Shirpur Education Society's

R. C. Patel Institute of Technology, Shirpur

(An Autonomous Institute)

DEPARTMENT OF ARTIFICIAL INTELLIGENCE & MACHINE LEARNING

Artificial Intelligence in Finance Laboratory

PEAI7042L

Final Year B. Tech. Semester VII,

Academic year 2024-24

Name of Student: ______________________________

PRN: ______________ Batch: ________ Roll No.: ________

R. C. PATEL
INSTITUTE OF TECHNOLOGY

DEPARTMENT OF ARTIFICIAL INTELLIGENCE & MACHINE LEARNING

Laboratory Report

Artificial Intelligence in Finance Laboratory PEAI7042L

Experiment No. 1

Name of Student: ______________________________

Division: ______ Batch: _______ Roll No.: ________

Date of Performance: _______________ Date of Submission: _______________

Lab Instructor Prof. K D Chaudhari Signature:

SHIRPUR EDUCATION SOCIETY

R. C. PATEL
INSTITUTE OF TECHNOLOGY

DEPARTMENT OF ARTIFICIAL INTELLIGENCE & MACHINE LEARNING

Laboratory Report

Artificial Intelligence in Finance Laboratory PEAI7042L

Experiment No. 2

Name of Student: ____

Division: ______ Batch: _______ Roll No.: ________

Date of Performance: ________________ Date of Submission: ________________

Lab Instructor Prof. K D Chaudhari Signature:

R. C. PATEL
INSTITUTE OF TECHNOLOGY

DEPARTMENT OF ARTIFICIAL INTELLIGENCE & MACHINE LEARNING

Laboratory Report

Artificial Intelligence in Finance Laboratory PEAI7042L

Experiment No. 3

Name of Student: ____________________

Division: ______ Batch: _______ Roll No.: ________

Date of Performance: ____________ Date of Submission: ____________

Lab Instructor Prof. K D Chaudhari Signature:

R. C. PATEL
INSTITUTE OF TECHNOLOGY

DEPARTMENT OF ARTIFICIAL INTELLIGENCE & MACHINE LEARNING

Laboratory Report

Artificial Intelligence in Finance Laboratory PEAI7042L

Experiment No. 4

Name of Student: ____

Division: ______ Batch: _______ Roll No.: ________

Date of Performance: __________________ Date of Submission: __________________

Lab Instructor Prof. K D Chaudhari Signature:

R. C. PATEL INSTITUTE OF TECHNOLOGY

DEPARTMENT OF ARTIFICIAL INTELLIGENCE & MACHINE LEARNING

Laboratory Report

Artificial Intelligence in Finance Laboratory PEAI7042L

Experiment No. 5

Name of Student: ______________________________

Division: ______ Batch: _______ Roll No.: ________

Date of Performance: ______________ Date of Submission: ______________

Lab Instructor Prof. K D Chaudhari Signature:

R. C. PATEL
INSTITUTE OF TECHNOLOGY

DEPARTMENT OF ARTIFICIAL INTELLIGENCE & MACHINE LEARNING

Laboratory Report

Artificial Intelligence in Finance Laboratory PEAI7042L

Experiment No. 6

Name of Student: ______________________________

Division: ______ Batch: _______ Roll No.: ________

Date of Performance: ____________ Date of Submission: ____________

Lab Instructor Prof. K D Chaudhari Signature:

R. C. PATEL
INSTITUTE OF TECHNOLOGY

DEPARTMENT OF ARTIFICIAL INTELLIGENCE & MACHINE LEARNING

Laboratory Report

Artificial Intelligence in Finance Laboratory PEAI7042L

Experiment No. 7

Name of Student: ____________________

Division: ______ Batch: _______ Roll No.: ________

Date of Performance: ______________ Date of Submission: ______________

Lab Instructor Prof. K D Chaudhari Signature:

R. C. PATEL
INSTITUTE OF TECHNOLOGY

DEPARTMENT OF ARTIFICIAL INTELLIGENCE & MACHINE LEARNING

Laboratory Report

Artificial Intelligence in Finance Laboratory PEAI7042L

Experiment No. 8

Name of Student: ____

Division: ______ Batch: _______ Roll No.: ________

Date of Performance: __________________ Date of Submission: __________________

Lab Instructor Prof. K D Chaudhari Signature:

R. C. PATEL
INSTITUTE OF TECHNOLOGY

DEPARTMENT OF ARTIFICIAL INTELLIGENCE & MACHINE LEARNING

Laboratory Report

Artificial Intelligence in Finance Laboratory PEAI7042L

Experiment No. 9

Name of Student: ______________________________

Division: ______ Batch: _______ Roll No.: ________

Date of Performance: ______________ Date of Submission: ______________

Lab Instructor Prof. K D Chaudhari Signature:

R. C. PATEL
INSTITUTE OF TECHNOLOGY

DEPARTMENT OF ARTIFICIAL INTELLIGENCE & MACHINE LEARNING

Laboratory Report

Artificial Intelligence in Finance Laboratory PEAI7042L

Experiment No. 10

Name of Student: ____________________

Division: ______ Batch: _______ Roll No.: ________

Date of Performance: ______________ Date of Submission: ______________

Lab Instructor Prof. K D Chaudhari Signature:

R. C. PATEL INSTITUTE OF TECHNOLOGY
An Autonomous Institute

Department of
Artificial Intelligence & Machine Learning

Laboratory Manual
Subject: - Artificial Intelligence in Finance Laboratory
Semester: - VII **Class: - B. Tech** **Experiment No. 1**

Title: Case Study: Predicting Stock Prices with a Simple Neural Network.

Objective: To build and train a simple neural network model to predict future stock prices based on historical data.

