MVE 441 RE-EXAM Report

Fanze Meng

1. Question 1a

1.1 The impact of imbalance & Smote technique & Model training

The data indeed has an imbalance issue, with only 39 labels “7” compared to 457 labels “2”. To explore the impact of imbalance, I documented the entire process of training and tuning based on the original data, using a test size of 20% as an example. For six different models (logistic regression, SVM, KNN, Decision Tree, Random Forest, MLP Classifier), I recorded the cross-validation accuracy, ROC AUC, sensitivity, and specificity for different hyperparameters.

Table 1 Classifier performance for data imbalanced without using SMOTE

Classifier (Data without SMOTE)	Accuracy	ROC AUC	Sensitivity	Specificity
Logistic Regression	0.775385	0.933684	0.970053	0.924422
SVM(RBF Kernel)	0.771077	0.934178	0.971975	0.932800
K-Nearest Neighbors	0.631385	0.808994	0.951742	0.782339
Decision Tree	0.747077	0.798697	0.976158	0.923188
Random Forest	0.788923	0.935018	0.975096	0.943009
MLP Classifier	0.741538	0.917995	0.956001	0.901774

Cross-Validation Accuracy across the classifiers shows that the SVM and Random Forest achieve the highest accuracy (~93%), followed by MLP, Decision Tree, and Logistic Regression. KNN has the lowest accuracy (~75%), indicating it may struggle more with this dataset.

To address this imbalance, I tried using SMOTE’s oversampling method and experimented with various settings (whether to oversample data to equal numbers for each class or to a specific value). For the data that was oversampled (i.e., rebalanced), I conducted similar operations as above.

Table 2 Classifier performance for data balanced through SMOTE

Classifier (Data with SMOTE)	Accuracy	ROC AUC	Sensitivity	Specificity
Logistic Regression	0.769621	0.952740	0.984968	0.921310
SVM (RBF Kernel)	0.887782	0.982419	0.963834	0.953279
K-Nearest Neighbors	0.825576	0.958161	0.899331	0.950363
Decision Tree	0.828701	0.900057	0.951199	0.945026
Random Forest	0.897781	0.988041	0.981106	0.956797
MLP Classifier	0.887153	0.981899	0.932100	0.953751

SMOTE improves overall accuracy, ROC AUC, and specificity for all classifiers, demonstrating its effectiveness in balancing the dataset and enhancing classification performance. While sensitivity decreases slightly in some models, the gain in specificity and overall accuracy indicates that SMOTE helps reduce overfitting and leads to better generalization. Random Forest and SVM (RBF Kernel) perform best with SMOTE, showing the highest accuracy, ROC AUC, and balanced sensitivity and specificity. KNN benefits greatly from SMOTE, addressing its poor performance on imbalanced data and achieving a much more balanced classification performance.

