National Yang Ming Chiao Tung University (國立陽明交通大學)

Data Mining HW3

Quentin Ducoulombier (昆丁)
Student Id: 312551811

Data Mining

顏安孜

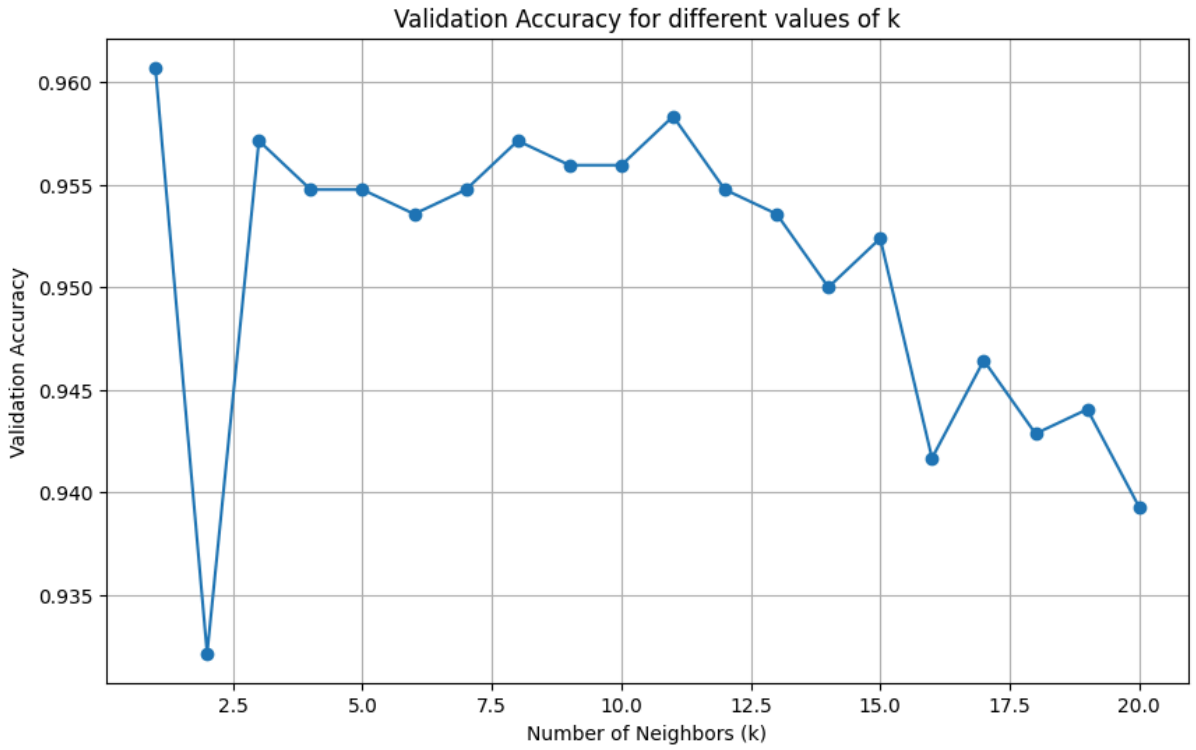
2024-06-20

1. Explain your implementation which get the best performance in detail.

Best Performing model: K-Nearest Neighbors (KNN)

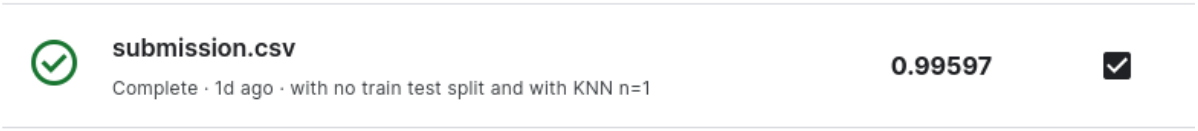
The K-Nearest Neighbors (KNN) algorithm is a straightforward and effective method for classification and anomaly detection. Here is the detailed explanation of the KNN implementation which achieved the best performance:

1. **Data Preprocessing:**
    - The training data was loaded and separated into features and target variables.
    - The features were standardized using StandardScaler to ensure that each feature contributes equally to the distance calculations.
2. **Hyperparameter Tuning:**
    - To determine the optimal number of neighbors (k), I conducted a hyperparameter search by training the KNN model with various values of k (from 1 to 20).
    - Each model was evaluated using a validation set created by splitting the training data with a train_test_split of 0.2.
    - Validation accuracies were recorded for each k value, and the best k was chosen based on the highest validation accuracy.



3. **Model Training:**
    - After identifying the best k (k=1 in this case), the KNN model was trained on the entire training dataset without using train_test_split.
    - This approach leverages the entire dataset for training, potentially improving the model's performance.
4. **Anomaly Detection:**
    - The model was then used to predict on the test data, calculating the mean distance to the k nearest neighbors.
    - These distances were used as the anomaly scores, which were saved into a submission file for evaluation.

The KNN model achieved the best performance with an AUC score of 0.99597 on Kaggle, indicating its effectiveness in detecting anomalies when trained on the entire dataset with k=1.