HARDWARE CONFIGURATION / KIT:

Sr. No	Hardware Configuration	
1	Processor	1.5 GHz or more
2	RAM	512 MB Minimum
3	HDD	Minimum 1GB free Space
4	Standard I/O devices	

SOFTWARE REQUIREMENT:

Sr. No	Software Configuration	
1	Operating System	Windows 8 or later
2	Google Colab or Jupiter Notebook	User Account
3	Libraries: TensorFlow, Keras, NumPy, Pandas, Matplotlib and historical price data	

THEORY:

Stock market prediction is a complex task that has attracted the attention of researchers and investors alike. With the advent of machine learning, particularly neural networks, we can leverage historical data to make informed predictions about future stock prices. In this lab, we will explore a basic neural network architecture to predict stock prices.

Procedure:

1. **Data Acquisition and Preprocessing:**
    - **Data Acquisition:**
        - Use libraries like Pandas to download historical stock price data for a specific company.
        - Ensure the data includes features like opening price, closing price, high price, low price, and trading volume.
    - **Data Cleaning:**
        - Handle missing values (e.g., imputation or removal).
        - Remove outliers if necessary.
    - **Feature Engineering:**
        - Create additional features like moving averages, momentum, and volatility.
        - Normalize or standardize the features to a common scale.
    - **Data Splitting:**
        - Divide the data into training and testing sets.
2. **Model Building:**
    - **Define the Model Architecture:**
        - Create a sequential model with the following layers:
            - Input layer: Takes the normalized features as input.
            - Hidden layers: Use Dense layers with appropriate activation functions (e.g., ReLU).
            - Output layer: A single neuron with a linear activation function to predict the stock price.
    - **Compile the Model:**
        - Specify the optimizer (e.g., Adam), loss function (e.g., Mean Squared Error), and metrics (e.g., Mean Absolute Error).
3. **Model Training:**
    - **Train the Model:**
        - Fit the model to the training data, specifying the number of epochs and batch size.

        - Monitor the training process by plotting the loss and accuracy curves.

    - **Model Evaluation:**

        - Evaluate the model's performance on the testing set using metrics like Mean Squared Error and Mean Absolute Error.

        - Visualize the predicted and actual stock prices to assess the model's accuracy.

4. **Model Prediction:**

    - **Prepare the Input Data:**

        - Collect the latest stock data for the company.

        - Preprocess the data to match the model's input format.

    - **Make Predictions:**

        - Use the trained model to predict the future stock price.

Conclusion:

R. C. PATEL INSTITUTE OF TECHNOLOGY
An Autonomous Institute

Department of
Artificial Intelligence & Machine Learning

Laboratory Manual
Subject: - Artificial Intelligence in Finance Laboratory
Semester: - VII Class: - B. Tech Experiment No. 2

Title: Exploratory Data Analysis (EDA) with Financial Data.

Objective: Use of Python libraries like Pandas and Matplotlib to analyze and visualize historical financial data. Identify trends, patterns, and correlations in stock prices, market indices, or other financial indicators.

HARDWARE CONFIGURATION / KIT:

Sr. No	Hardware Configuration	
1	Processor	1.5 GHz or more
2	RAM	512 MB Minimum
3	HDD	Minimum 1GB free Space
4	Standard I/O devices	

SOFTWARE REQUIREMENT:

Sr. No	Software Configuration	
1	Operating System	Windows 8 or later
2	Google Colab or Jupiter Notebook	User Account
3	Libraries: NumPy, Pandas, Matplotlib and historical price data.	

THEORY:

Exploratory Data Analysis (EDA) is a crucial initial step in any data science project, especially when working with financial data. It involves visually and statistically exploring data to uncover patterns, anomalies, and trends. For financial data, EDA helps in understanding market behavior, identifying potential investment opportunities, and mitigating risks.

Key Steps in Financial Data EDA:

1. **Data Cleaning and Preparation:**
   - **Handling Missing Values:** Imputation or deletion of missing data points.
   - **Outlier Detection and Treatment:** Identifying and addressing extreme values.

   - **Data Normalization and Standardization:** Scaling data to a common range.
2. **Univariate Analysis:**
   - **Summary Statistics:** Calculating measures like mean, median, mode, standard deviation, and quartiles.
   - **Data Visualization:** Creating histograms, box plots, and density plots to visualize data distribution.
3. **Bivariate Analysis:**
   - **Correlation Analysis:** Measuring the strength and direction of relationships between variables.
   - **Scatter Plots:** Visualizing the relationship between two numerical variables.
   - **Cross-Tabulations:** Analyzing the relationship between categorical variables.
4. **Time Series Analysis:**
   - **Trend Analysis:** Identifying upward or downward trends in data.
   - **Seasonality Analysis:** Detecting recurring patterns at specific intervals.
   - **Stationarity Testing:** Checking if the statistical properties of a time series remain constant over time.
5. **Feature Engineering:**
   - **Creating New Features:** Deriving new features from existing ones (e.g., moving averages, momentum, volatility).
   - **Feature Selection:** Identifying the most relevant features for modeling.

Data:

- Historical stock price data (e.g., from Yahoo Finance, Google Finance, or a financial data provider).
- Market indices data (e.g., S&P 500, NASDAQ, Dow Jones).
- Economic indicators (e.g., GDP, inflation rates, interest rates).

Lab Tasks:

Task 1: Data Loading and Cleaning

1. **Load Data:**
   - Use Pandas to read the financial data from CSV, Excel, or other formats.

- Explore the data using head(), tail(), info(), and describe() methods.

2. **Data Cleaning:**
    - Handle missing values using techniques like imputation or removal.
    - Identify and correct data inconsistencies or errors.
    - Convert data types as needed (e.g., string to numeric).

Task 2: Exploratory Data Analysis

1. **Statistical Summary:**
    - Calculate summary statistics (mean, median, mode, standard deviation, quartiles) for key variables.
    - Use describe() to get a quick overview.
2. **Data Visualization:**
    - **Time Series Plots:**
        - Visualize stock prices over time using line plots.
        - Identify trends (upward, downward, or sideways) and seasonality.
    - **Histograms and Density Plots:**
        - Analyze the distribution of stock returns.
        - Identify outliers and potential anomalies.
    - **Box Plots:**
        - Compare the distribution of stock returns across different time periods or stocks.
        - Identify potential outliers and variability.
    - **Scatter Plots:**
        - Explore relationships between different variables (e.g., stock price and trading volume).
        - Identify potential correlations.
3. **Correlation Analysis:**
    - Calculate correlation coefficients (Pearson, Spearman) between pairs of variables.
    - Visualize correlations using a heatmap.
    - Identify strong positive or negative correlations.