1.2 Model evaluation at overall and class level

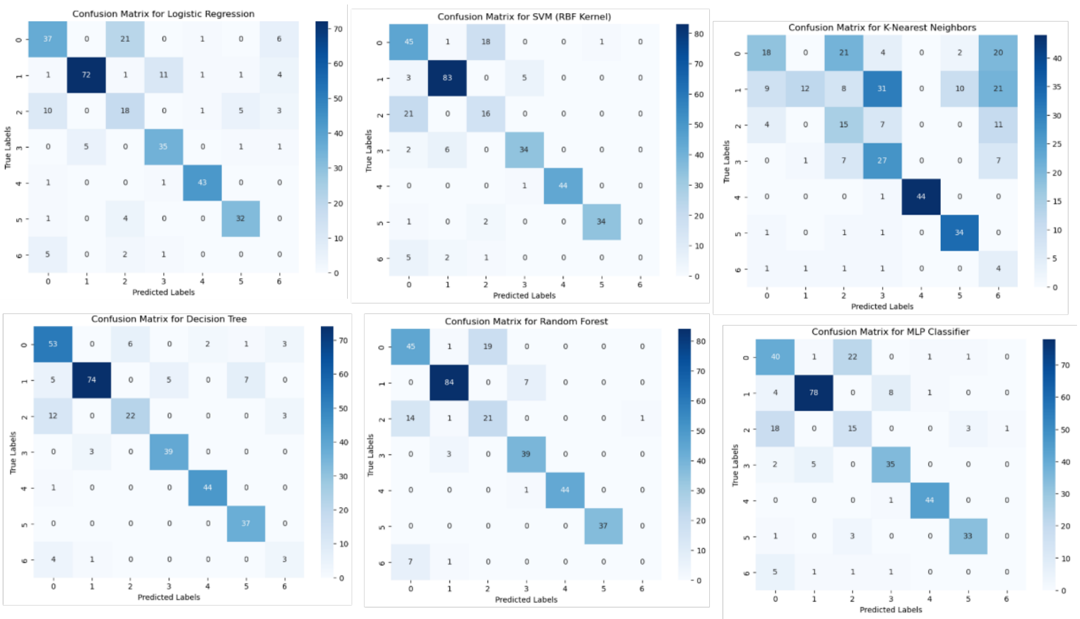

Figure 1 Confusion matric for the classifiers (Logistic Regression, SVM, KNN, Decision Tree, Random Forest, MLP Classifier)

As for class separation, **“2”, “5”**, and **“6”** are well-separated across most classifiers, with high precision and recall. These classes likely have distinct features that the models can easily identify. Typically, KNN has a lower ability on separating **“2”** compared to the other classifiers.

“7” is particularly problematic, showing near-zero precision, recall, and F1-score across multiple classifiers, indicating it is not well-separated from other classes. **“1”** and **“3”**, and **“4”** also exhibit lower precision and recall in some classifiers, indicating potential overlap with other classes.

1.3 Section Summary

Although SMOTE (Synthetic Minority Over-sampling Technique) has demonstrated significant effects in mitigating class imbalance during my experiments, I observed that more extensive SMOTE oversampling does not consistently yield optimal performance on test data. Upon further investigation, I found that there is no standardized approach for determining the degree of oversampling in techniques such as SMOTE. In the context of multi-class datasets, resampling to equalize the number of instances across all classes may seem to fully address the imbalance issue. However, this approach becomes questionable if there is mislabeling within the dataset. Additionally, excessive resampling can lead to a "dilution" of the original data, potentially compromising its integrity. In cases of data imbalance, modifying evaluation metrics—since accuracy may no longer be a reliable measure—and applying oversampling techniques are two key strategies to address the issue.

2. Question 1b

2.1 Feature selection & confidence

I employed three different types of feature selection methods: wrapper-based (forward greedy search), filter-based (ANOVA), and embedded (LASSO), conducting feature selection both at the overall level and at the class level. At the class level, the data was transformed into a binary format.

Firstly, train classifiers using the features selected by each method. Compare performance metrics (accuracy, ROC AUC, sensitivity, specificity) to determine which feature subsets lead to the best classification performance. Then analyze whether different classifiers prefer different feature subsets. Check if some features are consistently selected across different classifiers. And finally assess if different features are important for different classes. Evaluate how well the selected features perform for each class individually.

Table 3 Primary selected features between different methods

Class	Methods	Selected Features	CV Accuracy
"1"	LASSO	[X4, X5]	0.83460
	ANOVA	[X5, X10, X77]	0.87596
	Wrapper-LDA	[X2, X3, X5]	0.87070
"2"	LASSO	[X1, X3, X4]	0.91208
	ANOVA	[X4, X50, X80]	0.93057
	Wrapper-LDA	[X2, X3, X4]	0.92348
"3"	LASSO	[-]	-
	ANOVA	[X1, X3, X2]	0.88544
	Wrapper-LDA	[X1, X2, X5]	0.88743
"4"	LASSO	[X4]	0.90475
	ANOVA	[X4, X2, X3]	0.92978
	Wrapper-LDA	[X2, X3, X4]	0.92523
"5"	LASSO	[X2, X4, X5]	0.93779
	ANOVA	[X2, X3, X5]	0.98625
	Wrapper-LDA	[X1, X2, X5]	0.99649
"6"	LASSO	[X3, X5]	0.92855
	ANOVA	[X5, X3, X2]	0.97763
	Wrapper-LDA	[X1, X3, X5]	0.98856
"7"	LASSO	[-]	-
	ANOVA	[X5, X74, X53]	0.97673
	Wrapper-LDA	[X5, X25]	0.98241
Overall	LASSO	[X5, X4, X3]	0.82894
	ANOVA	[X3, X2, X1]	0.85033
	Wrapper-LDA	[X1, X3, X4]	0.84257

