CAP 4770 - Introduction to Data Science, Fall 2024
Final Project

This document outlines three projects in the fields of traffic, retail, and insurance, which will be evenly distributed among students. Each group will be assigned one of these projects as the final project for this course. For each project, the required deliverables include a 10-page report and the corresponding Python code, with all instructed steps completed.

Please make a single submission for your group through the "People > Group" section in Canvas. The submission should include two PDF files:

1. A PDF of your 10-page report.
2. A PDF of your code (Jupyter Notebook converted to PDF).

Table of Contents

Project A : Traffic

Objective:

This project aims to predict taxi fares based on pickup and drop-off locations, date, time, and other features using a neural network model. The workflow includes data preprocessing, exploratory analysis, data visualization, and building a neural network for fare prediction.

Dataset:

The **NYC Taxi Fare Prediction Dataset** provides over 55 million records of taxi trips in New York City from 2009 to 2015. Each record includes essential features like pickup_datetime, pickup_longitude, pickup_latitude, dropoff_longitude, dropoff_latitude, passenger_count, and the target variable, fare_amount. This dataset is valuable for analyzing urban transportation patterns and building machine learning models to predict taxi fares based on spatial-temporal data. The combination of geographic coordinates and time-based features makes it particularly suited for tasks involving spatial and temporal prediction modeling. You can access the dataset on Kaggle (https://www.kaggle.com/competitions/new-york-city-taxi-fare-prediction/data).

1. Data Preprocessing

- **Load and Clean Data**:
  - Drop rows with missing values and remove outliers based on unrealistic fare amounts or locations (e.g., fares under $2 or pickup/drop-off points outside New York City).
  - Extract features from pickup_datetime, such as hour, day, weekday, and month.
- **Feature Engineering**:
  - **Distance Calculation**: Calculate the distance between pickup and drop-off locations using the Haversine formula.
  - **Geographical Clustering**: Use K-means clustering to cluster the pickup and drop-off locations, assigning each coordinate to a region or cluster.
  - **Time-based Features**: Engineer features like peak hours or rush hours based on time.
- **Normalization**:
  - Normalize or scale numerical features (e.g., distance, latitude, longitude) to improve model convergence.

2. Exploratory Data Analysis (EDA)

- **Univariate Analysis**:
  - Visualize the distribution of target variable (fare_amount) and other key features, such as distance, pickup hour, and day of the week.
- **Bivariate Analysis**:
  - Explore correlations between fare amount and distance, pickup time, and pickup/drop-off locations.
- **Geospatial Analysis**:
  - Plot pickup and drop-off locations on a New York City map to observe hotspot areas for taxi rides and how they relate to fare amounts.

3. Data Visualization

- **Heatmaps**:
  - Create heatmaps showing the concentration of pickups and drop-offs across NYC.
- **Fare Distribution by Time of Day**:
  - Plot average fare amounts over different times of the day or days of the week.
- **Distance vs. Fare Scatter Plot**:
  - Plot a scatter graph of trip distance against fare to identify the relationship and outliers.

4. Model Building: Neural Network Implementation

- **Neural Network Architecture**:
  - Design a neural network with the following layers:
    - Input layer: Takes in features (distance, pickup and drop-off clusters, day of the week, etc.).
    - Hidden Layers: Include a few dense layers with ReLU activation.
    - Output Layer: Single neuron for regression output predicting fare_amount.
- **Training and Validation**:
  - Split the data into training and validation sets.
  - Use Mean Absolute Error (MAE) as the loss function to train the model.
  - Implement early stopping and learning rate scheduling to optimize training.

Hint 1: Be careful for robust preprocessing and tuning of the neural network to avoid overfitting on this dataset.

Hint2: To handle the large dataset, you may need to use distributed frameworks like PyTorch or TensorFlow on GPUs.

5. Model Evaluation and Tuning

- Use the test set to evaluate the model's performance.
- Experiment with hyperparameters (number of layers, learning rate, batch size) to optimize the model.

6. Analysis and Recommendations

1. **Spatial-Temporal Analysis**: Use geospatial clustering to identify high-density zones for pickups and drop-offs. This can reveal popular areas and times, such as airports, tourist locations, or business districts during rush hours.
2. **Peak Demand Analysis**: Analyze fare and trip volume distributions across different times of the day and week. High-fare periods could correlate with typical rush hours or weekends.
3. **Fare Trends Analysis**: Track average fares over different months or years to analyze if there’s seasonality in fare pricing.

Final Report:

A detailed report less than ten pages with the following outline:

1- Introduction: explaining the problem and dataset, and briefly describe your methodology, findings and insights.

2 - Data preprocessing: explaining all steps take for data cleaning and preprocessing, and feature engineering.

3 - Methodology: explaining model architecture, optimization policy, and training process.

4 - Experiments: discussion the process of tunning parameters and evaluating model performance through relevant metrics and error analysis.