Task 3: Feature Engineering and Data Preparation

1. **Feature Engineering:**
    - Create new features based on existing data (e.g., moving averages, momentum, volatility).
    - Consider technical indicators like RSI, MACD, and Bollinger Bands.
2. **Data Preparation for Modeling:**
    - Split the data into training and testing sets.
    - Scale or normalize the data as needed.

Task 4: Model Building and Evaluation (Optional)

1. **Model Selection:**
    - Choose appropriate models for time series forecasting or classification tasks (e.g., ARIMA, LSTM, Random Forest).
2. **Model Training and Evaluation:**
    - Train the model on the training data.
    - Evaluate the model's performance on the testing data using metrics like RMSE, MAE, or accuracy.

Reporting and Conclusion:

- Summarize the key findings from the EDA.
- Present visualizations and statistical results in a clear and concise manner.
- Discuss potential insights and implications for investment decisions.
- Identify limitations and areas for further exploration.

Conclusion:

R. C. PATEL INSTITUTE OF TECHNOLOGY
An Autonomous Institute

Department of
Artificial Intelligence & Machine Learning

Laboratory Manual
Subject: - Artificial Intelligence in Finance Laboratory
Semester: - VII Class: - B. Tech. Experiment No. 3

Title: Predictive Modelling for Stock Prices.

Objective: Build machine learning models (e.g., linear regression, decision trees, or LSTM neural networks) to predict future stock prices based on historical data. Evaluate the performance of the models using metrics like mean squared error (MSE) or accuracy.

HARDWARE CONFIGURATION / KIT:

Sr. No	Hardware Configuration	
1	Processor	1.5 GHz or more
2	RAM	512 MB Minimum
3	HDD	Minimum 1GB free Space
4	Standard I/O devices	

SOFTWARE REQUIREMENT:

Sr. No	Software Configuration	
1	Operating System	Windows 8 or later
2	Google Colab or Jupiter Notebook	User Account
3	Libraries: pandas, NumPy, Scikit-learn, Matplotlib and historical price data	

THEORY:

This experiment explores the use of artificial intelligence and machine learning in predicting stock prices. It discusses various techniques, including traditional statistical methods and advanced machine learning algorithms, for forecasting future stock movements. By understanding the principles and limitations of these models, it provides valuable insights into the potential of AI in finance.

The salient concepts involved are the challenges of stock price prediction, including market randomness, external factors, and traditional statistical methods like time series analysis, machine learning approaches like supervised algorithms, unsupervised techniques, and deep learning models. It also discusses the limitations and challenges of these methods, as well as the limitations and challenges of these methods. It also discusses the use of deep learning models for capturing complex temporal dependencies.

Procedure:

1. Data Acquisition and Preprocessing:

- **Data Acquisition:**
    - Use libraries like pandas_datareader to fetch historical stock price data for a specific stock or index.
    - Ensure the data includes relevant features like opening price, closing price, high, low, volume, and any other relevant indicators.
- **Data Cleaning:**
    - Handle missing values using techniques like imputation or removal.
    - Identify and remove outliers.
- **Feature Engineering:**
    - Create new features from existing ones, such as moving averages, momentum, volatility, and technical indicators.
    - Consider using domain knowledge to identify relevant features.
- **Data Splitting:**
    - Divide the data into training and testing sets.
    - Use a time-series split to ensure that the model learns from past data and predicts future values.

2. Model Building and Training:

- **Linear Regression:**
    - Create a linear regression model to predict future prices based on historical features.
    - Train the model on the training data.
- **Decision Trees:**
    - Build a decision tree model to capture non-linear relationships between features and the target variable.
    - Tune hyperparameters like maximum depth and minimum samples per leaf.

- **LSTM Neural Networks:**
  - Create a Long Short-Term Memory (LSTM) neural network to model time-series dependencies in the data.
  - Experiment with different architectures and hyperparameters.

3. Model Evaluation:

- **Metrics:**
  - Use metrics like Mean Squared Error (MSE), Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), and R-squared to evaluate model performance.
  - Consider using time-series specific metrics like Mean Absolute Percentage Error (MAPE) and Directional Accuracy.
- **Performance Analysis:**
  - Compare the performance of different models.
  - Visualize predicted vs. actual prices to gain insights.
  - Analyze the model's limitations and potential areas for improvement.

4. Model Deployment and Prediction:

- **Model Deployment:**
  - Deploy the best-performing model to a production environment.
  - Consider using cloud-based platforms or local servers for deployment.
- **Prediction:**
  - Use the deployed model to make predictions on new, unseen data.
  - Generate forecasts for future time periods.

Conclusion:

Department of
Artificial Intelligence & Machine Learning

Laboratory Manual
Subject: - Artificial Intelligence in Finance Laboratory
Semester: - VII Class: - B. Tech. Experiment No. 4

Title: Case Study: Research a historical example of a financial market crash.

Objectives:

To research a historical financial market crash.

To analyse how AI, if it had been present, could have potentially influenced the event.

HARDWARE CONFIGURATION / KIT:

Sr. No	Hardware Configuration	
1	Processor	1.5 GHz or more
2	RAM	512 MB Minimum
3	HDD	Minimum 1GB free Space
4	Internet access	

SOFTWARE REQUIREMENT:

Sr. No	Software Configuration	
1	Operating System	Windows 8 or later
2	Word processor	Office or any

THEORY:

The intersection of artificial intelligence (AI) and finance has become increasingly significant in recent years. As AI technologies continue to advance, their potential impact on financial markets is a subject of growing interest and debate. This paper delves into the historical context of a specific financial market crash, analysing how AI, if it had been advanced enough at the time, could have potentially influenced the event, both positively and negatively. By examining the past, we can gain valuable insights into the future of AI-driven finance and the potential risks and rewards it may bring.

