Forecasting and AI-Driven Customer Segmentation System (FACETS)

Participants:

- **David Ryan - 002021938**
- **Harshil Nandwani - 002762958**

Project Tasks

- **David:** Data preprocessing, customer segmentation (clustering), and LLM-based insight generation.
- **Harshil:** Demand forecasting (time-series modeling), report and visualization development, and model evaluation.

Problem Statement

Businesses need to understand both future demand trends and customer behavior patterns to optimize operations, marketing, and inventory. Traditionally, demand forecasting and customer segmentation are done separately using statistical or machine learning models. However, stakeholders often struggle to interpret the raw numbers, leading to poor decision-making. We propose a *hybrid AI-powered market intelligence system* that combines demand forecasting, customer segmentation, and LLM-Generated insights.

We decided to call this project idea “Forecasting and AI-Driven Customer System”, otherwise called FACETS, since were dealing with multiple facets of business operations, marketing, and inventory to make better decisions. This would not only enhance business decision-making by providing actionable insights in natural language but also bridge the gap between complex machine learning models and non-technical business users. This would fully demonstrate how LLMs can augment data science workflows by transforming raw model outputs into explainable insights.

Literature Review

(A) Demand Forecasting with Time-Series Models

Demand forecasting is a critical task for retail, finance, and supply chain management, where businesses need to predict future sales trends based on historical data.

(1) Classical Forecasting Models

- Xilong Chen from SAS Institute Inc introduced the VARIMAX (Vector Autoregressive Moving Average with eXogenous regressors) model, which remains a widely used statistical method for time-series forecasting (Intro to VARMAX).
- Hyndman and Athanasopoulos expanded on classical models with their work on Exponential Smoothing State Space Models (ETS), showing how seasonality and trends impact sales (Hyndman & Athanasopoulos, 2018).

Limitation: Classical models are effective but struggle with nonlinear dependencies and long-term forecasting.

(2) Machine Learning & Deep Learning Forecasting

- Facebook's Prophet model (Taylor & Letham, 2018) introduced a forecasting method that is more robust to missing data and changing trends than VARIMAX.
- LSTMs (Long Short-Term Memory Networks) (Hochreiter & Schmidhuber, 1997) are widely used for capturing long-term dependencies in sequential data. Studies have shown LSTM-based demand forecasting outperforms VARIMAX in volatile markets (Siami-Namini et al., 2018).
- Transformer-based forecasting models have been gaining traction, showing better performance in capturing hierarchical dependencies (Lim et al., 2021).

Our Contribution: Instead of relying solely on quantitative metrics, we integrate LLMs to explain demand patterns in natural language, making the insights more interpretable for non-technical users.

(B) Customer Segmentation with Clustering

Customer segmentation is crucial in marketing, personalization, and targeted advertising, allowing businesses to group customers based on purchasing behavior and demographics.

(1) Traditional Clustering Approaches

- K-Means Clustering is one of the most widely used methods for customer segmentation due to its simplicity and efficiency (Lloyd, 1982).
- Hierarchical Clustering has been used to visualize customer relationships but is computationally expensive for large datasets (Ward, 1963).

Limitation: K-Means assumes clusters are spherical and fails when dealing with non-uniform customer behaviors.

(2) Advanced Clustering for Customer Segmentation

- DBSCAN is a density-based algorithm that better handles outliers and complex clusters than K-Means (Ester et al., 1996).
- Gaussian Mixture Models (GMMs) provide a probabilistic approach to segmentation but require significant tuning (Reynolds, 2009).

Our Contribution: Instead of just providing raw cluster outputs, we use an LLM to describe each customer segment in business-friendly terms, enhancing interpretability.

(C) Large Language Models (LLMs) for Business Intelligence

LLMs like GPT-4 and LLaMA-2 have demonstrated the ability to summarize complex data, generate business insights, and assist in decision-making.

(1) LLMs for Data-to-Text Summarization

- Chen et al. (2023) demonstrated that LLMs can generate human-like explanations from numerical data, improving accessibility for non-technical users (Chen et al., 2023).

- RAG (Retrieval-Augmented Generation) improves factual accuracy in business insights by retrieving relevant knowledge before generating text (Lewis et al., 2020).

Limitation: Many existing studies use LLMs for Q&A chatbots rather than structured business insights.

(2) LLMs for Market Insights & Business Analytics

- Dimitrov explored LLMs for financial data analysis, showing that LLMs can summarize earnings reports and stock trends (Dimitrov et al., 2021).
- AI-generated executive summaries have been tested in corporate environments, showing that human decision-makers prefer LLM-assisted reports over raw numerical outputs (Song et al., 2022).