The methods did not show any particular consistency in feature selection, though the wrapper methods based on LDA displayed some consistency. Using ANOVA for feature selection did not prove effective enough. Even with a stringent p-value threshold ($p < 0.001$), it often failed to filter out even a single feature, perhaps because ANOVA does not consider the model itself but focuses solely on statistical properties.

At the class level, I calculated the average recall value for the three feature selection methods. The results indicated a relatively low confidence in the feature selection for “3” and “1”, especially for “1”.

Table 4 Average CV Recall by Class

Class	Mean CV Accuracy
“1”	0.86042
“2”	0.92204
“3”	0.88644
“4”	0.91992
“5”	0.97351
“6”	0.96491
“7”	0.97957

2.2 Section Summary

Different classifiers may prefer different feature subsets due to their inherent characteristics and the way they handle features. For example, tree-based methods might prioritize features differently compared to linear models. Feature importance can vary across classes. It’s crucial to analyze class-specific performance to understand which features contribute most to each class. Higher confidence is achieved when multiple methods agree on the importance of features and when performance metrics are consistently high across different feature sets.

3. Question 1c

3.1 How the results differ between methods

Firstly, I choose Linear Dimension Reduction Techniques (Singular Value Decomposition (SVD), Sparse SVD, Principal Component Analysis (PCA) and Nonlinear Dimension Reduction Techniques (Kernel PCA, Non-negative Matrix Factorization (NMF), Autoencoders (AE), t-SNE). And choose ANOVA, Lasso, Wrapper-based methods (LDA) as the Feature Selection Methods, which are set as the above.

The results are shown in Table 5.

Table 5 Dimension Reduction Performance Comparation

Method	Accuracy	Recall	Precision	F1-Score
SVD	0.782787	0.663493	0.691745	0.659451
PCA	0.77459	0.653356	0.669817	0.641202
Kernel PCA	0.760246	0.640137	0.65405	0.629406
NMF	0.491803	0.430172	0.377192	0.364969

Autoencoder	0.713115	0.628387	0.622031	0.621944
t-SNE	0.268443	0.268011	0.243297	0.244251
ANOVA	0.842213	0.734226	0.731436	0.728507
LASSO	0.670082	0.604308	0.658801	0.616078
Wrapper (RFE-RF)	0.844262	0.734727	0.736685	0.72936
Wrapper (RFE-LDA)	0.829918	0.722196	0.71957	0.71523

It can be seen that the best performing methods would be Wrapper (RFE-RF) and ANOVA methods, showing the highest performance in terms of Accuracy, Precision, Recall, and F1-Score. These methods seem to be the most effective for this dataset.

Moreover, SVD and PCA perform better than the nonlinear methods like Autoencoder, Kernel PCA, and t-SNE. t-SNE, despite being a powerful tool for visualization, does not perform well in this classification task, suggesting that it may not be ideal for reducing dimensions in this specific context. Also, NMF (MinMaxScaled) and t-SNE significantly reduce classification performance, indicating that not all dimension reduction techniques are beneficial for this dataset. In contrast, methods like PCA and SVD maintain decent performance, highlighting that linear methods might preserve the data structure more effectively for this classification task.

