

Group Project

BIA 5402 – Machine Learning II

1. Data Collection & Exploration (5 points)

- Choose a real-world time series dataset (from Kaggle, UCI, or public repositories).
- Perform Exploratory Data Analysis (EDA):
 - Visualize the time series.
 - Identify missing values, outliers, trends, and stationarity (e.g., ADF test).
 - Explain whether the data needs differencing or smoothing.
 - Discuss if external factors (e.g., holidays, economic changes) might impact your dataset.

2. Time Series Decomposition (5 points)

- Use statsmodels.seasonal_decompose to break the series into:
 - **Trend**
 - **Seasonality**
 - **Residuals**
- Compare Additive vs. Multiplicative decomposition.
- What insights can businesses gain from these components? (E.g., does seasonality indicate predictable demand cycles?)

3. Forecasting Models & Performance Evaluation (5 points)

Your group must compare **three forecasting models**:

1. **ARIMA (AutoRegressive Integrated Moving Average) or any alternatives**
2. **Exponential Smoothing (ETS) or alternatives.**
3. **Prophet** (or an alternative deep learning-based model)
4. **LSTM (bonus points)**

Split the data into training and testing. Tune hyperparameters and explain your choices. Evaluate model performance using:

- **RMSE, MAE, MAPE, MSE.**

Which model works best for short-term vs. long-term forecasting?

How would your choice of model change for different industries (e.g., finance vs. healthcare)?

5. Streamlit Web App Implementation (5 points)

Your interactive web app should allow users to:

- Upload a CSV file containing a time series dataset.
- Choose Additive or Multiplicative decomposition.
- Select a forecasting model and visualize predictions.
- Display evaluation metrics for forecasting accuracy.

Instructions for Streamlit

1. Install Streamlit:

```
pip install streamlit
```

2. Create app.py and write the Streamlit script.

Import Streamlit and required libraries:

```
import streamlit as st

import pandas as pd

import matplotlib.pyplot as plt

from statsmodels.tsa.seasonal import seasonal_decompose
```

Add file uploader:

```
uploaded_file = st.file_uploader("Upload a CSV file", type="csv")
```

Allow users to select a forecasting model:

```
model_choice = st.selectbox("Choose a forecasting model", ["ARIMA",
"ETS", "Prophet"])
```

3. Run the app:

```
streamlit run app.py
```

4. (Bonus) Deploy the app using Streamlit Cloud or Heroku.

5. Presentation, Report & Demo Video (10 points)

- **Report (PDF, max 5 pages):**
 - Dataset and key insights from EDA.
 - Time series decomposition results.
 - Forecasting model comparison.
 - Screenshots and explanation of your **Streamlit web app**.
 - Use [this](#) A4 Word template.

- **Presentation (10 minutes, all members):**
 - Clearly explain the problem statement & dataset.
 - Summarize key insights from decomposition & forecasting.
 - Demonstrate the Streamlit app as if pitching it to stakeholders.
- **Recorded Demo Video (5 min):**
 - Show your Streamlit app in action.
 - Explain how users interact with it.
 - Walk through a forecasting example.

EVALUATION CRITERIA (30 points)

Category	Points
Data Collection & EDA	5
Time Series Decomposition	5
Forecasting Models & Evaluation	5
Streamlit Web App Implementation	5
Presentation, Report & Demo Video	10

Submission Guidelines

1. **GitHub Repository** with:
 - Jupyter Notebook (.ipynb)
 - Streamlit script (app.py)
 - Sample dataset
2. **Google Drive link** with:
 - PDF report
 - Presentation slides
 - Demo video

Bonus Points (+2)

- Implement an LSTM-based deep learning model for forecasting.
- Deploy the Streamlit app on Streamlit Cloud and share the live link.