Weekly Report: Nusrat Jahan Nuha-Alpha AI

Week-03 (May 5 - May 9)

1. Course Documentation

Machine Learning Specialization: Advanced Learning Algorithms

This week, I learned about the foundational concepts and practical aspects of neural networks. I explored their definition and gained intuition on how they mimic the human brain to model complex patterns in data. I studied the structure of a neural network, understanding the roles of different layers, including input, hidden, and output layers, and how each contributes to the overall model. I delved into the mechanism of forward propagation, which involves passing input data through the network to generate predictions, and examined strategies for efficiently implementing neural networks using modern tools like TensorFlow. I also learned about activation functions such as ReLU and SoftMax—their purpose in introducing non-linearity and how to select the appropriate one based on the task, such as using SoftMax for multiclass classification. Optimization techniques, especially backpropagation, were covered in detail to understand how neural networks learn by adjusting weights to minimize error. I explored the importance of regularization methods like dropout and L2 regularization in preventing overfitting, and the bias-variance tradeoff as a key consideration in model generalization.

Additionally, I learned various data splitting techniques for training, validation, and testing to ensure robust evaluation. One-hot encoding was introduced as a crucial method for converting categorical variables into a numerical format suitable for machine learning models. Beyond neural networks, I also studied decision trees—their working principles and how they split data based on features—and advanced ensemble methods like XGBoost, which combine multiple models to improve predictive performance.

2. TESLA-S (Tesla Stock Price Forecasting) Project

Project Overview and Objective

Financial time series forecasting plays a key role in quantitative finance, and in this project, the focus is on predicting Tesla's monthly closing stock price using historical data. Tesla's stock is known for its volatility and high trading volume, which makes it both a challenging and interesting case for forecasting.

The project walks through the entire machine learning pipeline—from data preprocessing and exploratory data analysis (EDA) to feature engineering, model training, and performance evaluation. The main goal is to forecast Tesla's next-month closing price using supervised learning techniques and to compare the performance of traditional, machine learning, and deep learning models. This helps assess how well each model captures patterns and generalizes under different market conditions. Along the way, EDA is used to get a better sense of how the stock behaves over time and to spot trends or volatility that might affect predictions. Model accuracy is evaluated using standard metrics to see which approach performs best.

Completed Tasks

- Exploratory Data Analysis (EDA)
- Data Preprocessing
- Model Evaluation for Machine Learning Models

A. Exploratory Data Analysis (EDA)

Close Price Plot: This plot provides a visual representation of Tesla's closing stock prices over time. By charting the price chronologically, it becomes easier to observe overall trends, significant spikes or drops, and periods of stability or heightened movement. It serves as a foundational view for understanding how the stock has behaved historically.

Moving Averages Plot: The moving averages plot highlights both short- and long-term trends by overlaying moving average lines—calculated over 5, 10, and 20 days—on the closing price data. These smoothed lines help reduce noise and make it easier to detect sustained upward or downward trends, which are crucial for technical analysis.

Volatility Plot: This visualization shows how much Tesla's stock price fluctuates over time, helping to gauge the level of risk or uncertainty in the market. High volatility indicates rapid price changes, while low volatility suggests more stable movements. Understanding volatility is essential for both risk management and timing market entries or exits.

Seasonality Plot: The seasonality plot analyzes repeating patterns and periodic behaviors in Tesla's stock prices. It helps identify whether certain months or quarters consistently show similar trends, such as higher returns or increased volatility. This is useful for timing predictions and aligning strategies with historical cycles.

Correlation Heatmap: The correlation heatmap displays the relationships between key numerical features in the dataset. It visually indicates how strongly variables like volume, moving averages, and volatility are related to each other and to the closing price. This insight is valuable for feature selection and understanding multicollinearity in modeling.

Close Price vs. Volume Plot: This scatter or line plot explores the relationship between Tesla's closing price and its trading volume. It helps determine whether high trading activity corresponds with price movements, potentially indicating periods of market sentiment shifts or major news impacts.