Procedure:

1. **Choose a Historical Market Crash:**
   - Select a significant financial market crash from history. Some potential options include:
     - The 1929 Stock Market Crash
     - The 1987 Black Monday
     - The 2008 Financial Crisis
     - The 2020 COVID-19 Market Crash
2. **Research the Crash:**
   - Gather information about the event, including:
     - **Causes:** What triggered the crash?
     - **Impact:** How did the crash affect the global economy?
     - **Timeline:** What were the key events leading up to and during the crash?
     - **Regulatory Response:** How did governments and regulatory bodies respond to the crisis?
3. **Analyze the Potential Impact of AI:**
   - Consider how AI, if it had been advanced enough at the time, could have potentially influenced the crash.
   - **Positive Influences:**
     - **Early Warning Systems:** AI could have analyzed vast amounts of data to identify early warning signs of market instability.
     - **Improved Risk Assessment:** AI algorithms could have assessed the risks associated with complex financial instruments and strategies.
     - **Enhanced Regulatory Oversight:** AI could have helped regulators monitor market activity more effectively and detect fraudulent behavior.
   - **Negative Influences:**
     - **Algorithmic Trading:** AI-powered trading algorithms could have exacerbated market volatility by triggering rapid buy and sell orders.
     - **Herding Behavior:** AI algorithms could have amplified market trends, leading to "flash crashes" and other extreme market events.
     - **Systemic Risk:** Widespread reliance on AI could have made the financial system more vulnerable to systemic failures.

4. **Write a Report:**
    - Summarize your research on the historical market crash.
    - Discuss the potential positive and negative impacts of AI on the event.
    - Conclude with your overall assessment of how AI might have influenced the course of the crash.
    - Cite your sources using a recognized citation style (e.g., APA, MLA, Chicago).

R. C. PATEL INSTITUTE OF TECHNOLOGY
An Autonomous Institute

Department of
Artificial Intelligence & Machine Learning

Laboratory Manual
Subject: - Artificial Intelligence in Finance Laboratory
Semester: - VII Class: - B. Tech. Experiment No. 5

Title: Develop an algorithm to optimize a portfolio of stocks based on risk and return objectives. Use techniques like mean-variance optimization or Monte Carlo simulation.

Objective:

To develop an algorithm to optimize a portfolio of stocks based on risk and return objectives.

HARDWARE CONFIGURATION / KIT:

Sr. No	Hardware Configuration	
1	Processor	1.5 GHz or more
2	RAM	512 MB Minimum
3	HDD	Minimum 1GB free Space
4	Internet access	

SOFTWARE REQUIREMENT:

Sr. No	Software Configuration	
1	Operating System	Windows 8 or later
2	Google Colab or Jupiter Notebook	
3	Libraries: pandas, NumPy, Matplotlib and historical price data	

THEORY:

Investing in stocks is a delicate balance between risk and return. Investors aim to maximize returns while minimizing risk. Portfolio optimization techniques help achieve this equilibrium.

Key Concepts:

- **Risk:** The potential for loss or variability in investment returns.
- **Return:** The profit or loss generated by an investment over a specific period.
- **Diversification:** Spreading investments across various assets to reduce risk.

Optimization Techniques:

1. **Mean-Variance Optimization (MVO):**
   - **Core Concept:** MVO seeks to construct an efficient frontier, a curve that represents the optimal combination of risk and return for a given set of assets.
   - **Process:**
     1. **Calculate Expected Returns:** Estimate the future returns of each asset.
     2. **Calculate Covariance Matrix:** Measure the relationship between asset returns.
     3. **Optimize Portfolio:** Use mathematical optimization techniques to find the optimal weights for each asset in the portfolio, considering risk tolerance and return objectives.
2. **Monte Carlo Simulation:**
   - **Core Concept:** Monte Carlo simulation involves running multiple simulations to assess potential outcomes under various scenarios.
   - **Process:**
     1. **Generate Random Returns:** Simulate future returns for each asset using probability distributions.
     2. **Create Portfolios:** Construct thousands of random portfolios with different asset weights.
     3. **Evaluate Portfolios:** Calculate the risk and return of each portfolio.
     4. **Identify Optimal Portfolios:** Select portfolios that meet specific risk and return criteria.

Procedure:

1. Data Acquisition and Cleaning:

- **Obtain Historical Data:**
  - Use libraries like yfinance to fetch historical stock prices.
  - Ensure data is clean and complete, handling missing values and outliers.
- **Calculate Returns:**
  - Calculate daily, weekly, or monthly returns using appropriate methods (e.g., simple or log returns).

2. Risk and Return Analysis:

- **Calculate Expected Returns:**
    - Calculate the historical average return for each stock.
- **Calculate Covariance Matrix:**
    - Compute the covariance matrix to measure the relationship between stock returns.
- **Calculate Risk (Volatility):**
    - Calculate the standard deviation of returns for each stock and the overall portfolio.

3. Portfolio Optimization Techniques:

A. Mean-Variance Optimization:

- **Formulate the Optimization Problem:**
    - Define the objective function (e.g., maximizing Sharpe ratio or minimizing portfolio variance).
    - Set constraints (e.g., budget constraints, minimum/maximum weight limits).
- **Solve the Optimization Problem:**
    - Use libraries like scipy.optimize or PyPortfolioOpt to solve the quadratic programming problem.
- **Analyze the Optimal Portfolio:**
    - Evaluate the optimal portfolio's expected return, risk, and Sharpe ratio.
    - Visualize the efficient frontier.

