SUMMARY & INSIGHTS
ON TASKS
OF
DATA SCIENCE INTERNSHIP at ARCHTECHNOLOGIES

Intern: Sana Azeem
Intern ID: ARCH-2509-0783

Task 1: Titanic Survival Classification

Use the Titanic dataset, build a machine learning model to predict whether a passenger survived or not based on features like age, gender, ticket class, and fare. Your task is to clean and preprocess the data, train a classification model, and evaluate its performance.

Summary & Insights

In this project, we successfully built a **machine learning model** to predict Titanic passenger survival.

Steps Completed:

1. **Data Exploration & Cleaning** – handled missing values and removed irrelevant columns.

2. **Feature Engineering** – created new features such as *FamilySize* to enrich the dataset.

3. **Data Preprocessing** – encoded categorical variables into numerical form for model compatibility.

4. **Model Training** – used a **Random Forest Classifier** with 200 trees and max depth 8, and applied **class weights** to ensure better balance.

5. **Evaluation** – achieved around **83% accuracy** on the test set.

    - The model performed very well at predicting **non-survivors (class 0, recall = 0.89)**.

    - Performance for **survivors (class 1, recall = 0.76)**.

6. **Feature Importance** – the most influential factors were:
    - **Sex (0.35)** – strongest predictor of survival.
    - **Fare (0.22)** – higher fares indicated better survival chances.
    - **Age (0.17)** – younger passengers had higher probability of survival.
    - **Pclass_3 and FamilySize** also contributed, while other features had less impact.

Key Insights:

- **Gender (Sex)** was the strongest predictor, confirming the “women and children first” policy.
- **Fare and Class** reflected socio-economic influence on survival chances.
- **Age** showed that younger passengers were more likely to survive.
- **Family Size** added useful information, but with smaller influence compared to the top three features.

Conclusion:

The Random Forest model performed strongly with **over 83% accuracy** and provided meaningful insights into survival patterns.

The project demonstrates the complete workflow of a machine learning classification task: **Data Cleaning → Feature Engineering → Model Training → Evaluation → Insights.**

Task 2: Stock Price Prediction

Build a model to predict future stock prices based on historical stock data, including features like opening price, closing price, high, low, and trading volume. Your task is to preprocess the data, choose a suitable model (e.g., linear regression or LSTM), train it, and evaluate its prediction accuracy.

Summary & Insights:

In this project, we successfully built a **machine learning model** to predict Apple Stock Prices.

Steps Completed:

1. **Data Collection**
    - Stock data for **Apple (AAPL)** was downloaded from *Yahoo Finance* covering **2015–2023**.
2. **Feature Engineering**
    - Added technical indicators to enrich the dataset:
        - **RSI (14-day)** → Momentum indicator.
        - **MACD (12–26 EMA)** → Trend-following indicator.
        - **Bollinger Bands (20 MA ± 2 STD)** → Volatility indicator.
    - Final feature set included **OHLCV + RSI + MACD + Bollinger Bands**.
3. **Data Preprocessing**
    - **MinMaxScaler** used to normalize features.
    - Data split: **80% training / 20% testing**.
    - **Sequence length = 60 days** → Model predicts the next day's *closing price*.
4. **Model Architecture**
    - **LSTM Neural Network** with two stacked LSTM layers and Dropout for regularization.
    - Dense layers for final regression output.
    - Optimizer: **Adam (lr=0.001)**.
    - Loss function: **MSE**.
    - **EarlyStopping** used to prevent overfitting.

5. **Training**
    - Trained up to **50 epochs** with **validation split (10%)**.
6. **Evaluation & Results**
    - Predictions inverse-transformed back to USD scale.
    - **Metrics:**
        - **MSE : 42.40**
        - **RMSE : 6.51 USD**
        - **R^2 : 0.8848**
        - **MAPE : 3.46%**
7. **Visualization**
    - Line plot comparing **true prices vs. predicted prices**.

Insights:

The LSTM model performed well (**$R^2 \approx 0.88$, MAPE ≈ 3.46%**), showing it can capture short-term patterns in Apple's stock prices.

Adding technical indicators (RSI, MACD, Bollinger Bands) improved predictions compared to using only price data.

The model is useful for identifying historical trends, but real-world performance would benefit from including external factors like news and market sentiment.

Conclusion:

The LSTM model achieved **high accuracy** in predicting Apple's stock price trends with an **R^2 of 0.88** and **MAPE of 3.46%**, showing that it captures historical patterns and technical indicators effectively. However, for real-world trading or portfolio use, external factors beyond technical data should be integrated for improved reliability.

The project demonstrates the complete workflow of a machine learning classification task: **Data Cleaning → Feature Engineering → Model Training → Evaluation → Insights.**

Additional Notes:

Model Training Observations:

During training, I observed that the evaluation metrics (MSE, RMSE, R^2, and MAPE) showed slight variation across different runs of the model.
This is a natural characteristic of deep learning models such as LSTMs, since they rely on random weight initialization, dropout layers, and data shuffling.

Dataset Choice

For this project, we used Apple's daily stock data from **January 2015 to December 2023**. This period was chosen because:

- It provides **enough historical data** (almost 9 years) for the LSTM model to effectively learn price patterns.
- It reflects Apple's **modern market behavior**, avoiding very old data that may no longer be relevant.
- Ending at 2023 ensures a **complete and closed dataset**, making model training and evaluation consistent.

Although more recent data (2024–2025) exists, it was not included to maintain a fixed evaluation window. However, the trained model can later be applied to this newer data to assess real-world generalization.