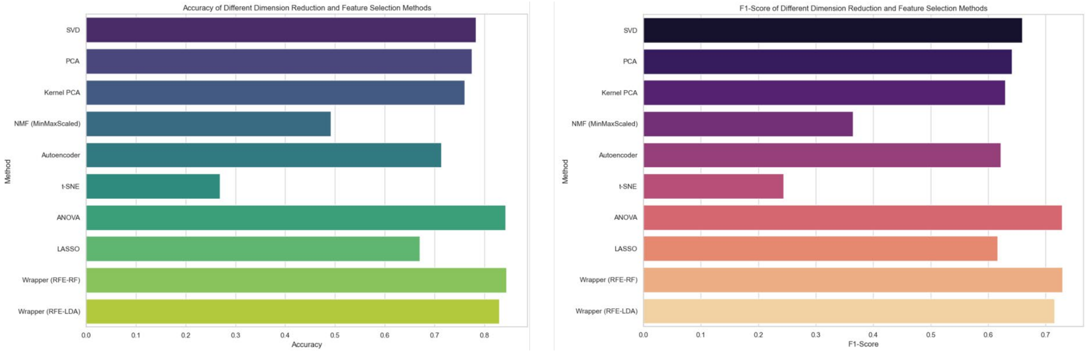

Figure 2 Dimension reduction performance

Linear Methods (like SVD and PCA) generally retain better classification performance compared to nonlinear methods. Filtering/Feature Selection Methods (like ANOVA and Wrapper methods) significantly enhance performance, likely by removing irrelevant or noisy features. Nonlinear Methods may sometimes over-complicate the data structure, leading to poorer performance in some cases, as seen with t-SNE.

3.2 Section Summary

The choice of dimension reduction technique can drastically impact classification performance. Linear methods and appropriate feature selection seem to be more effective for this particular dataset. Nonlinear methods should be applied with caution and may require more tuning or might be better suited for other types of data or tasks.

4. Question 1d

4.1 Method comparisons and possible break-down points - how much mislabeling can the methods handle?

Set contamination level 0, 5%, 10%, 20%, 30% to the data and apply ANOVA, LASSO and Wrapper-LDA to six different classifiers (logistic regression, SVM, KNN, Decision Tree, Random Forest, MLP Classifier). Then export each F1-score, recall and precision as the results, which are shown in Figure .

The classifiers show varying degrees of robustness to label contamination, with Random Forest and MLP Classifier performing slightly better at lower contamination levels. However, none of the classifiers are immune to the effects of mislabeling, and all show significant performance degradation as the level of contamination increases.

Observations with low confidence are more likely to be associated with mislabeling, especially at higher contamination levels. Identifying these observations could potentially allow for corrective measures, such as re-labeling or using more robust models designed to handle noisy labels. For very high contamination, classification may become unreliable, and alternative strategies, such as robust learning methods or semi-supervised learning, could be considered.