B. Monte Carlo Simulation:

- **Simulate Future Returns:**
    - Use historical return distributions to simulate future returns for each stock.
    - Generate a large number of random portfolios.
- **Evaluate Simulated Portfolios:**
    - Calculate the expected return, risk, and Sharpe ratio for each simulated portfolio.
- **Identify Optimal Portfolios:**
    - Filter portfolios based on specific criteria (e.g., minimum return, maximum risk).
    - Visualize the distribution of simulated portfolios.

4. Backtesting:

- **Historical Backtesting:**
  - Apply the optimized portfolio strategy to historical data.
  - Calculate the performance metrics (e.g., cumulative returns, Sharpe ratio) of the backtested portfolio.
- **Evaluate Performance:**
  - Compare the backtested performance to a benchmark index (e.g., S&P 500).
  - Analyze the sensitivity of the portfolio to different market conditions.

5. Refinement and Iteration:

- **Adjust Parameters:**
  - Experiment with different risk aversion levels, transaction costs, and other parameters.
- **Incorporate Additional Factors:**
  - Consider factors like liquidity, dividend yield, and factor-based models.
- **Rebalance the Portfolio:**
  - Periodically rebalance the portfolio to maintain the desired risk and return profile.

Conclusion:

R. C. PATEL INSTITUTE OF TECHNOLOGY
An Autonomous Institute

Department of
Artificial Intelligence & Machine Learning

Laboratory Manual
Subject: - Artificial Intelligence in Finance Laboratory
Semester: - VII Class: - B. Tech. Experiment No. 6

Title: Credit Risk Assessment.

Objective:

To build a machine learning model to predict the creditworthiness of individuals or companies based on financial and non-financial data.

HARDWARE CONFIGURATION / KIT:

Sr. No	11Hardware Configuration	
1	Processor	1.5 GHz or more
2	RAM	512 MB Minimum
3	HDD	Minimum 1GB free Space
4	Internet access	

SOFTWARE REQUIREMENT:

Sr. No	Software Configuration	
1	Operating System	Windows 8 or later
2	Google Colab or Jupiter Notebook	
3	Libraries: pandas, NumPy, Scikit-learn, Matplotlib and A dataset containing relevant financial and non-financial information	

THEORY:

Credit risk is the potential loss a lender or investor may incur if a borrower fails to meet their debt obligations. Credit risk assessment is the process of evaluating a borrower's creditworthiness to determine the likelihood of default.

Credit Risk Modeling Techniques:

1. **Statistical Models:**
    - **Linear Regression:** Predicts default probability based on historical data and relevant factors.

- **Logistic Regression:** Estimates the probability of default as a binary outcome.
- **Survival Analysis:** Models the time until default.

2. **Machine Learning Models:**
    - **Decision Trees:** Creates a decision tree to classify borrowers as default or non-default.
    - **Random Forest:** Combines multiple decision trees to improve accuracy.
    - **Support Vector Machines (SVM):** Finds the optimal hyperplane to separate default and non-default classes.
    - **Neural Networks:** Learns complex patterns in data to predict default probabilities.

Credit Rating Agencies:

- Assign credit ratings to borrowers (e.g., Moody's, S&P, Fitch).
- Ratings reflect the perceived creditworthiness of the borrower.
- Higher ratings indicate lower credit risk.

Credit Risk Management Strategies:

- **Diversification:** Spreading credit risk across various borrowers and industries.
- **Collateral:** Requiring collateral to secure loans.
- **Credit Limits:** Setting limits on the amount of credit extended to borrowers.
- **Risk-Based Pricing:** Charging higher interest rates to higher-risk borrowers.
- **Credit Insurance:** Transferring credit risk to insurers.
- **Early Warning Systems:** Monitoring borrower behavior and economic indicators.

Procedure

1. Data Acquisition and Cleaning:

- **Load the dataset:** Use Pandas to load the dataset into a DataFrame.
- **Handle missing values:** Employ techniques like imputation (mean, median, mode) or deletion to address missing data.
- **Outlier detection and treatment:** Identify and handle outliers using methods like z-score or IQR.
- **Data normalization:** Scale numerical features to a common range (e.g., min-max scaling or standardization).
- **Feature engineering:** Create new features from existing ones (e.g., debt-to-income ratio, payment-to-income ratio).

2. Exploratory Data Analysis (EDA):

- **Univariate analysis:** Analyze each feature individually (e.g., histograms, box plots).
- **Bivariate analysis:** Examine the relationship between pairs of features (e.g., scatter plots, correlation matrices).
- **Multivariate analysis:** Explore the relationship among multiple features (e.g., principal component analysis).

3. Data Preprocessing:

- **Split the data:** Divide the dataset into training and testing sets.
- **Feature selection:** Identify the most relevant features using techniques like correlation analysis, feature importance, or dimensionality reduction.

4. Model Building and Training:

- **Choose a suitable algorithm:** Consider algorithms like logistic regression, decision trees, random forests, support vector machines, or neural networks based on the problem and dataset characteristics.
- **Train the model:** Fit the chosen algorithm to the training data.
- **Hyperparameter tuning:** Optimize the model's performance by tuning hyperparameters (e.g., learning rate, number of trees, regularization strength).

5. Model Evaluation:

- **Evaluate the model:** Use metrics like accuracy, precision, recall, F1-score, and ROC curve to assess the model's performance on the testing set.
- **Confusion matrix:** Visualize the model's predictions and true labels.

6. Model Deployment:

- **Integrate the model:** Deploy the model into a production environment (e.g., web application, API) to make real-time predictions.
- **Monitor and retrain:** Continuously monitor the model's performance and retrain it with new data as needed to maintain accuracy.

Conclusion:

R. C. PATEL INSTITUTE OF TECHNOLOGY
An Autonomous Institute

Department of
Artificial Intelligence & Machine Learning

Laboratory Manual
Subject: - Artificial Intelligence in Finance Laboratory
Semester: - VII Class: - B. Tech. Experiment No. 7

Title: Risk Management and Back testing Trading Strategies.

Objectives:

To implement a simple trading strategy (Moving Average Crossover) using Python.

