so that I can optimize shelf space allocation.*
    4. *As a finance analyst, I want to predict revenue based on sales trends so that I can create more accurate financial projections.*
    5. *As a marketing executive, I want to identify seasonal demand patterns so that I can launch targeted promotions.*
    6. *As a business owner, I want real-time dashboards displaying sales forecasts so that I can make informed operational decisions.*

Functional Requirements

- The system must support:
    - **Automated data ingestion** from multiple sources.
    - **Accurate sales forecasting** using machine learning models.
    - **Interactive dashboards** to visualize trends and insights.
    - **User authentication & role-based access control**.

Non-functional Requirements

- The system should exhibit:

- **Scalability** – Ability to handle increasing data volumes.
- **Security** – Protection against unauthorized access and data breaches.
- **Performance Efficiency** – Fast response times for forecasts.

System Analysis & Design

Problem Statement & Objectives

- The inability to accurately predict sales leads to **overstocking, lost revenue, and suboptimal marketing strategies**. This project addresses these challenges by **developing an AI-driven sales forecasting solution that improves planning and profitability**.

Use Case Diagram & Descriptions

- A **use case diagram** will illustrate the system's core interactions, including **data input, model training, forecasting generation, and result visualization**.

Software Architecture

- The system will follow a **modular microservices-based architecture**, ensuring flexibility and scalability.

Database Design & Data Modeling

- **ER Diagram:** The database will consist of well-structured tables for **sales transactions, product details, promotions, and external influencing factors**.
- **Schema Optimization:** Normalization techniques will be applied to enhance data integrity and retrieval efficiency.

Key Features for Regression Model

Feature Name	Description	Data Type
Date	The timestamp for each sales record	Object
Quantity Sold (liters/kg)	Number of units sold	Integer

Price per Unit (sold)	Selling price per unit	Float
Approx. Total Revenue	Total revenue generated	Float
Customer Location	Geographical region of purchase	Object
Sales Channel	Method of sales (Online/Retail/Wholesale)	Object
Quantity in Stock	Available stock quantity	Integer

Data Flow & System Behavior

- **DFD (Data Flow Diagram):** Visual representation of data processing pipelines.
- **Sequence Diagrams:** Step-by-step interaction breakdown for **forecast generation and user queries**.
- **Activity & State Diagrams:** Comprehensive mapping of **system processes and state transitions**.
- **Class Diagram:** Object-oriented design showcasing the **relationships between key system entities**.

UI/UX Design & Prototyping

- **Wireframes & Mockups:** High-fidelity UI prototypes ensuring intuitive design.
- **User-Centric Design Principles:** Emphasizing **ease of navigation, responsiveness, and accessibility**.

System Deployment & Integration

Technology Stack

- **Backend:** Python (Flask, FastAPI) for API development.
- **Frontend:** Dash/Streamlit for interactive user dashboards.

- **Database:** PostgreSQL for structured storage and efficient querying.

Deployment Diagram

- The system will be **deployed on a cloud environment (AWS/GCP)** with **automated CI/CD pipelines** for seamless updates.

Component Diagram

- The system's **high-level architecture** will depict interactions between **data storage, machine learning, and user interfaces**.

Additional Deliverables (if applicable)

API Documentation

- **RESTful API endpoints** will be documented, allowing integration with external business intelligence tools.

Testing & Validation

- **Comprehensive testing approach**, including:
    - **Unit Testing** – Validating individual components.
    - **Integration Testing** – Ensuring smooth interactions between modules.
    - **User Acceptance Testing** – Gathering feedback from end users.

Deployment Strategy

- The system will be **monitored using cloud-based analytics**, with **automated alerts for performance degradation**.
- **Model retraining pipelines** will be established to **adapt forecasts to new sales trends dynamically**.

Milestone	Step	Detailed Tasks
Milestone 1: Data Collection, Exploration, and Preprocessing	1. Data Collection	- Obtain a dataset from sources like Kaggle, UCI Repository, or create synthetic data. - Ensure the dataset includes relevant features: customer demographics, subscription details, usage behavior, etc.
	2. Data Exploration	- Load the dataset into a pandas DataFrame. - Inspect the dataset structure using `.info()` and `.describe()`. - Check for missing values using `.isnull().sum()`. - Identify duplicates and remove them. - Analyze data distributions with histograms, boxplots, and pair plots.
	3. Data Preprocessing	- Handle missing values: Use imputation (mean/median/mode) or drop rows with excessive missing data. - Detect and handle outliers using IQR or Z-score methods. - Convert categorical variables into numerical format using one-hot encoding or label encoding. - Scale numerical features using MinMaxScaler or StandardScaler.

