Data Mining - 2024 Fall at The Department of Computer Science

Assignment 2

(Github Code:https://github.com/Ashishlathkar77/Data-Mining---Assignment-2)

Ashish Balkishan Lathkar (AL23S)
Master of Science Data Science
Florida State University

Introduction

I tackled three different machine learning problems, each focusing on distinct tasks: classification using Support Vector Machines (SVMs), clustering using K-means, and classification with techniques to address class imbalance. The main goals of this report are to analyze the accuracy of different classification models, assess the quality of K-means clustering using various values of `k`, and evaluate how class imbalance handling methods impact classification performance.

Problem 1: Support Vector Machine (SVM) Classification with Polynomial and Gaussian Kernels

In this problem, the task was to classify a dataset using Support Vector Machines (SVMs) with two different types of kernels: Polynomial and Gaussian (RBF). SVMs are powerful classifiers that attempt to find a hyperplane that maximizes the margin between different classes in the dataset. The Polynomial kernel maps data into a higher-dimensional space, while the Gaussian kernel, or Radial Basis Function (RBF), is a more flexible kernel that can handle non-linear relationships.

Data Preprocessing:

To begin with, the dataset was scaled using the `StandardScaler` to normalize the features and ensure that the model's performance wouldn't be biased by varying feature scales. This step is crucial in SVMs since they are sensitive to the scale of the data.

Models:

1. **Polynomial Kernel (degree=2):**
   - An SVM classifier was trained using a polynomial kernel with degree 2. This kernel is suitable for problems where the decision boundary between classes is a polynomial function of the input features.
2. **Gaussian Kernel (RBF, γ=2):**
   - The second model used the Gaussian kernel (RBF) with a fixed gamma value of 2. The Gaussian kernel is a non-linear kernel that maps data into an infinite-dimensional space, making it suitable for more complex datasets.

Results:

- **Accuracy with Polynomial Kernel (degree=2):** 40.98%
- **Accuracy with Gaussian Kernel (γ=2):** 0.55%

The Polynomial kernel achieved an accuracy of 40.98%, which is reasonable but not outstanding, indicating that the dataset may not perfectly align with a polynomial decision boundary. The Gaussian kernel, however, performed poorly, with an accuracy of only 0.55%, suggesting that the data may not be well suited for a Gaussian kernel with the specified gamma value.

Problem 2: K-Means Clustering

The second problem involved K-Means clustering, a type of unsupervised learning that aims to partition a dataset into `k` clusters. The goal was to identify the optimal number of clusters for a given dataset and evaluate the performance of K-means using different values of `k` (3, 5, and 7).

K-Means Algorithm:

K-Means clustering is based on iteratively assigning data points to the nearest cluster center and then updating the cluster centers based on the mean of the points assigned to each cluster. This process continues until convergence.

Results:

- **Sum of Squared Errors (SSE) for k=3:** 587.3186
- **Sum of Squared Errors (SSE) for k=5:** 385.6879
- **Sum of Squared Errors (SSE) for k=7:** 280.7045

As the number of clusters `k` increased, the SSE decreased. This indicates that with a higher `k`, the clusters became more compact and closer to the actual data distribution. This pattern suggests that the data might be better represented with more clusters, though there is a trade-off between model complexity and overfitting.

Problem 3: Classification with Random Forest and Logistic Regression Addressing Class Imbalance

The third problem dealt with class imbalance, a common challenge in many real-world datasets where one class significantly outnumbers the other. Imbalanced datasets can lead to biased models that favor the majority class, resulting in poor generalization. To address this issue, I employed three different strategies:

1. **SMOTE (Synthetic Minority Over-sampling Technique):** SMOTE is a technique that generates synthetic samples by interpolating between existing minority class instances. This technique helps balance the class distribution without duplicating minority class samples.
2. **Random Undersampling:** This technique reduces the size of the majority class by randomly removing samples, thereby achieving a balanced class distribution.
3. **Class Weights:** In this approach, the algorithm assigns different weights to the classes based on their frequencies, allowing the classifier to pay more attention to the minority class.

Models:

- **Random Forest:** Random Forest is an ensemble learning method that combines multiple decision trees to make predictions. It was used with the three class imbalance handling techniques.

- **Logistic Regression:** A linear model for binary classification, which was also trained using the same class imbalance techniques.

Results:

- **Random Forest with SMOTE (Oversampling) F1-score:** 0.761
- **Random Forest with Random Undersampling F1-score:** 0.713
- **Random Forest with Class Weights F1-score:** 0.714
- **Logistic Regression with SMOTE F1-score:** 0.763

From the results, we observe that both the Random Forest and Logistic Regression models benefited from the oversampling technique (SMOTE), achieving the highest F1-scores of 0.761 and 0.763, respectively. The Random Undersampling and Class Weight methods also improved the F1-score but were slightly less effective, with F1-scores of 0.713 and 0.714.

This demonstrates that oversampling the minority class with SMOTE is the most effective method for addressing class imbalance, as it increases the model's ability to correctly classify minority class instances without losing too much information from the majority class.

Conclusion

Key Findings:

1. **SVM Classification:** The Polynomial kernel yielded a reasonable accuracy (40.98%), but the Gaussian kernel performed poorly, suggesting it was not suitable for this dataset with the chosen parameters.
2. **K-Means Clustering:** The K-Means algorithm showed that increasing the number of clusters reduced the Sum of Squared Errors (SSE), indicating that more clusters might better represent the data distribution. However, choosing the optimal number of clusters requires balancing model complexity and interpretability.
3. **Class Imbalance in Classification:** Techniques for handling class imbalance, particularly SMOTE (oversampling), were effective in improving classification performance. Both Random Forest and Logistic Regression models performed best with SMOTE, achieving F1-scores above 0.75, demonstrating the importance of addressing class imbalance in real-world datasets.

Conclusions:

The results highlight the importance of selecting the right algorithm and technique for the problem at hand. For SVM classification, a deeper exploration of kernel parameters may be required for better performance. In clustering, a careful selection of `k` is essential to avoid overfitting while maintaining interpretability. Finally, when dealing with class imbalance, oversampling techniques like SMOTE proved to be the most effective for improving classification accuracy and F1 scores.

These findings emphasize the need for careful model selection and the application of advanced techniques like SMOTE when dealing with imbalanced datasets, which are common in many machine learning tasks, especially in domains like fraud detection, medical diagnosis, and financial risk assessment.