To backtest the strategy on historical stock price data.

HARDWARE CONFIGURATION / KIT:

Sr. No	Hardware Configuration	
1	Processor	1.5 GHz or more
2	RAM	512 MB Minimum
3	HDD	Minimum 1GB free Space
4	Internet access	

SOFTWARE REQUIREMENT:

Sr. No	Software Configuration	
1	Operating System	Windows 8 or later
2	Google Colab or Jupiter Notebook	
3	Libraries: pandas, NumPy, Scikit-learn, Matplotlib and historical financial information	

THEORY:

Risk Management

Risk management is a critical aspect of trading. It involves identifying, assessing, and mitigating potential risks to minimize losses and protect capital. Key risk management techniques include:

- **Position Sizing:** Determining the appropriate size of each trade based on risk tolerance and account balance.

- **Stop-Loss Orders:** Pre-setting exit points to limit potential losses if a trade moves against the trader's position.
- **Take-Profit Orders:** Pre-setting exit points to secure profits when a trade reaches a predetermined target.
- **Diversification:** Spreading investments across various assets to reduce risk.
- **Hedging:** Using offsetting positions to reduce exposure to market fluctuations.

Backtesting

Backtesting involves applying a trading strategy to historical market data to evaluate its performance. It helps assess a strategy's profitability, risk, and potential drawdowns. Key benefits of backtesting include:

- **Historical Performance Evaluation:** Assessing a strategy's past performance under various market conditions.
- **Risk Assessment:** Identifying potential risks and optimizing risk management parameters.
- **Strategy Optimization:** Fine-tuning entry and exit signals to improve performance.
- **Identifying Inefficiencies:** Discovering market anomalies or inefficiencies that can be exploited.

Combining Risk Management and Backtesting

By combining risk management principles with rigorous backtesting, traders can develop robust and sustainable trading strategies. Backtesting can help identify optimal risk parameters and evaluate the effectiveness of risk management techniques.

Key considerations for effective backtesting include:

- **Data Quality:** Ensuring accurate and reliable historical data.
- **Transaction Costs:** Accounting for real-world trading costs like commissions and slippage.
- **Overfitting:** Avoiding overfitting the model to historical data by using appropriate validation techniques.
- **Market Regime Changes:** Considering how a strategy might perform in different market environments.

Procedure:

1. Data Acquisition:

- Import Libraries:
- Download Historical Data for stocks, mutual funds, etc.

2. Data Cleaning and Preprocessing:

- Handle Missing Values:
- Calculate Moving Averages like SMA or EMA for long and short durations.

3. Implementing the Trading Strategy:

- **Define the Strategy:**
    - Buy Signal: Short-term MA crosses above the long-term MA.
    - Sell Signal: Short-term MA crosses below the long-term MA.
- **Create a Trading Signal Column in dataframe.**

4. Backtesting the Strategy:

- Simulate Trading:
- Calculate the portfolio's value over time, considering buy and sell signals.
- Account for transaction costs (e.g., brokerage fees) if any
- Calculate Performance Metrics for the testing:

    1. Cumulative Return: Total return over the backtesting period.
    2. Sharpe Ratio: Risk-adjusted return.
    3. Maximum Drawdown: Largest peak-to-trough decline.
    4. Sortino Ratio: Downside risk-adjusted return.

5. Visualizing the Results:

- Plot Stock Price and Moving Averages:
- Plot Cumulative Return:

Conclusion:

R. C. PATEL INSTITUTE OF TECHNOLOGY
An Autonomous Institute

Department of
Artificial Intelligence & Machine Learning

Laboratory Manual
Subject: - Artificial Intelligence in Finance Laboratory
Semester: - VII Class: - B. Tech. Experiment No. 8

Title: Fraud Detection in Financial Transactions.

Objective:

To develop a fraud detection model using machine learning techniques to identify fraudulent transactions in a financial dataset.

To evaluate the model's performance using metrics like precision, recall, and ROC-AUC.

HARDWARE CONFIGURATION / KIT:

Sr. No	Hardware Configuration	
1	Processor	1.5 GHz or more
2	RAM	512 MB Minimum
3	HDD	Minimum 1GB free Space
4	Internet access	

SOFTWARE REQUIREMENT:

Sr. No	Software Configuration	
1	Operating System	Windows 8 or later
2	Google Colab or Jupiter Notebook	
3	Libraries: pandas, NumPy, Scikit-learn, Matplotlib and A financial dataset containing labeled transactions	

THEORY:

Fraud detection in financial transactions involves identifying and preventing fraudulent activities that aim to exploit financial systems for illicit gains. It's a critical task for financial institutions, businesses, and individuals to safeguard their assets and maintain trust.

Core Concepts:

- **Fraudulent Activity:** Any deceptive act or trickery intended to obtain financial advantage illegally.

- **Fraud Detection Systems:** Technologies and methodologies designed to identify anomalous patterns or behaviors indicative of fraud.
- **Machine Learning:** A subset of AI that enables systems to learn from data and make predictions or decisions without explicit programming.

Key Techniques:

1. **Rule-Based Systems:**
   - Define specific rules or thresholds to flag suspicious transactions.
   - Effective for simple, well-defined fraud patterns.
2. **Statistical Analysis:**
   - Analyze historical data to identify outliers or anomalies.
   - Statistical tests (e.g., Z-test, t-test) can be used to assess the significance of deviations.
3. **Machine Learning Algorithms:**
   - **Supervised Learning:** Train models on labeled data to classify transactions as fraudulent or legitimate.
     - **Classification Algorithms:** Decision trees, random forests, support vector machines, neural networks.
   - **Unsupervised Learning:** Identify patterns and anomalies in unlabeled data.
     - **Clustering Algorithms:** Group similar transactions to detect outliers.
     - **Anomaly Detection:** Identify unusual behaviors or transactions.
4. **Feature Engineering:**
   - Create relevant features from raw data to enhance model performance.
   - Examples: transaction amount, time of day, IP address, device information.
5. **Model Evaluation and Deployment:**
   - Assess model accuracy, precision, recall, and F1-score.
   - Deploy models into real-time systems to monitor transactions and trigger alerts.

