Hacettepe University

Computer Engineering Department

BBM479 End of Term Development Report

Project Details

Title	Credit Card Fraud Detection Using ML Techniques & Comparing
Supervisor	Dilmurod Vahabdjanov

Group Members

	Full Name	**Student ID**
1	İlkim İclal Aydoğan	21992814
2	Diren Boran Sezen	21946553
3	Hasan Çağrı Sarıkaya	2200356852
4		

Current State (/ 40 Points)

Explain the current state of the project. By giving references to the plan proposed at the beginning of the project, explain what is achieved so far. Provide details of what has been done by explaining the technology and methodology used. Is the current state of the project inline with the plan? If you are behind the schedule, explain in detail the reasons.

We've completed the ***Researching and Learning***, ***Selecting, Analyzing, and Organizing the dataset***, ***Implementing and Training the dataset with various algorithms and different parameters***, ***Finalizing the models, metrics, and graphs***, ***Analyzing the results*** and ***Debugging and Testing, Preparing demo*** steps that were in our plan.

1 - Researching and Learning - 3 weeks

During the initial three weeks, each team member delved into academic papers and theses focused on credit card fraud detection using machine learning models. Additionally, we familiarized ourselves with the operations of the ML algorithms we intended to employ and gained proficiency in Python libraries like pandas, numpy, and scikit-learn.

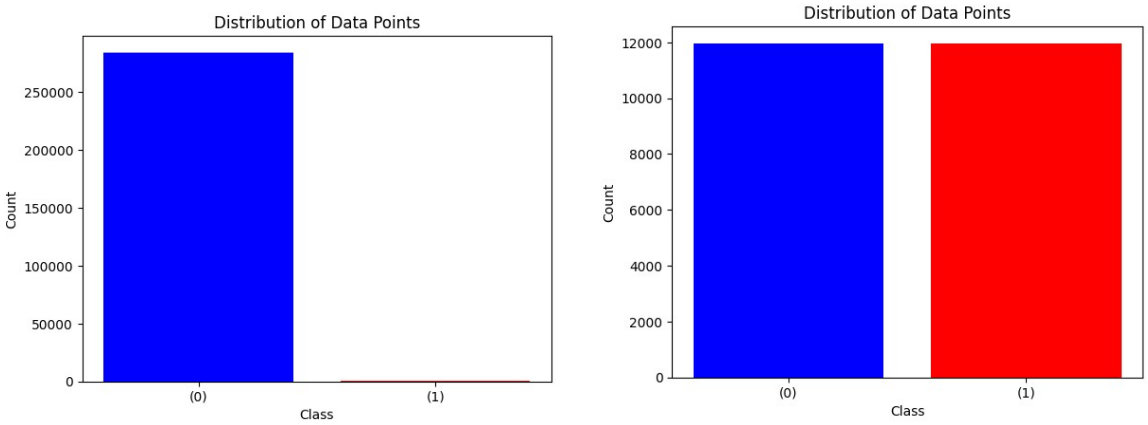
2 - Selecting, Analyzing and Organizing the dataset - 1 week

Upon conducting our research and exploring 2-3 datasets, we observed a limited availability of publicly accessible datasets for this particular issue. Consequently, we opted for the dataset available on Kaggle, accessible via the link: Kaggle Credit Card Fraud Dataset. This dataset was chosen due to its inclusion of real-life transactions and its substantial size.

3 - Implementing and Training the dataset with various algorithms and different parameters - 2 weeks

Once we finalized the dataset, addressing its skewed distribution became a primary concern, given the disproportionate number of fraud cases relative to genuine transactions. After exploring various sampling techniques, we chose OneSidedSelection for undersampling and SMOTE for oversampling. However, we encountered challenges, particularly with SVM, as SMOTE exhibited suboptimal performance and posed significant computational time issues. To enhance performance and maintain reasonable computational efficiency, we experimented with combining SMOTE with other undersampling methods.