Our Contribution: Unlike previous studies that focus on financial analysis, we apply LLMs to business intelligence for demand forecasting & customer segmentation. We incorporate human feedback mechanisms to ensure generated insights are reliable and business friendly.

Methodology

1. Data collection & Preprocessing
   1.1. We will be using **two datasets**:
      1.1.1. Rossman Store Sales – for demand forecasting
      1.1.2. Online Retail II from UCI – for customer segmentation
   1.2. Collect historical sales data (e.g., e-commerce transactions, financial records, retail purchases)
   1.3. Handle missing values, outliers, and normalize time-series data
   1.4. Extract features like seasonality, customer purchase frequency, and spending patterns
2. Demand Forecasting
   2.1. Use ARIMA for baseline time-series forecasting of data
   2.2. Then use LSTM/Transformer-based forecasters for deeper temporal patterns
   2.3. Metrics: MAE and RMSE for evaluating forecast accuracy
3. Customer Segmentation
   3.1. Utilizing clustering techniques such as K-Means and DBSCAN
   3.2. May have to use dimensionality techniques like PCA.
   3.3. Metrics Silhouette Score
4. LLM-Based Insight Generation
   4.1. Main approach is to use DeepSeek-V3 with Hugging Face library to generate natural language reports from model outputs and export them as documents. Here are some example prompts:
      4.1.1. "Explain the seasonal trends observed in the last six months and give suggestions for optimizing our inventory for the next six months."
      4.1.2. "Describe key customer segments and suggest marketing strategies"
   4.2. We can evaluate this with possibly human assessment (fluency, coherence, relevance) as well as metrics such as perplexity, Rouge-2 precision, and recall.
5. Dashboard (Optional)
   5.1. Visualization Tools: Plotly, Google Sheets, or Tableau.

5.2. Interactivity: Users can select time periods or customer segments to generate insights dynamically.

Libraries and Tools

- Python Libraries:
    - Analysis & Cleaning: NumPy, Pandas
    - Forecasting: statsmodels, prophet, pytorch-forecasting
    - Clustering: scikit-learn, hdbscan
    - LLM Integration: OpenAI API, LangChain, transformers
    - Visualization: Plotly, Matplotlib, Seaborn
- Platform Options:
    - Google Colab / Jupyter Notebook (for analysis)
    - Google Sheets API / Tableau / Power BI (for reports and dashboards)

Expected Results

Our project aims to develop a hybrid AI-powered market intelligence system that integrates demand forecasting, customer segmentation, and LLM-generated business insights. We expect our demand forecasting models (ARIMA, Prophet, LSTMs, and Transformers) to accurately predict future sales trends, with deep learning methods likely improving performance on complex patterns. These numerical predictions will be supplemented by LLM-generated reports, which will transform raw data into interpretable business insights, helping non-technical users make informed decisions. Our evaluation will compare traditional numerical outputs against LLM-enhanced explanations, using metrics like MAE, RMSE, and human readability scores to assess effectiveness. The goal is to determine whether integrating LLMs improves understanding and usability for business stakeholders.

For customer segmentation, we anticipate that K-Means, DBSCAN, and hierarchical clustering will reveal at least 3–5 distinct consumer groups, categorized by purchasing behavior and demographics. The LLM will generate natural language descriptions of these segments, providing actionable marketing strategies. Evaluation metrics like Silhouette Score and Davies-Bouldin Index will measure clustering quality, while human assessment will gauge the usefulness of LLM-generated insights. Additionally, we will analyze system efficiency, ensuring the LLM operates within a 5-second response time while keeping API costs below $5 or leveraging free alternatives like DeepSeek from Hugging Face. Ultimately, our project seeks to enhance business intelligence workflows by combining machine learning and natural language AI, bridging the gap between complex data science models and practical decision-making.

There are potential risks and mitigation strategies that we may need to consider. If the forecasting is continually inaccurate, it could lead to LSTMs/Transformers underperforming on noisy data. If that's the case, we may need to do feature engineering to improve forecasting model performance. If the LLM is hallucinating, this may lead to inaccurate insights. To fix that, we may need to use retrieval-augmented generation (RAG) to ground responses. There also clustering misinterpretation, where the customer segments may not align with business expectations, so to fix that we may need to Perform manual validation and refine clustering parameters. Finally, there's the risk of LLM inferencing taking too long, in which that's the case, we may need to do optimize token length and use batch processing.