Class Revision Notes on Handling Imbalanced Data in Machine Learning

Welcome to the revision notes on handling imbalanced data in machine learning. This session covered techniques like SMOTE, concepts like precision-recall curves, thresholds, and strategies for both handling imbalanced datasets and measuring model performance. Let's delve into each aspect in detail.

1. Understanding Imbalanced Data

Definition: Imbalanced data occurs when the number of instances of one class significantly outnumbers the instances of another class. This typically presents challenges in training models as they may become biased towards the majority class【4:6†source】.

Degrees of Imbalance:

- **Slight Imbalance:** 70-30% class ratio.
- **Imbalanced:** 80-20% or 90-10% class ratio.
- **Extremely Imbalanced:** 95-5% class ratio【4:6†source】.

2. Challenges with Imbalanced Data

Imbalanced datasets can lead to models that poorly generalize to the minority class, which is often of higher interest. This occurs because loss functions generally reward correctly predicting the majority class more, leading to biases in model training 【4:8†source】【4:17†source】.

3. Techniques to Address Imbalance

Synthetic Minority Oversampling Technique (SMOTE)

neighbors, then creates synthetic points along the line segments joining them【4:0†source】.

Implementation:

- Use `from imblearn.over_sampling import SMOTE`.
- Apply `smote.fit_resample(X_train, Y_train)` to generate balanced datasets for training【4:0†source】.

Oversampling and Undersampling

- **Oversampling**: Increase the number of minority class samples to match the majority class by replicating existing data or using techniques like SMOTE【4:1†source】.
- **Undersampling**: Reduce the majority class instances to balance the dataset, but may lead to loss of information【4:6†source】.

Class Weighting

In cases where sampling is not preferred or possible, weights can be assigned to the classes. This involves giving higher importance to the minority class to counteract the imbalance:

- Define class weights in models (e.g., `class_weight='balanced'` in Scikit-learn)【4:17†source】.

4. Evaluation in the Presence of Imbalance

Precision, Recall, and F1-Score

- **Precision**: Measures the accuracy of the positive predictions (true positives / true positives + false positives).
- **Recall (Sensitivity)**: Measures the ability of a model to find all relevant instances (true positives / true positives + false negatives).
- **F1-Score**: Harmonic mean of precision and recall, useful for imbalanced data【4:6†source】.

ROC and Precision-Recall Curves

【4:19†source】.

- **Precision-Recall Curve**: Plots precision against recall for different threshold values, often more informative in imbalanced scenarios 【4:5†source】【4:13†source】.

Area Under the Curve (AUC)

- **AUC-ROC and AUC-PR**:
  - AUC-ROC might be misleading on imbalanced datasets; thus, the precision-recall AUC is more appropriate 【4:5†source】【4:15†source】.

5. Practical Considerations

- **Threshold Adjustment**: Determining optimal decision thresholds can aid in maximizing relevant metrics according to the problem needs, such as high recall in disease detection 【4:10†source】【4:16†source】.
- **Sensitivity vs. Specificity**:
  - Sensitivity focuses on capturing as many positives as possible (important in medical diagnosis to avoid missing disease cases).
  - Specificity focuses on correctly identifying negatives (important when false positives have high costs) 【4:10†source】【4:14†source】【4:18†source】.

Conclusion

Handling imbalanced data effectively involves understanding the problem space, applying techniques to balance datasets, and choosing appropriate evaluation metrics. These strategies can significantly improve model performance and better meet the goals of specific applications.