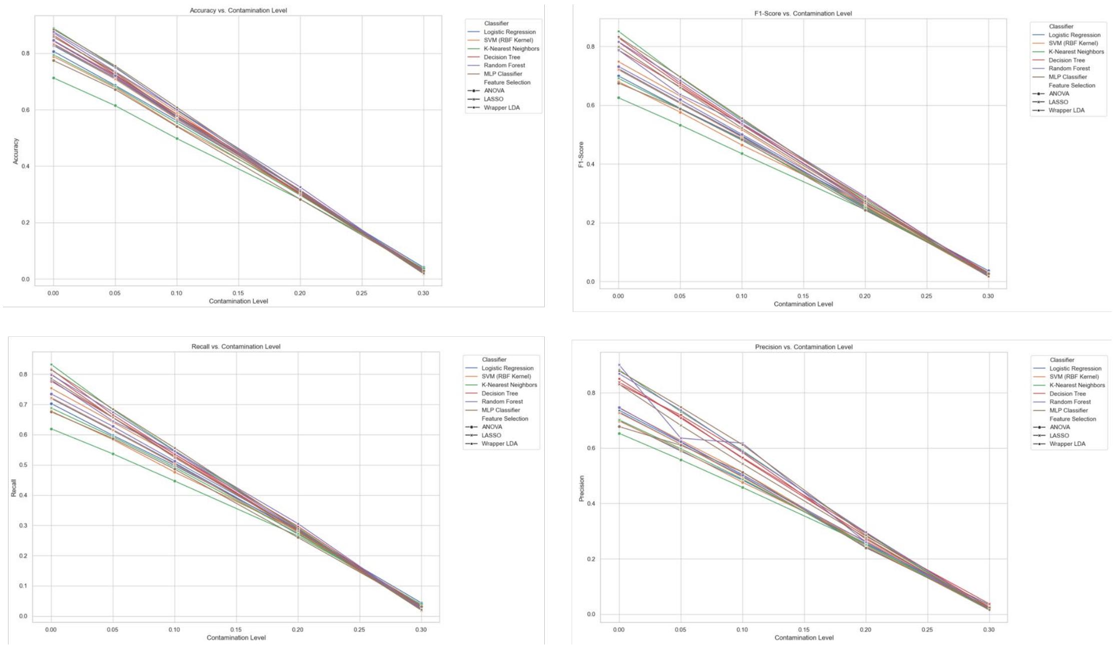

Figure 3 Parameters (Accuracy, F1-score, Recall and Precision) vs contamination level

Confidence is assessed using the Confidence metric provided. This metric helps in identifying how confidently a classifier can assign an observation to a particular class. At low contamination (0% to 5%), confidence levels are higher, with many observations being classified with high confidence. The classifiers maintain their ability to distinguish between classes with reasonable certainty.

Mislabeled observations can be associated with lower confidence scores. As the contamination level increases, the confidence in predictions decreases, suggesting that the model is struggling to differentiate between classes.

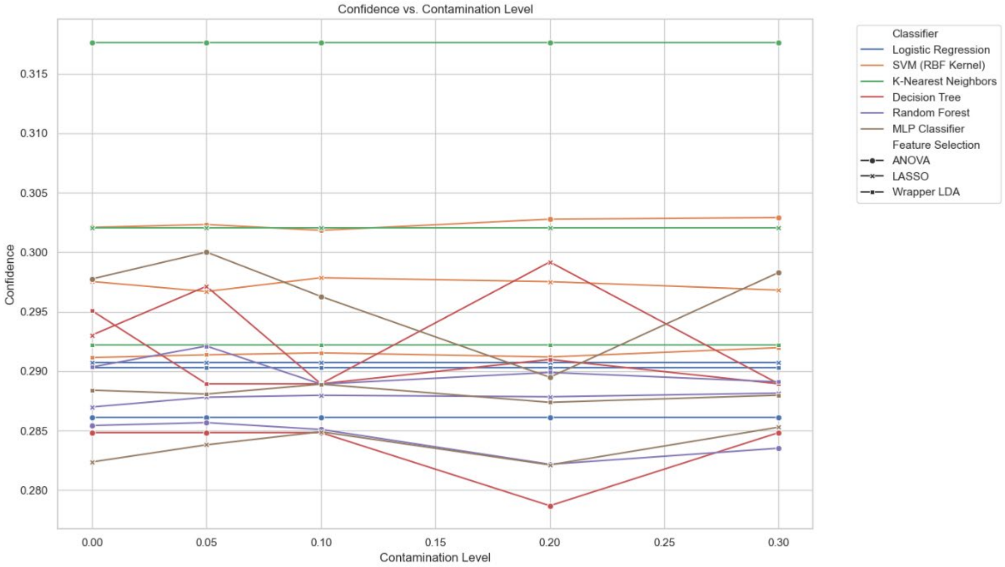

Figure 4 Classify confidence level shows relatively similar with contamination level changing

At Lower Contamination 0%-5%, mislabeled observations are likely to be those with lower confidence scores, even if overall performance is still reasonable. At Higher Contamination 20%-30%, the number of observations with low confidence scores increases significantly. In this case, most of the observations are likely mislabeled, making it challenging for the classifiers to learn effectively from the data.