On the left, we have the original dataset, while on the right, we see the dataset post-OneSidedSelection and SMOTE transformations.




```
In [9]: def undersampling(X_train, y_train):
            undersampling = OneSidedSelection(n_neighbors=3, n_seeds_S=200)
            X_train, y_train= undersampling.fit_resample(X_train, y_train)
            return X_train,y_train
```

```
In [10]: def oversampling_SMOTE(X_train, y_train):
             oversample = SMOTE(sampling_strategy='auto')
             X_train, y_train = oversample.fit_resample(X_train, y_train)
             return X_train,y_train
```

```
In [11]: def SMOTE_tomek(X_train, y_train):
             oversample = SMOTE(sampling_strategy='auto')
             X_train, y_train = oversample.fit_resample(X_train, y_train)
             undersample = TomekLinks()
             X_train, y_train = undersample.fit_resample(X_train, y_train)
             return X_train,y_train
```

4 - Finalizing the models, metrics and graphs, Analyzing the results - 2 weeks

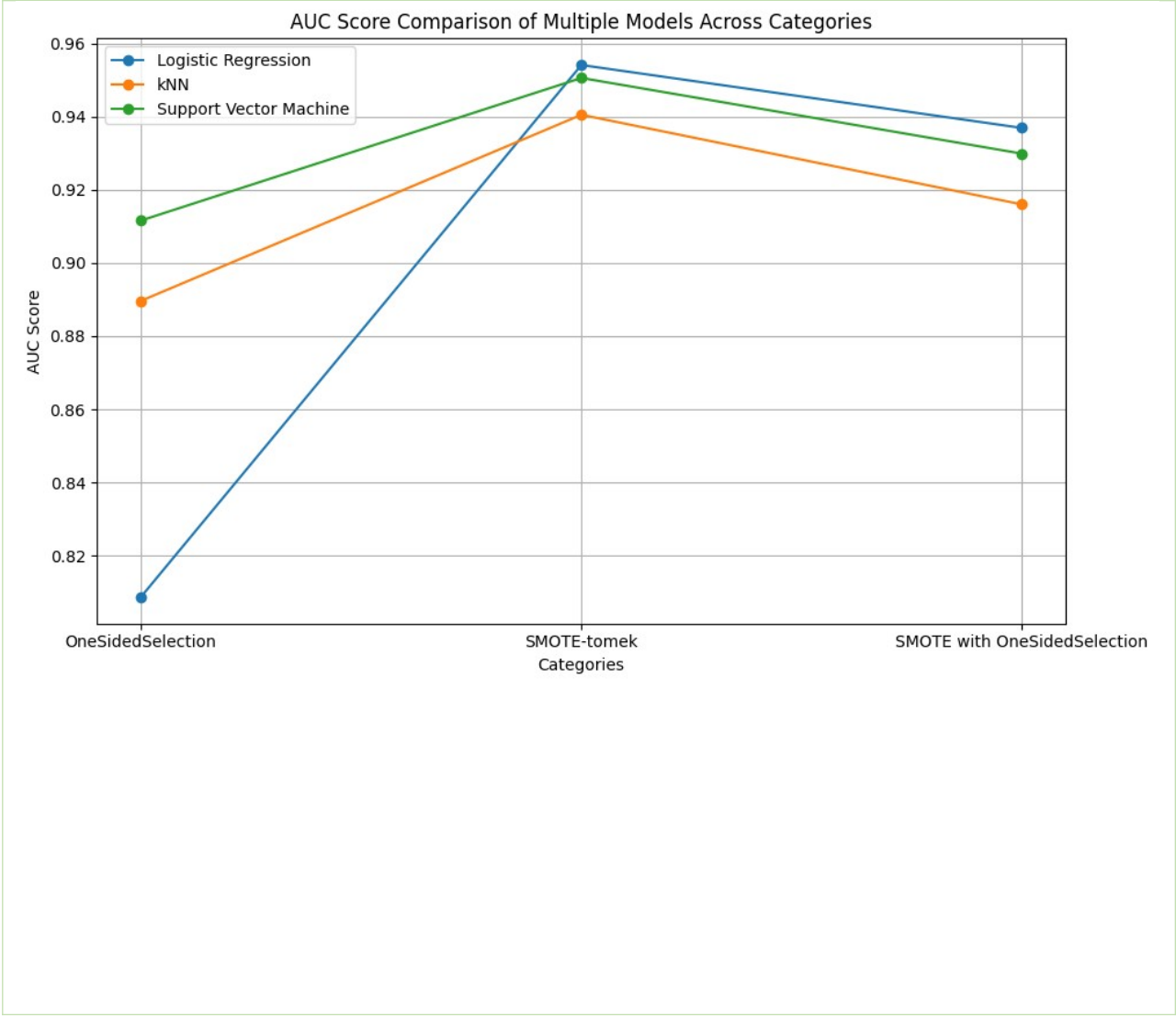
Following the sampling process, in accordance with the problem definition, we employed three distinct models: Logistic Regression, K-Nearest Neighbors (KNN), and Support Vector Machine (SVM).