Challenges and Considerations:

- **Evolving Fraud Tactics:** Fraudsters constantly adapt their techniques, necessitating continuous model updates.
- **Imbalanced Datasets:** Fraudulent transactions are often rare, making it challenging to train accurate models.
- **Privacy and Security:** Protecting sensitive financial data while extracting valuable insights.
- **False Positives and Negatives:** Balancing the trade-off between detecting all fraud and minimizing false alarms.

Procedure:

1. **Data Exploration and Preprocessing:**
    - Load the dataset into a Python environment.
    - Explore the data to understand its structure, missing values, and outliers.
    - Clean the data: handle missing values, outliers, and inconsistencies.
    - Feature engineering: Create new features or transform existing ones to improve model performance.
2. **Data Splitting:**
    - Divide the dataset into training and testing sets.
3. **Model Selection and Training:**
    - Choose a suitable machine learning algorithm (e.g., Random Forest, Logistic Regression, XGBoost, or Neural Networks).
    - Train the model on the training set.
    - Tune hyperparameters to optimize model performance.
4. **Model Evaluation:**
    - Evaluate the model's performance on the testing set using metrics like:
        - **Accuracy:** Overall correctness of predictions.
        - **Precision:** Proportion of true positive predictions.
        - **Recall:** Proportion of actual positive cases correctly identified.
        - **F1-score:** Harmonic mean of precision and recall.
        - **ROC-AUC curve:** Visual representation of model performance.
5. **Model Deployment (Optional):**
    - If applicable, deploy the model to a production environment to monitor real-time transactions.

Conclusion:

Department of
Artificial Intelligence & Machine Learning

Laboratory Manual
Subject: - Artificial Intelligence in Finance Laboratory
Semester: - VII Class: - B. Tech. Experiment No. 9

Title: Time Series Forecasting for Financial Data.

Objective:

To apply time series forecasting models to predict future values of financial indicators and evaluate the accuracy of these models.

HARDWARE CONFIGURATION / KIT:

Sr. No	**Hardware Configuration**	
1	Processor	1.5 GHz or more
2	RAM	512 MB Minimum
3	HDD	Minimum 1GB free Space

SOFTWARE REQUIREMENT:

Sr. No	**Software Configuration**	
1	Operating System	Windows 8 or later
2	Google Colab or Jupiter Notebook	
3	Libraries: pandas, NumPy, Scikit-learn, Matplotlib and A dataset	

THEORY:

Time series forecasting is a statistical method used to predict future values of a variable based on its historical data. In the realm of finance, it's employed to forecast stock prices, exchange rates, interest rates, and other financial time series.

Key Concepts:

- **Time Series Data:** A sequence of observations recorded at regular time intervals.
- **Stationarity:** A time series is stationary if its statistical properties (mean, variance, autocorrelation) remain constant over time. Stationarity is crucial for many forecasting models.
- **Trend:** A long-term upward or downward movement in the data.
- **Seasonality:** A pattern that repeats itself over a fixed period.
- **Cyclicality:** A pattern that repeats itself over an irregular period.

- **Noise:** Random fluctuations in the data.

Common Time Series Forecasting Methods:

1. **ARIMA Models:** AutoRegressive Integrated Moving Average models capture relationships between past values and past errors.
2. **Exponential Smoothing:** Assigns exponentially decreasing weights to past observations, giving more weight to recent data.
3. **Machine Learning:**
    - **Regression Models:** Predict future values based on historical data and other relevant features.
    - **Time Series-Specific Models:**
        - **Long Short-Term Memory (LSTM) Networks:** Capture long-term dependencies in the data.
        - **Gated Recurrent Units (GRU):** A simplified version of LSTM.

Steps in Time Series Forecasting:

1. **Data Preprocessing:** Clean and preprocess the data, handle missing values, and transform the data to stationarity.
2. **Model Selection:** Choose an appropriate model based on the data's characteristics and the desired forecasting horizon.
3. **Model Training:** Fit the model to the historical data to learn patterns and relationships.
4. **Model Evaluation:** Assess the model's performance using metrics like Mean Squared Error (MSE) or Root Mean Squared Error (RMSE).
5. **Forecasting:** Use the trained model to generate future predictions.

Challenges and Considerations:

- **Non-Stationarity:** Non-stationary data can lead to inaccurate forecasts.
- **Outliers:** Outliers can significantly impact model performance.
- **Overfitting:** Models that are too complex may overfit the training data and perform poorly on new data.
- **External Factors:** Economic events, geopolitical factors, and market sentiment can influence financial time series.

Procedure:

Data Acquisition and Preparation:

- **Obtain Financial Data:**
    - Use financial APIs (e.g., Yahoo Finance, Alpha Vantage) to fetch historical data.
    - Consider factors like stock prices, exchange rates, interest rates, etc.

Data Cleaning:

- Handle missing values (e.g., imputation, deletion).
- Address outliers (e.g., winsorization, capping).
- Convert data to appropriate formats (e.g., time series index).

Data Visualization:

- Plot time series to identify trends, seasonality, and cyclic patterns.

Stationarity:

- Ensure time series data is stationary before applying models using Dickey fuller of Advanced Dickey fuller test.
- If not stationary apply techniques like differentiation to make it stationary.

Model Selection and Training:

- **ARIMA Model:**
  - Determine the order of the model (p, d, q) using techniques like ACF and PACF plots.
  - Fit the ARIMA model to the historical data.
  - Forecast future values and calculate error metrics.
- **Prophet Model:**
  - Prepare data in the required format (historical dates and values).
  - Train the Prophet model on the historical data.
  - Forecast future values and calculate error metrics.