5 - Discussion: covering requested tasks on visualizations and analyses, providing insights into spatial-temporal trends, fare distribution, and peak demand. Based on these findings, you may offer actionable recommendations for pricing strategies, driver allocation, and promotional activities.

6 - Conclusion: summarizing you findings and understanding of the problem.

Deliverables:

1. Code base with step-by- step instructions. (.ipynb file, 70 points)
2. Final Report. (.pdf file, 30 points)

Project B : Retail

Objective:

The project's goal is to replicate a real-world situation in which you study consumer data from e-commerce to create predictive models and extract useful insights. Finding trends in consumer behaviour, maximising marketing initiatives, and guiding corporate plans to improve client retention, boost revenue, and customise marketing strategies are the objectives. You can make use of any open source libraries in python.

Dataset:

The UCI Machine Learning Repository's Online Retail II dataset is a real-world dataset that includes transactional data from a UK-based online retailer. It records different facets of retail transactions, including item purchases, customer information, and sales dates. This dataset offers a comprehensive understanding of consumer buying patterns and can be used for a variety of analytics, such as product recommendation systems, sales trends, and customer segmentation. The dataset can be found here:

https://archive.ics.uci.edu/dataset/502/online+retail+ii

Key columns include **InvoiceNo**, **StockCode**, **Description**, **Quantity**, **InvoiceDate**, **UnitPrice**, **CustomerID**, and **Country**.

InvoiceNo is a unique identifier for each transaction, while **StockCode** and **Description** provide product-specific information.

Quantity and **UnitPrice** offer insights into the volume and value of each transaction, allowing for revenue calculations.

CustomerID enables segmentation and customer-based analyses, while **Country** supports geographical segmentation.

Project Components & Key Insights:

1. **Data Cleaning and Exploration:**

    - Analyse historical data, such as customer demographics, browsing sessions, product views, cart additions, and completed sales, as part of the initial dataset analysis.
    - Address missing values, eliminate duplicates, and fix irregularities. To determine the best times to shop, add new options like "Time of Day" and "Day of Week."
    - To uncover significant aspects, use visualisation tools to investigate customer trends, distributions, and correlations.

2. **Customer Segmentation**

    Use clustering techniques (such K-Means or DBSCAN) to divide up your clientele based on their browsing habits, preferences, and purchasing habits.

    - Personalised Marketing: Make use of segments to advise tailored promotions (e.g., offering personalised discounts or suggesting products).
    - Optimised Communication Times: By examining when each consumer category is most engaged, you can ascertain the ideal times to interact with them.

3. **Customer Purchase Prediction:**

    Using information such as time spent on the website, products viewed, and previous purchases, create a classification model to forecast whether a customer's browsing session will end in a purchase.

    - Conversion-Boosting Strategies: To increase conversions, use forecasts to instantly present special offers or free shipping incentives to prospective customers.

    - When to Offer a Discount to Cart Abandoners: Determine the best times and kinds of discounts to entice cart abandoners to finish their purchases.

4. **Sales Forecasting:**

    Use attributes such as goods in a cart, average transaction size, and frequency of purchases to create regression models that predict future sales.

- Seasonal Discount Timing: To ensure steady revenue, pinpoint periods of low sales and recommend well-timed discounts or exclusive deals during off-peak hours.

5. **Product Bundling:**

Using measures like support, confidence, and lift, use association rule mining (such as the Apriori algorithm) to identify products that are frequently purchased together.

- Product Bundling: Provide discounts for combined items to raise the average order size and recommend product bundles based on consumer purchasing patterns.

- Cross-Promotional Opportunities: Find related items and provide real-time, targeted marketing for customers (e.g., "Customers who bought X also bought Y").

Final Report:

A detailed report less than ten pages with the following outline:

1- Introduction: Explain the problem and dataset, and briefly describe your methodology, findings and insights.

2 - Data preprocessing: explaining all steps taken for data cleaning, preprocessing, and feature engineering.

3 - Methodology: Explain model architecture, optimisation policy, and training process.

4 - Experiments: Explain the logic behind using every algorithm that you have used during this project.

5 - Discussion: Cover all requested tasks on visualisations and analyses, providing insights. Based on these findings, you may offer actionable recommendations for strategies, and promotional activities (discounts).

6 - Conclusion: Summarize your findings and understanding of the problem.

Deliverables:

1. **Models for Clustering, Classification, and Regression:** Including code and step-by- step instructions. (30 points)
2. **Association Guidelines and Suggestions:** Code, analysis, and recommendations for cross-promotions and product bundling. (40 points)
3. **Final Report:** An extensive document that includes all methods, conclusions, and practical business suggestions. (30 points)

Project C : Insurance

Objective:

The Vehicle Insurance Risk Profiling and Claim Prediction System aims to build an analytical framework to predict claims and assess risk profiles for insured vehicles. Using machine learning and statistical methods, this project will help insurance providers identify high-risk segments and improve claim prediction accuracy, optimizing underwriting and claims management. Covering the full data science lifecycle—from preprocessing to model development—the project will apply machine learning, clustering, and anomaly detection techniques to deliver insights on risk management and customer segmentation in insurance.

