Supervised Machine Learning Classification

CA1 Report

Author

Akash Jayakumar (20020040)

Academic Year: 2024 - 2025/ January

Dublin Business School

Under the Guidance of

Dr. Assem Abdelhak

Supervised Machine Learning – Classification – CA 1

Dataset Given - CustomerChurn.csv

Each row in CustomerChurn.csv corresponds to a bank's credit card customer.
Number of Instances: 6237
Number of Attributes: 15 independent variables + 1 target variable

Target Variable - Attrition_Flag

1. Data Preparation (What steps would you take to prepare your data? Discuss your approach)

To Prepare the data, I followed a Systematic Approach:

- I Checked Excel (CSV) file given to me by applying filters and isolating the columns with binary (1,0) kind of value and uploaded the file inside my Colab.

- After Uploading I stored it inside a "dataset1" variable through read_csv and then used various commands like.
  - print(dataset1)
  - print(dataset1.head())
  - print(dataset1.tail())
  - print(dataset1.describe())
  - print(dataset1.info())
  - print(dataset1.shape)

- These gave a basic idea of how many fields have int value and how many have float value and how many have str value - float64(1), int64(11), object(4).

- The object types need to be converted to integer format for compatibility with machine learning programming.

- Then I split the dataset into target features and rest of features and stored them in their respective variables. This draws a clear distinction between input features and output features.

- Then used train_test_split to split dataset into training and test series, then applied SMOTE to address imbalance in the dataset.

- After Doing Balancing to my Training dataset stored inside my x_train and y_train. I went forward in doing other further process.

2. Model Hyperparameter Tuning (Which hyperparameters would you tune and why? How would you tune them?)

For Hyper Parameter Tuning I focused on These models:

- I used Pipeline method and GridSearch Method first instead of Random Forest method 1as the result would have better balanced results.
- I used cv = 5 and precision as scoring factor here since I need to reduce false positive as much as I can, because recall is tolerable, precision is not.
- The 'classification_n_estimators' I got is 350, precision – 93%
- Random Forest classifier – Method 1 with n_estimators value as 350 and mapped out important features.
  - Total_Trans_Ct – 0.228018
  - Total_Trans_Amt – 0.177853
  - Total_Revolving_Bal – 0.131902
- Then process continued by narrowing down n_estimators – 300 to 270 and then reached to 285 with precision value - 0.9344217896764787 or 93%.
- After the reaching the conclusion on Random Forest , I moved on to Support Vector Classifier.
- Employed similar GridSearch CV approach with precision as scoring metric and cv=5.
- The Best Parameters are
  - classification__C - 0.001
  - classification__kernel – sigmoid
  - Precision value - 0.9723519613005995 or 97%

3. Choice of Evaluation Metric (Which metric would be suitable for model evaluation and why?)

- Precision was chosen as the evaluation metric for model evaluation.
- Precision is suitable for this scenario because I consider false positives as non-tolerable and to focus on minimizing false positive predictions. I went with prediction.
- In this specific dataset, if I predict the customer will churn and doesn't churn, it will not have high impact on bank. But still I want to consider a scenario where it does matter like in case of Silicon Valley Bank, Happened last year.

4. Overfitting avoidance mechanism (Which mechanism (feature Selection/ regularization) would you use and why?)

    - To avoid overfitting, I used regularization techniques. For feature selection, I used Random Forest to identify important features and find their precision and accuracy metrics.

5. Results analysis

    a. Which of the two models (random forest or support vector classifier) would you recommend for deployment in the real-world?

    - Based on the results and evaluation metrics, I would recommend the Support Vector Classifier for deployment in the real world. Because It has higher precision compared to the Random Forest around 97%.

    b. Is any model underfitting? If yes, what could be the possible reasons?

    - None of the models have an indication of underfitting. Both models high precision scores.