Model Evaluation:

- **Error Metrics:**
  - Calculate Mean Squared Error (MSE), Mean Absolute Error (MAE), and Root Mean Squared Error (RMSE).
  - Visualize the actual vs. predicted values to assess model performance.

Model Refinement and Optimization:

- **Feature Engineering:**
  - Create additional features (e.g., technical indicators, economic indicators).
  - Experiment with different transformations (e.g., log transformation, differencing).
- **Hyperparameter Tuning:**
  - Adjust hyperparameters of the models (e.g., ARIMA order, Prophet seasonality and trend components).
- **Ensemble Methods:**
  - Combine multiple models to improve accuracy.

Summary:

- Compare the performance of different models.
- Discuss the limitations and potential improvements.
- Analyze the implications of the forecasts for investment decisions.

Conclusion:

R. C. PATEL INSTITUTE OF TECHNOLOGY
An Autonomous Institute

Department of
Artificial Intelligence & Machine Learning

Laboratory Manual
Subject: - Artificial Intelligence in Finance Laboratory
Semester: - VII Class: - B. Tech. Experiment No. 10

Title: Anomaly Detection in Historical Price Data.

Objective:

To identify anomalous price movements in historical stock price data using statistical and machine learning techniques.

HARDWARE CONFIGURATION / KIT:

Sr. No	Hardware Configuration	
1	Processor	1.5 GHz or more
2	RAM	512 MB Minimum
3	HDD	Minimum 1GB free Space

SOFTWARE REQUIREMENT:

Sr. No	Software Configuration	
1	Operating System	Windows 8 or later
2	Google Colab or Jupiter Notebook	
3	Libraries: pandas, NumPy, Scikit-learn, Matplotlib and a dataset	

THEORY:

Anomaly detection in historical price data involves identifying data points that deviate significantly from expected patterns. These anomalies can signal unusual market behavior, potential errors in data, or significant events affecting the stock.

Key Techniques:

1. **Statistical Methods:**
   - **Z-Score:** Measures how many standard deviations a data point is from the mean. Outliers typically have a Z-score above a certain threshold (e.g., 3).
   - **Box Plot:** Identifies outliers based on quartiles and interquartile range (IQR).
   - **Statistical Process Control (SPC):** Uses control charts to monitor data and detect anomalies.
2. **Machine Learning:**
   - **Isolation Forest:** Isolates anomalies by randomly partitioning data.

- **Local Outlier Factor (LOF):** Identifies anomalies based on their local density deviation.
- **One-Class Support Vector Machine (OCSVM):** Defines a boundary around normal data points, classifying outliers outside this boundary.

3. **Time Series Analysis:**
   - **Time Series Decomposition:** Breaks down time series into trend, seasonal, and residual components. Anomalies can be detected in the residual component.
   - **ARIMA and GARCH Models:** Used to forecast future values. Deviations from these forecasts can signal anomalies.

Applications:

- **Fraud Detection:** Identifying unusual trading patterns.
- **Market Event Detection:** Detecting significant news events or economic indicators.
- **Data Quality Assurance:** Identifying errors or inconsistencies in data.
- **Risk Management:** Identifying potential market risks or extreme events.

Procedure:

1. **Data Acquisition:**
   - Obtain historical stock price data from a reliable source (e.g., Yahoo Finance, Google Finance).
   - Clean the data, handling missing values and outliers as needed.
2. **Exploratory Data Analysis (EDA):**
   - Visualize the time series data using line plots.
   - Calculate summary statistics (mean, median, standard deviation, etc.).
   - Identify potential trends, seasonality, and cyclical patterns.
3. **Statistical Anomaly Detection:**
   - **Z-Score Method:**
     - Calculate the Z-score for each data point.
     - Flag data points with Z-scores exceeding a certain threshold as anomalies.
   - **Bollinger Bands:**
     - Calculate the moving average and standard deviation of the price series.
     - Set upper and lower bands at a specified number of standard deviations from the moving average.
     - Flag data points outside these bands as anomalies.
4. **Machine Learning Anomaly Detection:**
   - **Isolation Forest:**
     - Randomly select a feature and a random split value.
     - Recursively partition the data until all data points are isolated.
     - Anomalies are identified as data points that require fewer partitions to isolate.
   - **One-Class SVM:**
     - Train a support vector machine on normal data points.

        - Anomalies are identified as data points that lie outside the decision boundary.

5. **Model Evaluation:**
    - Confusion Matrix: Evaluate the performance of the anomaly detection models.
    - Precision, Recall, F1-Score: Calculate these metrics to assess the accuracy of the models.
    - ROC Curve: Visualize the trade-off between true positive rate and false positive rate.
6. **Visualization:**
    - Plot the original time series data with detected anomalies highlighted.
    - Visualize the decision boundaries of the machine learning models.

Conclusion:

Expt.	Conclusion
1	In this lab, we have learned how to build and train a simple neural network to predict stock prices. While this is a basic example, neural networks can be further enhanced with techniques like recurrent neural networks (RNNs) and long short-term memory (LSTM) networks to capture temporal dependencies in the data. Remember that stock market prediction is inherently uncertain, and no model can guarantee perfect accuracy.
2	We gained practical experience in EDA using financial data and develop skills for data-driven decision-making in the financial industry by conducting the lab.
3	Predictive modeling for stock prices is a challenging task due to the inherent noise and volatility in financial markets. It is essential to continuously monitor and improve your models as market conditions change.
4	No conclusion.
5	Write in your own words.
6	With Machine Learning one can build a robust credit risk assessment model that can help financial institutions make informed decisions and mitigate credit risk.
7	Using Machine learning we gain valuable insights into the effectiveness of trading strategies and the importance of risk management in investment decision-making.
8	Machine learning skills can be employed to develop a robust fraud detection model that can help financial institutions mitigate risks and protect their customers.
9	Exploring different machine learning and AI techniques, we can gain valuable insights into the behaviour of financial markets and make informed predictions.
10	Exploring different machine learning and AI techniques, we can gain valuable insights into the behaviour of financial markets and make informed predictions.