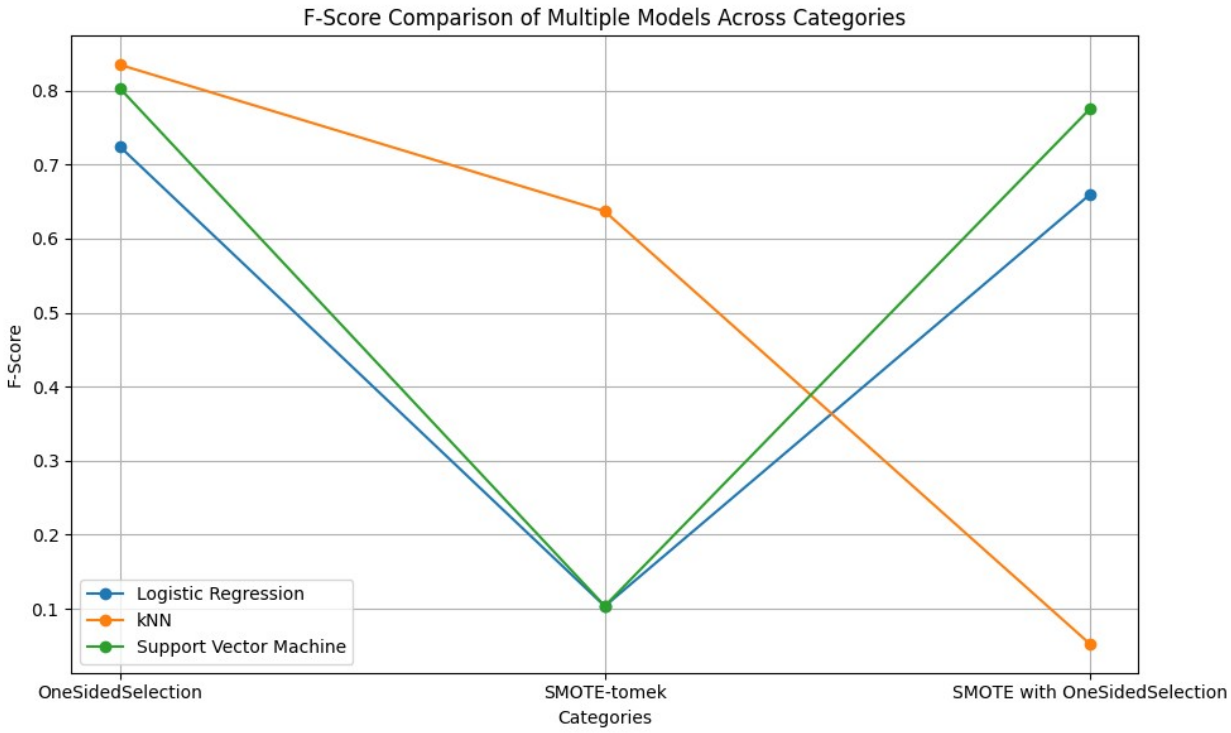
From the accompanying images, you can observe the implementations we've executed. After numerous iterations and experiments, we settled on specific parameters and model configurations. Subsequently, we analyzed and documented the outcomes of our models.