Dataset:

Link to the Dataset: https://www.kaggle.com/datasets/litvinenko630/insurance-claims/data

Key Features:

1. **Policyholder Information:** This includes demographic details such as age, gender, occupation, marital status, and geographical location.
2. **Claim History:** Information regarding past insurance claims, including claim amounts, types of claims (e.g., medical, automobile), frequency of claims, and claim durations.
3. **Policy Details:** Details about the insurance policies held by the policyholders, such as coverage type, policy duration, premium amount, and deductibles.
4. **Risk Factors:** Variables indicating potential risk factors associated with policyholders, such as credit score, driving record (for automobile insurance), health status (for medical insurance), and property characteristics (for home insurance).
5. **External Factors:** Factors external to the policyholders that may influence claim likelihood, such as economic indicators, weather conditions, and regulatory changes.

Data Preprocessing

Objective: Prepare the dataset by cleaning and engineering features to ensure quality inputs for analysis and modeling.

Task 1: **Data Cleaning and Initial Processing**

- **Handle Missing Values:** Identify and handle missing values in numerical and categorical features. Use mean or median imputation for numerical fields and mode for categorical.
- **Standardization of Engine Specifications:** Normalize continuous features to ensure consistent units and scales.

  Encoding Categorical Variables: Use one-hot encoding or label encoding for categorical fields

Task 2: Exploratory Data Analysis (EDA)

- **Claims Distribution Analysis:** Explore how claims (claim_status) vary across different types of vehicles and demographics.
- **Correlation Analysis:** Investigate relationships between continuous features and
  claim_status using correlation heatmaps.
- **Regional Claim Analysis:** Examine region_code and region_density to understand geographic patterns in claims.

  Risk Segmentation

  Objective: Segment customers and vehicles based on risk factors using clustering and anomaly detection techniques.

Task 1: **Customer Segmentation**

- **K-means Clustering:** Group customers into risk profiles based on:
  - Vehicle specs (e.g., vehicle_age, max_power, safety score)
  - Customer demographics (customer_age, region_code, region_density)
  - Policy characteristics (subscription_length)
- **Hierarchical Clustering:** Create nested risk categories within the broader clusters to capture nuanced risk profiles.

Task 2: **Anomaly Detection**

- **Isolation Forest:** Detect unusual vehicle-claim patterns (e.g., high-power vehicles with no safety features having low claim rates).
- **DBSCAN:** Identify outlier claim behaviors in specific regions (e.g., regions with low claim frequency but high claim amounts).

Predictive Modeling

Objective: Develop classification models to predict claim_status and identify key risk factors.

Task 1: Classification Model

- **Develop a Neural Network** to model intricate relationships in high-dimensional data.
- **Feature Importance Analysis:** Identify which features contribute most significantly to predicting claim_status.

Task 2: Model Evaluation

- **Performance Metrics:** Evaluate models using:
  - **Precision-Recall Curves** for imbalanced data insight.
  - **ROC Curve** and **AUC Score** to assess overall performance.
  - **Confusion Matrix** for detailed accuracy and error analysis
  - Pattern Mining

Objective: Identify associations and sequences in claim-related behaviors for improved risk assessment.

Task 1: Association Rule Mining

- **Mining Relationships:** Discover connections between vehicle features and claim frequency.
- **Metrics:** Calculate support, confidence, and lift to quantify rule strength and relevance for patterns like:
  - High-power vehicles with low safety scores having frequent claims.
  - Specific regions with higher claim incidences.

Task 2: Sequential Pattern Analysis

- **Subscription Analysis:** examine how claim pattern evovles over subscription_length.
- **High-Risk Feature Combinations:** Identify combinations of features (e.g., old vehicle with no airbags in high-density regions) that correlate with higher claims

Final Report:

A detailed report less than ten pages with the following outline:

1. **Introduction:** Explain the problem and dataset, and briefly describe your methodology, findings and insights.
2. **Data preprocessing:** explaining all steps taken for data cleaning, preprocessing, and feature engineering.
3. **Methodology:** Explain model architecture, optimisation policy, and training process.
4. **Experiments**: Explain the logic behind using every algorithm that you have used during this project.
5. **Discussion**: Cover all requested tasks on visualisations and analyses, providing insights.
6. **Conclusion:** Summarize your findings and understanding of the problem.

Deliverables:

2. **Models for Clustering, Classification and Neural Network:** Including code and step-by-step instructions. (40 points)
3. **Pattern Mining:** Code, analysis, and recommendations for cross-promotions and product bundling. (30 points)
4. **Final Report:** An extensive document that includes all methods, conclusions, and practical business suggestions. (30 points)