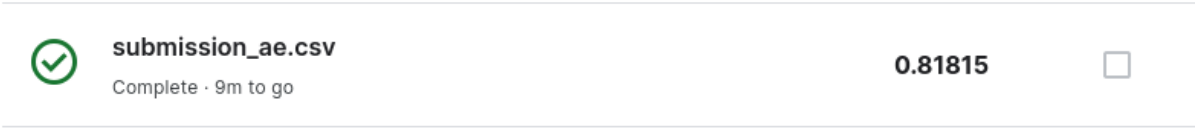


Autoencoder

The Autoencoder is a neural network designed for unsupervised learning, typically used for anomaly detection by reconstructing the input data and measuring the reconstruction error. Here's an overview of the Autoencoder implementation:

1. **Model Architecture:**
    - The Autoencoder consists of an encoder that compresses the input data into a lower-dimensional representation and a decoder that reconstructs the original data from this representation.
2. **Training:**
    - The model was trained to minimize the mean squared error (MSE) between the input and the reconstructed output using the training dataset.
3. **Anomaly Detection:**
    - The model was used to reconstruct the validation and test data.
    - The reconstruction error (MSE) was calculated for each instance.
    - A threshold was determined based on the 95th percentile of the reconstruction errors on the validation set.
    - Instances with reconstruction errors above this threshold were classified as anomalies.

The Autoencoder achieved an AUC score of 0.81815, indicating its potential in detecting anomalies but performing slightly worse than KNN.



One-Class SVM

The One-Class SVM (Support Vector Machine) is a method for anomaly detection that identifies the boundary around the normal data points and classifies points outside this boundary as anomalies. Here's a summary of the One-Class SVM implementation:

1. **Model Training:**
    - The One-Class SVM was trained on the standardized training data using the radial basis function (RBF) kernel.
2. **Prediction**:
    - The decision_function of the SVM was used to obtain continuous scores for the validation and test data.
    - These scores represent the distance from the decision boundary, with higher absolute values indicating a higher likelihood of being an anomaly.
3. **Thresholding:**
    - A threshold was applied to the decision function scores to classify instances as normal or anomalies.
    - The threshold was chosen based on the 1st percentile of the decision function scores on the validation set.

The One-Class SVM achieved an AUC score of 0.85047, showing its effectiveness but still performing slightly worse than KNN in this specific task.

In summary, the KNN implementation with k=1 and using the entire dataset for training without train_test_split achieved the highest AUC score of 0.99597. The hyperparameter tuning process, involving the evaluation of different k values on a validation set, was crucial in identifying the optimal k. The Autoencoder and One-Class SVM, while effective, achieved lower AUC scores of 0.81815 and 0.85047, respectively. Each method has its strengths and may be more suitable for different types of anomaly detection tasks.

2. Explain the rationale for using auc score instead of F1 score for binary classification in this homework.

AUC Score

1. **Threshold-Independent:**
    - The AUC score evaluates the model's performance across all possible classification thresholds, providing a comprehensive measure of its ability to distinguish between anomalies and normal instances.
2. **Overall Model Performance:**
    - AUC reflects the trade-off between true positive rate and false positive rate, offering a holistic view of the model's ranking capability.
3. **Insensitive to Class Imbalance:**
    - AUC is robust in the presence of class imbalance, a common scenario in anomaly detection tasks, ensuring reliable performance evaluation.

F1 Score

1. **Threshold-Dependent:**
    - The F1 score depends on a specific threshold, which might not be optimal or consistent, leading to less reliable performance evaluation.
2. **Single Threshold Focus:**
    - F1 provides a snapshot of model performance at one threshold, missing the broader perspective of how the model performs across all thresholds.
3. **Sensitive to Class Imbalance:**
    - F1 can be misleading in imbalanced datasets, often failing to provide an accurate picture of model performance.

Using the AUC score ensures a more reliable comparison of models, especially in imbalanced datasets, and captures the model's overall performance better than the F1 score. While the F1 score can vary significantly based on the chosen threshold, the AUC score remains consistent, providing a stable and reliable evaluation metric

3. Discuss the difference between semi-supervised learning and unsupervised learning.

Semi-Supervised Learning

1. **Definition:**
    - Semi-supervised learning uses a combination of a small amount of labeled data and a large amount of unlabeled data for training.
2. **Labeled and Unlabeled Data:**
    - This approach leverages the labeled data to guide the learning process, enhancing the model's ability to generalize from the larger set of unlabeled data.
3. **Applications:**
    - It is particularly useful in scenarios where labeled data is scarce or expensive to obtain, such as image recognition or medical diagnosis.
4. **Techniques:**
    - Techniques include self-training, co-training, and generative models that utilize both labeled and unlabeled data to improve learning.
5. **Advantages:**
    - Reduces the need for large labeled datasets, which are often costly to produce, and improves model performance by utilizing vast amounts of unlabeled data.

Unsupervised Learning

1. **Definition:**
    - Unsupervised learning uses only unlabeled data to find hidden patterns or intrinsic structures within the data.
2. **No Labeled Data:**
    - It works without any labeled responses, aiming to uncover the underlying structure from the data itself.
3. **Applications:**
    - Commonly used for clustering, dimensionality reduction, and anomaly detection, such as customer segmentation and exploratory data analysis.
4. **Techniques:**
    - Techniques include clustering algorithms like K-means, hierarchical clustering, and dimensionality reduction methods like PCA and t-SNE.
5. **Advantages:**
    - Useful for discovering previously unknown insights and natural groupings in data, aiding in exploratory data analysis.

Relevance to My Work

In this homework, the use of semi-supervised learning is unnecessary because the results obtained from the supervised methods (KNN, Autoencoder, and One-Class SVM) are already highly satisfactory. The KNN model, in particular, achieved an AUC score of 0.99597, demonstrating its really good performance. Given these results, the complexity and additional steps involved in implementing semi-supervised learning are not warranted for this task. The existing models provide a robust and accurate solution for the anomaly detection problem at hand.