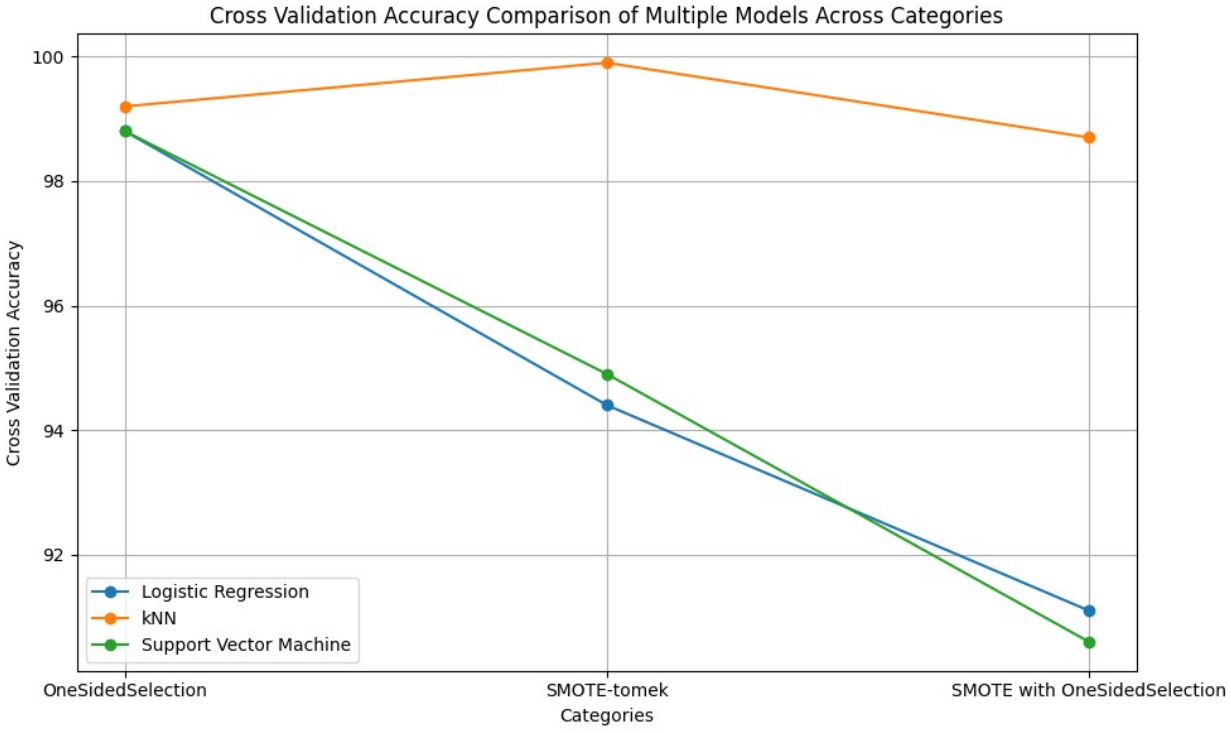
In [12]:
```
def logistic_reg_model(X_train, y_train, X_test):
    LR_model = LogisticRegression(max_iter=100000)
    LR_model.fit(X_train, y_train)
    print(LR_model.score(X_train, y_train))
    pred = LR_model.predict(X_test)
    return pred
```