Outlier Detection Plot (IQR Method): Using the Interquartile Range (IQR) method, this plot highlights anomalies in Tesla's stock price data. By identifying values that fall significantly outside the typical price range, it becomes easier to detect sudden market shocks, data entry errors, or unusual trading activity that might require special attention during modeling.

B. Data Preprocessing

Preparing the data

Defined a function `dataloader()`. The `dataloader()` function is responsible for loading the Tesla stock dataset from a CSV file named Tasla_Stock_Updated_V2.csv. It reads the data using pandas, converts the Date column to a datetime format, and sets it as the index to allow for time series operations. It also removes any rows with missing values to ensure the dataset is clean and ready for further processing. This function serves as the initial step in the data preprocessing pipeline.

Feature Engineering

The `add_features()` function enriches the raw dataset by generating new features that are valuable for time series forecasting. It calculates moving averages (MA5, MA10, MA20), which help smooth out price trends, and volatility using a rolling standard deviation. It also includes momentum features (1-day and 5-day percentage changes) to capture recent price movements. A monthly return feature is computed and forward-filled to align with the daily frequency of the dataset. Additionally, the target variable is defined as the next day's closing price (target), and any rows with missing values due to feature calculations are removed. This function transforms the dataset into a richer form suitable for supervised learning.

Data Splitting

The `data_split()` function is designed to divide the dataset into training and testing sets while preserving the temporal structure, which is critical in time series forecasting. It takes the input features X and target y, converts them to NumPy arrays, and optionally sorts them based on a specified key (column index or custom function). The function then determines a split index based on the test size and splits the data accordingly—assigning earlier records to training and more recent ones to testing. This time-aware split ensures that future data isn't leaked into the training set, preserving the integrity of the model evaluation.

Data Scaling

The `scale()` function normalizes both the input features and target values using MinMaxScaler from sklearn.preprocessing. It fits the scaler on the training data and applies the same transformation to the test data to maintain consistency. Both the input features (X_train, X_test) and target values (y_train, y_test) are scaled to a range between 0 and 1. This scaling process is critical for machine learning algorithms—especially neural networks—which perform better when input values are on a similar scale. The function returns the scaled datasets along with the target scaler (y_scaler) so that predictions can be inverse-transformed later if needed.

C. Model Evaluation

Completed evaluation on Machine Learning Models.

Linear Regression

Linear Regression achieved the best performance among all models with a test RMSE of 0.021 and an R^2 score of 0.9714. This model assumes a linear relationship between the input features and the target variable. It is simple, fast, and interpretable, making it a strong baseline for time series forecasting when the underlying relationships are mostly linear.

Random Forest

Random Forest, an ensemble learning method based on multiple decision trees, performed reasonably well with a test RMSE of 0.029 and an R^2 of 0.9461. It reduces overfitting by averaging multiple trees and captures non-linear relationships effectively. While slightly less accurate than linear regression in this case, it remains a robust model for complex data.

XGBoost

XGBoost, a gradient boosting technique optimized for speed and performance, delivered competitive results with a test RMSE of 0.027 and an R^2 score of 0.9505. It builds trees sequentially, improving on the residuals of prior models, and handles non-linear patterns efficiently. Its performance was close to Random Forest but with slightly better accuracy.

Decision Tree

The standalone Decision Tree model showed the weakest performance, with a test RMSE of 0.039 and an R^2 of 0.9024. It creates splits in the data based on feature thresholds, making it easy to understand but prone to overfitting. Without ensemble techniques, its generalization ability is limited, which is reflected in the lower accuracy.

Support Vector Regression (SVR)

SVR performed modestly with a test RMSE of 0.036 and an R^2 of 0.9141. It works by finding a hyperplane that best fits the data within a defined margin and is effective in handling high-dimensional spaces and non-linear relationships using kernels. However, it can be sensitive to hyperparameters and scaling, which may explain the slightly lower performance compared to tree-based models.

3. Ice-Breaking Session

Had an interactive session led by the business head of Cloudly, along with senior members from the HR and Sales & Marketing departments. Gained valuable insights into key career development concepts, future planning strategies, the founder's vision, and Cloudly's organizational culture.