	4. Exploratory Data Analysis (EDA)	- Create visualizations: histograms, bar charts, and heatmaps to analyze feature relationships. - Check for correlations using `.corr()` and visualize with a heatmap. - Analyze class imbalance in churn labels.
Milestone 2: Advanced Data Analysis and Feature Engineering	5. Advanced Data Analysis	- Conduct statistical tests (e.g., t-tests, ANOVA, chi-squared) to identify significant features. - Use feature selection techniques like Recursive Feature Elimination (RFE) and Mutual Information to determine important features.
	6. Feature Engineering	- Create new features based on domain knowledge (e.g., tenure, average monthly charges, frequency of service usage). - Perform transformations like log-scaling, standardization, and normalization for better model performance.
	7. Data Visualization for Insights	- Develop customer segmentation plots. - Create interactive dashboards with tools like `seaborn`, `plotly`, or `Tableau` for better business insights.
Milestone 3: Machine Learning Model	8. Model Selection	- Choose classification models like Logistic Regression, Decision Trees, Random Forest, or Gradient

Development and Optimization		Boosting. - Justify model selection based on dataset characteristics.
	9. Model Training	- Split the dataset into training and testing sets (e.g., 80-20 or 70-30 split). - Apply oversampling (SMOTE) or undersampling techniques if the dataset is imbalanced. - Train models and evaluate performance using cross-validation.
	10. Model Evaluation	- Assess models using accuracy, precision, recall, F1-score, and ROC-AUC. - Plot confusion matrices to visualize model predictions.
	11. Hyperparameter Tuning	- Use Grid Search, Randomized Search, or Bayesian Optimization to improve model performance.
	12. Model Comparison	- Compare models based on evaluation metrics. - Select the best-performing model for deployment.
Milestone 4: MLOps, Deployment, and Monitoring	13. MLOps Implementation	- Use MLflow or DVC to track experiments, hyperparameters, and model versions.
	14. Model Deployment	- Deploy the model as a REST API using Flask or FastAPI. - Deploy on cloud platforms (AWS, Google Cloud, Azure) for scalability. - (Optional) Create an interactive dashboard using Streamlit or Dash.
	15. Model Monitoring	- Set up monitoring tools to detect performance degradation. - Implement alerting

		mechanisms for significant prediction errors.
	16. Model Retraining Strategy	- Define a retraining pipeline to update the model periodically based on new data.
Milestone 5: Final Documentation and Presentation	17. Final Report	- Document the problem statement, dataset details, model performance, and business impact. - Include challenges faced and solutions implemented.
	18. Final Presentation	- Create a stakeholder-friendly presentation summarizing project results. - Provide a live demo of the deployed model or dashboard.
	19. Future Improvements	- Suggest enhancements such as adding new data sources, improving model interpretability, or optimizing deployment.

Steps to Implementation the model:

1. **Data Preparation:**
   - Collect and clean sales data.
   - Handle missing values and outliers.
   - Convert the dataset into a time-series format.
2. **Exploratory Data Analysis (EDA):**
   - Visualize sales trends using time-series plots.
   - Perform stationarity tests (ADF Test).
   - Identify seasonality using autocorrelation plots.
3. **Model Selection:**
   - Use **ACF (AutoCorrelation Function)** and **PACF (Partial AutoCorrelation Function)** to determine the best parameters.
   - Identify the optimal values for (p, d, q) and (P, D, Q, S) using grid search.
4. **Model Training & Evaluation:**
   - Train the model on historical sales data.
   - Evaluate using metrics like RMSE, MAE, and MAPE.
   - Fine-tune parameters for better accuracy.
5. **Forecasting Future Sales:**
   - Use the trained model to predict future sales.

- Generate confidence intervals for predictions.
- Compare actual vs. predicted sales to assess accuracy.

6. **Model Deployment & Monitoring:**
   - Deploy using **Flask/Streamlit** for real-time forecasting.
   - Set up automated model retraining as new data arrives.
   - Implement monitoring for accuracy tracking and updates.

1. Essential Columns for Sales Forecasting

These columns are **mandatory** to train the model effectively:

Column Name	Description	Required for model?
Date	The timestamp for each sales record (daily, weekly, or monthly). **Format:** YYYY-MM-DD	Yes (Primary index)
Sales	The total number of units sold or revenue generated on a given date.	Yes (Target variable)

2. Additional Recommended Columns (Enhance Forecasting Accuracy)

These columns help the model **capture seasonality, trends, and external effects**:

Column Name	Description	Why It's Important?
`Store_ID`	Unique identifier for each store (if multiple locations exist).	Helps analyze store-level sales patterns.
`Product_ID`	Unique identifier for each product.	Useful for product-level forecasting.
`Category`	Product category (e.g., electronics, clothing, food)	Allows grouping sales trends by category.
`Day_of_Week`	The day name (e.g., Monday, Tuesday).	Captures weekly seasonality.
`Month`	The month number (1 to 12).	Helps in identifying yearly seasonality.
`Holiday`	Binary flag (1 = Holiday, 0 = Non-Holiday).	Sales often spike on holidays.

Promotion	Binary flag (1 = Promotion active, 0 = No promotion).	Promotions can significantly affect sales.
`Weather`	Categorical (Sunny, Rainy, Snowy, etc.) or Temperature (°C/°F).	Some products are weather-sensitive (e.g., winter clothes).
`Stock_Availability`	Binary flag (1 = In stock, 0 = Out of stock).	Ensures stockouts don't cause misleading trends.

3. Derived Features (Feature Engineering)

These features can be created to improve model performance:

Feature	Description	How It Helps?
`Lag_Sales_7`	Sales from the previous week (7-day lag).	Captures short-term seasonality.
`Lag_Sales_30`	Sales from the previous month (30-day lag).	Helps in long-term trend analysis.
`Rolling_Avg_7`	7-day moving average of sales.	Smoothens fluctuations in sales.
`Rolling_Avg_30`	30-day moving average of sales.	Captures longer trends.