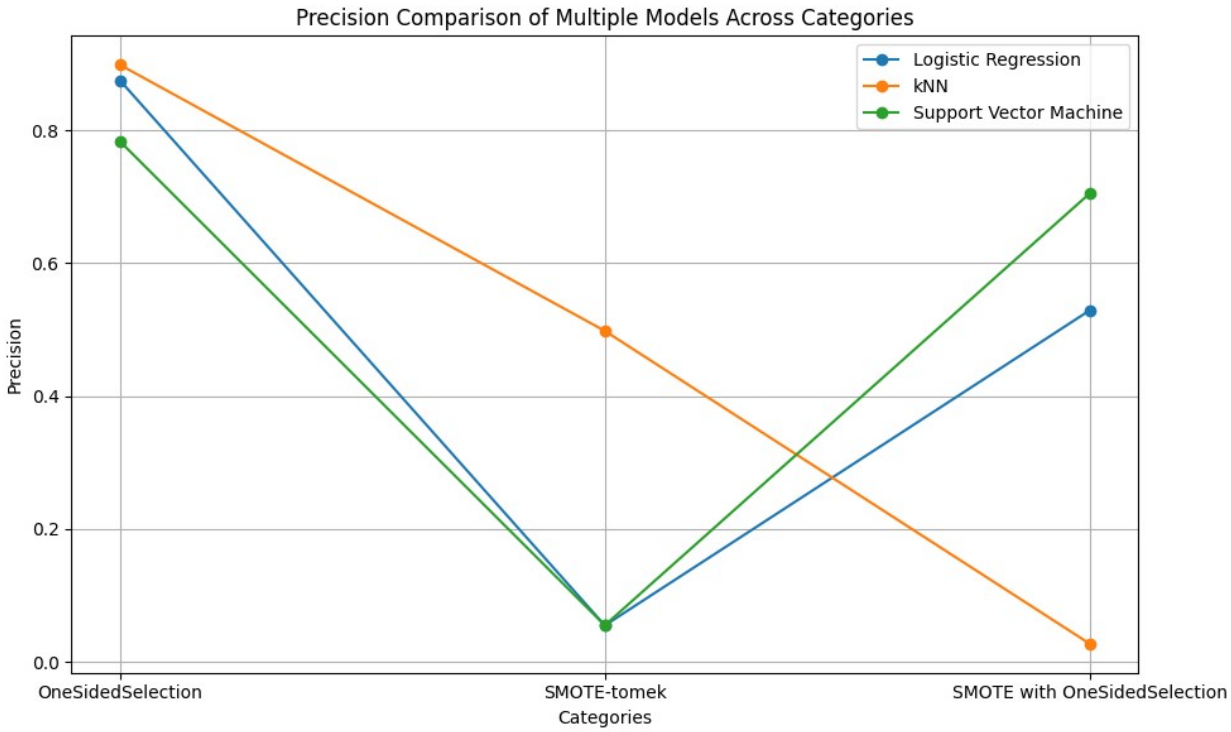
In [13]:
```
def knn_model(X_train, y_train, X_test):
    knn = KNN(n_neighbors=3)
    knn.fit(X_train, y_train)
    print(knn.score(X_train, y_train))
    pred = knn.predict(X_test)
    return pred
```

In [14]:
```
def SVM_model(X_train, y_train, X_test):
    svm_classifier = SVC(kernel='linear', C=1.0, random_state = 42)
    svm_classifier.fit(X_train, y_train)
    print(svm_classifier.score(X_train, y_train))
    pred = svm_classifier.predict(X_test)
    return pred
```


AUC Score Comparison of Multiple Models Across Categories
Logistic Regression
kNN
Support Vector Machine
AUC Score
0.96
0.94
0.92
0.90
0.88
0.86
0.84
0.82
OneSidedSelection
SMOTE-tomek
SMOTE with OneSidedSelection
Categories


F-Score Comparison of Multiple Models Across Categories
F-Score
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
Logistic Regression
kNN
Support Vector Machine
OneSidedSelection
SMOTE-tomek
SMOTE with OneSidedSelection
Categories

