Suggesteria Programming Assignment-1 Report

Google Developers Group Project (GDG)

By-> Keshav Bansal

Colab File Link -> **Suggesteria-Programming-Assignment-1.ipynb**

Online Shoppers Purchasing Intention Dataset

Dataset Overview:

Source ->
https://archive.ics.uci.edu/dataset/468/online+shoppers+purchasing+intention+dataset

The dataset captures the purchasing behavior of online shoppers, with 10 numerical features and 8 categorical features. The target variable **Revenue** indicates whether a session resulted in a purchase (True) or not (False).

Numerical features explanation:

1. **Administrative**: Count of administrative pages visited.
2. **Administrative Duration**: Total time spent on administrative pages.
3. **Informational**: Count of informational pages visited.
4. **Informational Duration**: Total time spent on informational pages.
5. **Product Related**: Count of product-related pages visited.
6. **Product Related Duration**: Total time spent on product-related pages.
7. **Bounce Rate**: Percentage of sessions that exited the site after viewing one page.
8. **Exit Rate**: Percentage of sessions where a page was the last one viewed.
9. **Page Value**: Average value of a page, reflecting its contribution to eventual transactions.
10. **Special Day**: Proximity of the session to a special day (e.g.Mother's Day).

Categorical features explanation:

1. **Operating System**: OS used during the session.
2. **Browser**: Browser used during the session.
3. **Region**: Geographical region of the session.
4. **Traffic Type**: Type of traffic source.
5. **Visitor Type**: Whether the visitor is new or returning.
6. **Weekend**: Boolean indicating if the session occurred on a weekend.
7. **Month**: Month when the session took place.
8. **Revenue** (Target Variable): Boolean indicating if a purchase was made.

Key Observations:

- **Class Imbalance**: Only 15.5% of sessions result in purchases, posing challenges for classification.
- Range of features: Numerical features vary significantly in scale (e.g., BounceRates vs. ProductRelated Duration). **Used Standard Scaler to address this issue .**

Implemented Algorithms

1) K-Nearest Neighbors (KNN)

- A distance-based algorithm that predicts a class based on the majority class of its k nearest neighbors.
- Sensitive to feature scaling and requires normalization.

2) **Logistic Regression**

- A linear model for binary classification that estimates the probability of a class using a logistic function.
- A good algorithm for binary classification like in this case (True/False).

3) **Random Forest**

- An ensemble method that builds multiple decision trees and aggregates their outputs for prediction.
- Single decision tree is prone to overfitting , Random Forest addresses this by averaging out predictions from multiple decision trees .

4) **Gradient Boosting**

- An ensemble technique that builds trees sequentially, minimizing prediction errors at each step.
- Effective for imbalanced datasets.

5) **AdaBoost**

- A boosting algorithm that combines weak classifiers (e.g., decision stumps) to improve overall performance.

6) **Support Vector Machine (SVM)**

- Constructs a hyperplane in a high-dimensional space to separate classes.

7) **XGBoost (Extreme Gradient Boosting):**

- It is an ensemble learning technique based on gradient boosting. It builds sequential models where each new model corrects the errors of the previous one, optimizing for speed and performance using advanced regularization (L1 & L2).

8) Naive Bayes

- It is a simple and fast probabilistic model based on Bayes Theorem. It assumes conditional independence among features, making it computationally efficient.

Comparison of Classification Accuracies

(with Class Imbalance)

Algorithm	Accuracy	ROC-AUC
KNN	87%	0.78
Logistic Regression	88%	0.86
Random Forest	90%	0.89
Gradient Boosting	89%	0.90
AdaBoost	89%	0.91
SVM	88%	0.85
Naive Bayes	78%	0.80
XGBoost	88%	0.82

Addressing Class Imbalance:

I tried resolving the issue of Class Imbalance using ->

1) Oversampling the minority class (By duplicating entries) .
2) The SMOTE (Synthetic Minority Over-sampling Technique) to generate new samples for the minority class .
3) Class Weights in Logistic Regression and SVM to penalize the model more for misclassifying the minority class and Bayesian learning with imbalanced class priors .

These methods were not much effective in improving the models since the class imbalance is very vast in the original dataset , but the Duplication method of oversampling minority class was the most effective .

Insights on Algorithm Performance

1. **Best Performing Algorithm (on Class Imbalanced data):**

    - **Random Forest** with ROC-AUC (0.89) and strong accuracy of (90%) and **AdaBoost** with ROC-AUC (0.91) and strong accuracy of (89%) performed best , demonstrating their ability to handle the imbalanced dataset effectively.

    - When dealing with imbalanced datasets, tree-based algorithms like Random Forest and ensemble methods like AdaBoost are generally considered the best classification algorithms as they are naturally robust to class imbalances and can effectively capture complex relationships within the data.

Conclusion

- Ensemble methods particularly **Random Forest** and **AdaBoost** provided the best performance for predicting purchasing intentions.
- Addressing class imbalance is essential for a reliable model.