Matthew Yankovsky (my518)
Roee Shalom
Professor Gunawardena
198:210:06 Data Management for Data Science

CS 210 : Data Management in Data Science Course Project Proposal

Project Name: Credit Card Fraud Detection Proposal (link to dataset)

Project Definition: Our project tackles the growing issue of credit card fraud, which poses a serious threat and results in significant financial losses for both financial institutions and consumers. Credit card companies need accurate and reliable methods to identify fraudulent transactions amidst large volumes of legitimate activity. This is challenging due to the sheer volume of transactions and the very small percentage that are actually fraudulent. To address this, we aim to develop a fraud detection system that can effectively differentiate between fraudulent and legitimate transactions. Our approach will leverage advanced data processing techniques to handle large datasets and a highly imbalanced class distribution, as well as machine learning models designed to detect subtle patterns indicative of fraud. Through careful data management, preprocessing, and predictive modeling, our project will contribute to building a robust system for early fraud detection, ultimately improving financial security for consumers and businesses.

Our project closely connects to the CS210 course as we address essential data management tasks, such as data collection, transformation, and cleaning. Since the dataset includes PCA-transformed features without direct context, we will need to perform careful preprocessing, like handling missing values and detecting outliers, echoing course topics on data handling. The high imbalance in fraud cases requires us to use advanced evaluation metrics, such as the (AUPRC), instead of simple accuracy, aligning with our course focus on meaningful metrics for unbalanced data. We will also apply SQL database management to organize and query our data efficiently, reinforcing essential skills in data storage and retrieval. Lastly, this project lets us apply predictive models, like logistic regression, to a real-world problem, connecting course concepts on model selection, performance assessment, and practical deployment.

- **Data Management**: We will design efficient data handling processes for collecting, storing, and preprocessing credit card transaction data, ensuring its readiness for analysis and machine learning application.
- **Machine Learning and Evaluation:** The model will be trained to classify transactions as fraudulent or legitimate, leveraging techniques suited for high-imbalance data.
- **Database and Visualization:** Our system will use SQL databases for structured data storage, facilitating efficient data retrieval and analysis. Additionally, we will develop visualizations to provide insights into fraud patterns.

Novelty & Importance: Our project addresses the critical issue of credit card fraud, a growing concern due to increasing online transactions and digital payments. Accurately detecting fraudulent activities within massive amounts of transaction data is essential to prevent financial losses and protect consumers. We chose this project because of our interest in financial data, and felt that it was an appropriate sized project for the two of us to handle. While both of us have experience with individual projects that involve classification and data management techniques, we have not yet worked on something as big as fraud detection. This project also highlights existing challenges in data management, particularly in handling imbalanced data and ensuring model interpretability, which are relevant to course topics in data science and data integrity.
Project Plan:

- **Data Collection**
  - **Data Source**: We will use the Kaggle "Credit Card Fraud Detection" dataset, which contains anonymized transaction data for European cardholders, including a target variable indicating fraud.
  - **Data Characteristics**: This dataset includes over 280,000 transactions, with about 0.17% labeled as fraudulent. The features have undergone PCA transformation, except for `Time` and `Amount`.
- **Data Preprocessing and Balancing**
  - **Steps**: Clean the data by checking for any missing or inconsistent values. Address class imbalance, as the dataset has a high number of legitimate transactions compared to fraudulent ones.
  - **Balancing Technique**: Apply the Synthetic Minority Oversampling Technique (SMOTE) to generate synthetic samples of fraudulent transactions, creating a balanced dataset.
- **Feature Engineering**
  - **Steps**: Analyze important features (`Time`, `Amount`) and apply scaling. We will create additional time-based features (e.g., hour of day, day of week) to uncover patterns that may help distinguish fraud.
  - **Purpose**: This step improves model interpretability and can reveal trends that distinguish fraud from legitimate transactions.
- **Data Visualization**
  - **Steps**: Create initial visualizations to explore patterns within the data and compare fraudulent vs. non-fraudulent transactions.
  - **Class Distribution**: Use bar charts to visualize the class imbalance between fraudulent and non-fraudulent transactions.
  - **Transaction Amount**: Plot histograms or box plots to examine transaction amounts, comparing distributions across classes.
  - **Time Features**: Create line plots or heatmaps showing transaction counts by hour or day of the week to detect any time-based patterns in fraud.
  - **Tools**: Use Matplotlib or Seaborn for straightforward visualization.

- **Model Selection and Training**
  - **Model**: Implement a Random Forest Classifier, which is well-suited for handling complex data structures and prevents overfitting.
  - **Training Steps**: Train the model using the balanced data from SMOTE and perform hyperparameter tuning to optimize accuracy and recall.
- **Evaluation and Success Metrics**
  - **Evaluation Techniques**: Precision, recall, and F1-score, focusing on precision and recall to address the class imbalance. Confusion matrix to understand the distribution of true/false positives and negatives.
  - **Visualization**: Plot Precision-Recall curves to assess model performance at different thresholds, ensuring the model effectively detects fraud without overwhelming false positives.
- **Testing and Validation**
  - **Testing Method**: Apply the model on a holdout test set to confirm generalizability.
  - **Success Measure**: Success will be determined by achieving high precision and recall values, indicating the model's accuracy in detecting fraud without excessive false positives.
- **Deployment and Saving**
  - **Steps**: Save the model using `pickle` or `joblib` for reusability, allowing us to reload it for testing or deployment.
  - **Documentation**: Summarize findings, key features, and model performance, with insights for practical fraud detection applications.