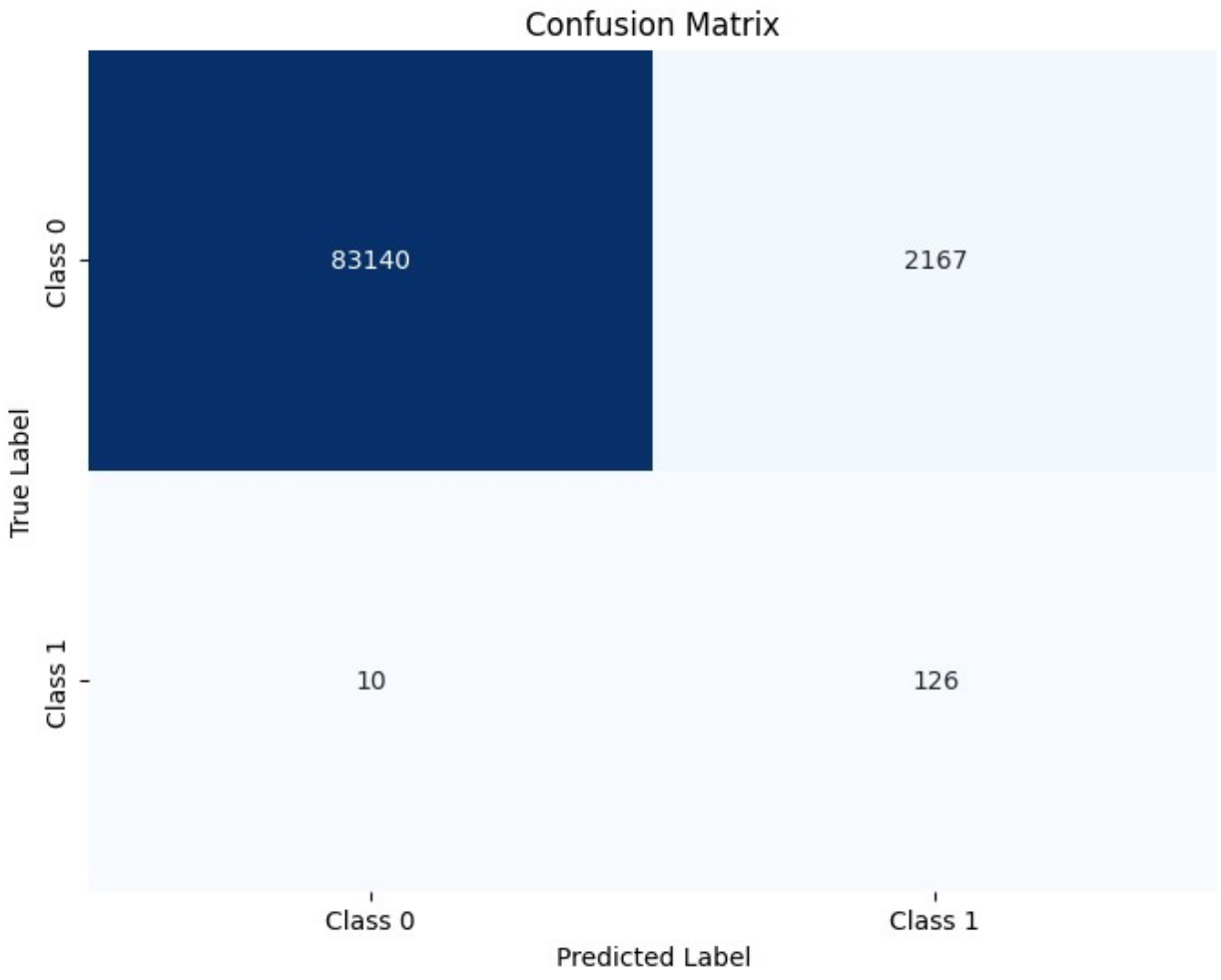

Cross Validation Accuracy Comparison of Multiple Models Across Categories
Cross Validation Accuracy
100
98
96
94
92
Logistic Regression
kNN
Support Vector Machine
OneSidedSelection
SMOTE-tomek
SMOTE with OneSidedSelection
Categories



5 - Debugging and Testing, Preparing demo - 1 week

Upon evaluating the outcomes, we noticed that the model employing a combination of SMOTE and OneSidedSelection sampling exhibited superior performance compared to others. For standard metrics such as precision and F-score, the undersampled dataset produced consistent results across all models due to its limited size. However, there was significant variability in the AUC scores among the models. Interestingly, when utilizing OneSidedSelection with SMOTE, the AUC scores remained relatively consistent across different models, contrasting sharply with other metric disparities. Given the limited number of fraud cases, we can assess the model performance primarily by examining its accuracy using confusion matrices to determine correct and incorrect classifications.

Analyzing the confusion matrices, specifically for class 1 (representing fraud), the optimal outcomes were observed with SMOTE Tomek sampling combined with SVM. Both KNN and Logistic Regression manifested nearly identical confusion matrices, differing only by a minor margin of one or two data points.



Continuous Learning (/ 25 Points)

You have been working on the problem for almost 3 months. In these 3 months, what did you learn about this problem that you didn't know at the beginning? Did this new knowledge change your perspective on the problem? What else do you need to learn in the future?

Initially, our team approached the problem of fraud detection with a mindset typical to other machine learning challenges. However, as we delved deeper into the fraud detection task, we quickly realized unique aspects that distinguished it from generic machine learning tasks. Despite some team members possessing prior experience in the domain, the details we encountered necessitated collective learning and adaptation.