4.2 Impact on classification performance at the class level

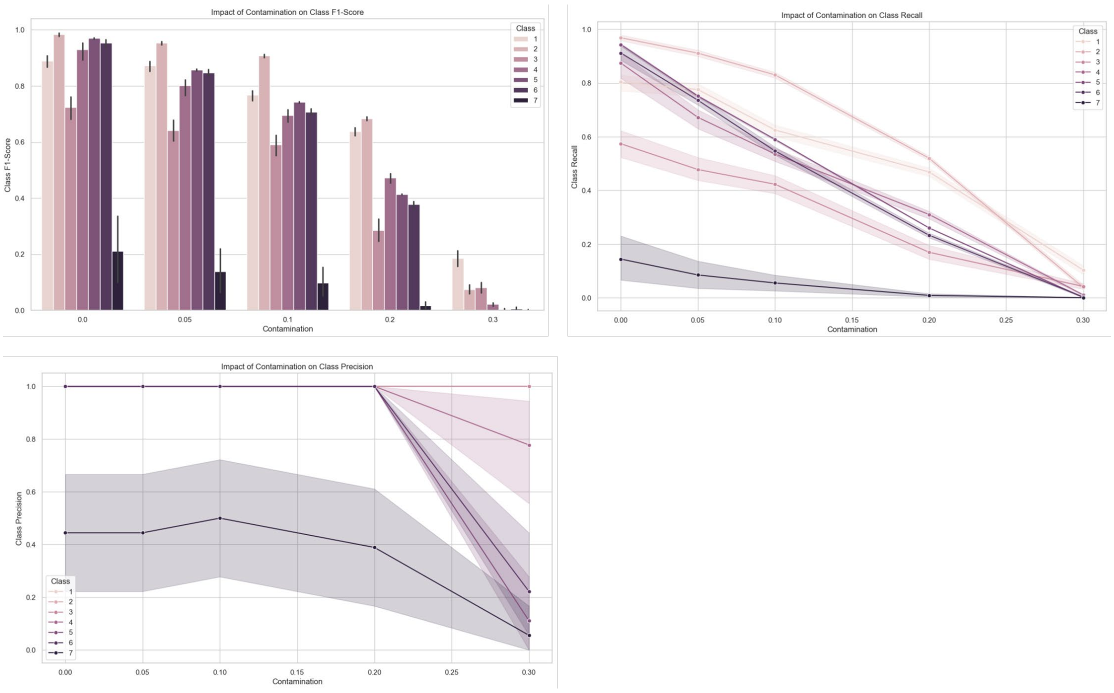

Figure 5 Impact of contamination on class for F1-score, Recall and Precision

Typically, as the contamination levels increasing, the classify performance for the class level shows some significant decrease. For class “2”, “5” and “6”, their F1-score doesn’t vary a lot from contamination from 0% to 20%. Besides, class “7” decent a lot. However, all classes aren’t classified well at contamination 30%, showing a very poor score. The other values like recall and precision both show a similar result. Especially, class “7” is classified more worse than the others because of the imbalance.

The robustness of classification methods to label contamination varies significantly across different classes. For instance, Class 1 and Class 2 are more resilient, maintaining higher performance metrics, whereas Class 7 is particularly susceptible to degradation in both recall and precision. There are some possible reasons for the results shows. If some classes have fewer samples, they may be more sensitive to label noise, which might explain the sharp decline in performance for certain classes. Some classes might be inherently more distinguishable based on the available features, leading to better performance even in the presence of label contamination.

4.3 Section Summary

Label contamination adversely impacts classification performance, with a more pronounced effect on certain classes. Ensuring robustness to label noise is crucial, especially for classes that are more vulnerable. This analysis underscores the importance of careful data cleaning, feature selection, and classifier choice to mitigate the effects of mislabeling, thereby improving overall model robustness.