Our key learnings were:

Understanding of Fraud Detection in Real-world Contexts: We delved into and learned the traditional mechanisms and methodologies employed in fraud detection before the advent of machine learning. This foundational knowledge was important for us to implement a machine-learning model in this domain.

Class Imbalance in Fraud Detection Datasets: One of the standout challenges was recognizing and addressing the skewed class distribution inherent in fraud detection datasets. Traditional algorithms like SVM exhibited subpar performance in such scenarios. Consequently, we undertook an in-depth study of sampling techniques and learned techniques like undersampling, oversampling, and SMOTE. These emerged as pivotal solutions. Through rigorous experimentation, we honed our understanding of when and how to deploy these techniques to optimize model performance.

Mathematical Underpinnings of Key Algorithms: We spent time understanding and learning how certain computer methods, K-Nearest Neighbors (KNN), Logistic Regression, and Support Vector Machines (SVM) do their job. This helped us pick the right tools for our task.

Reevaluation of Performance Metrics: We learned that just looking at accuracy doesn't tell the whole story in fraud detection. The skewed distribution of data rendered traditional metrics, such as accuracy, inadequate for gauging model efficacy. So we researched other methods. Through our research and experimentation, we decided on the Area Under the Receiver Operating Characteristic Curve (AUC-ROC) scores and F-scores. These metrics provided nuanced

insights, capturing the model's performance more holistically in the context of class imbalances.

Literature Review and Industry Practices: We read up on what experts and others have done in the past for fraud detection. This involved studying the evolution of models, their real-world applications, challenges encountered by professionals, and best practices adopted in the industry. Thanks to this we got smarter about our own work and realized the places we lacked.

We probably need to learn more about the new models we are going to implement. Decision Tree and Random Forest Classifier.

Risk Assessment and Management (/ 20 Points)

Try to identify the potential risks in the rest of the project. Were you able to identify these risks at the beginning of the project, or did you recently recognize them? What are your proposed solutions to each risk item? Are there any risks that will require a significant change in the project? If so, explain how this will affect the end results (also, outline your proposed revisions in the next section).

1. Dataset Limitations:

- Identification: Recognized early in the project during the data handling phase.

- Risk: The Kaggle dataset may have limitations in representing diverse fraud scenarios, potentially affecting the models' generalizability.

- Solution: Explore additional datasets or generate synthetic data to supplement the existing dataset. Conduct thorough sensitivity analysis to understand the impact of dataset variations.

2. Class Imbalance:

- Identification: Acknowledged at the project outset.

- Risk: The significant class imbalance may lead to biased models, as the majority of transactions are legitimate, making it challenging for the models to identify fraudulent ones effectively.

- Solution: Implement techniques such as oversampling, undersampling, or synthetic data generation to address class imbalance. Carefully evaluate model performance metrics to ensure

a balanced assessment.

3. Algorithmic Complexity:

- Identification: Recognized during the implementation of machine learning algorithms.

- Risk: The complexity of algorithms like Support Vector Machine (SVM) may lead to longer training times, potentially affecting the overall efficiency of the project.

- Solution: Optimize algorithm parameters, consider algorithm-specific optimizations, and explore parallel processing to mitigate training time challenges.

4. Dynamic Fraud Patterns:

- Identification: Identified as a potential risk during the comparative analysis.

- Risk: The models may not adapt well to evolving fraud patterns, limiting their effectiveness in real-world scenarios.

- Solution: Develop a mechanism for dynamic model training that can continuously learn and adapt to changing fraud patterns. Regularly update the models based on new data patterns.

Revisions (/ 15 Points)

If you feel like you need to revise your earlier plan, suggest your changes here. These changes may include changes in outcomes of the project, changes in milestones, changes in calendar, changes in workload distribution, etc. If you do not present any revisions here, at the end of the project you will be responsible for all the proposed outcomes in the Project Proposal.

We successfully achieved all our objectives for this semester. Moving forward to the next semester, after implementing the Random Forest Classifier and Decision Tree Classifier, we might explore acquiring a more refined dataset. Given our enhanced experience, we anticipate completing these models more efficiently, allowing us to potentially focus on refining our modeling approach further and explore more different approaches, like combining models or implementing a cost function.