Lovedeep Bajaj

What Classification Technique to Be Used Inside of The Pipeline to Figure Out If A Person Will Churn

**Abstract:**

A pipeline allows you to have a true workflow and that was what I wanted to achieve with this project. The stage of the pipeline that I was mostly focused on was the estimator stage. There are many different classification techniques that I can use, and I wanted to see which one yielded me the best results. Once I had figured that out, I tuned that machine learning technique to make the pipeline perform even better. The dataset that was used in this project is telling you whether an individual is/was a customer of the bank. To help classify the individuals there are 22 features that are given with the dataset. Some of the features are the customers age, gender, dependent count, education level, marital status, income category, card category, and many more. The results that I obtained after my research was that for this dataset using a bagging classifier was the best estimator for the pipeline.

**Related Work:**

I obtained this dataset from the Kaggle website. If you go onto the website and search for the user Sakshi Goyal, you will find the dataset and see that there has been analysis done on this dataset before. The analysis done on this dataset was by people using logistic regression, random forest, and other classification techniques. Also, of the analysis that is available on the Kaggle website for this dataset is done in python but, people have done analysis on this dataset using R. For example, there is an article titled “Machine learning with R: Churn Prediction” and it uses this dataset and random forest to classify the customers. One thing that I saw none of these researchers use was a pipeline in their analysis. That is what differentiates the research that I did on this dataset from the research that has already been done before. Since this is an imbalanced dataset and there aren’t as many customers who churned, I was more focused on the pipelines f-score and sensitivity rather than the accuracy of the pipeline.

**Data:**

The dataset originally contained 23 but, I removed 3 of those features. One of those features was an id and the other 2 involved naive bayes classifier information. After removing those features, I checked to see if there were any missing values or duplicates and there was none.

When it came to outlier analysis, I analyzed each feature individually and made sure that the values inside made sense based on the feature. For the age attribute I wanted to make sure that the minimum age was greater than or equal to 18. Since the minimum age was 26 which was greater than 18 and the maximum age was 73 which is less than 100, I concluded that the age feature didn’t contain any outliers. Then came the gender attribute and for that I made sure that the only values it contained was male and female. Then I replaced males and females with the values 0 and 1. Similarly to the gender feature for dependent count I checked to see the unique values and that there was no value over 5. For the education level, marital status, and income category features I did the same type of analysis on all of them. Firstly, I checked to see their distinct values and since all of them contained unknown I replaced that with the features name concatenated with unknown. The reason I did this was because I planned on doing one hot encoding on these features and I couldn’t have multiple features named unknown. Card category was the next feature, and I checked its unique values and for all the unique values in the feature I concatenated the features name to the end of the value since I also planned on doing one hot encoding on this feature. All of the remaining features I just checked their min values and made sure they were no negatives.

After the data had been cleaned, I wanted to get a better understanding of the features I was working with thus I created visualizations for each feature. I created a function that would plot the charts for me since I created the same type of chart for every feature.

**Chart, bar chart

Description automatically generatedFigure 1.**

The image labeled Figure 1 shows you that most of the people in this dataset are still customers of the bank.

**Chart, histogram

Description automatically generatedFigure 2.**

From looking at Figure 2 you can see the customers age chart and notice it forms a bell curve.

**Figure 3.**

**Chart, bar chart

Description automatically generated**

The image labeled Figure 3 shows you there are a couple of more females in the dataset then males.

**Figure 4.**

**Chart, bar chart

Description automatically generated**

The image labeled Figure 4 shows that most of the customers have a graduate degree or a high school diploma.

**Chart

Description automatically generatedFigure 5.**

The image labeled Figure 5 shows you most of the customers of the bank are married or single.

**Chart, bar chart

Description automatically generatedFigure 6.**

The image labeled Figure 6 informs you that most of the customers of the bank have an income of less than 40 thousand dollars.

**Methodology:**

I planned on doing feature selection and I would need to edit the training and testing set each time I removed a feature. Thus, for time efficiency I made a function that returns me the training and testing set. Since the dataset is imbalanced, I used stratified k folds to create the training and testing set. This would ensure that the percentage of the attrition\_flag attribute was preserved. Then I created functions that returned me a list that contained metric measures of the pipeline’s accuracy, sensitivity, specificity, and f-score.

Once that was completed, I was able to move onto creating the stages of the pipeline. The first stage is one hot encoding of the features that contain string values. For the non-string data type features I scaled the values using standard scaler. After that stage came the estimator stage that would use a machine learning technique to classify the customers. The first technique that I tried was logistic regression with the maximum iterator set to 500. After that I tried a decision tree and random forest. The results I obtained were used as a base value for when I did feature selection.

The first feature that I removed was card\_category and the reason for that was because the majority of customers had a blue card. But I noticed that none of the pipelines improved when I did that. Then I decided to remove the customers age and gender and noticed that I was able to increase the pipeline with logistic regressions f-score to 31 percent. The last test I ran for the feature selection portion involved removing the gender, dependent\_count, education\_level, marital\_status, and income\_category. When I ran all the different pipelines, I noticed that the pipeline with the logistic regression as a classifier had improved its f-score once again and this time it was 40 percent. Based off of this I was able to conclude that I should focus my time tuning the logistic regression in order to increase the pipelines f-score even further.

When I was tuning the logistic regression, I pinpointed that focusing on figuring out the best C value to use would be the best way to improve the f-score value. I ran the pipeline over and over again using a logistic regression with different C values and was able to conclude that using a C value of 40, 50, 60, 70, 80, 90, or 100 yielded me the best results. That’s when I decided to see if using a bagging classifier with a base estimator of logistic regression would give me a better pipeline f-score. I tested different number of estimators as well as setting the C parameter in the logistic regression to be 40, 50, 60, 70, 80, 90, and 100. This was all to see if using one of these C values with the bagging classifier could allow me to obtain a better f-score. Once the bagging classifier was done running, I discovered that it was the best machine learning technique to use.

**Results:**

By setting the estimator stage of the pipeline to be the bagging classifier with a base estimator of logistic regression. Also, for the logistic regression set the C and maximum iteration parameter to 100 and 500. The number of estimators to use for the bagging classifier is 2. By using the bagging classifier in the pipeline, I am able to increase the f-score to 43 percent.

**References:**

1. <https://www.kaggle.com/sakshigoyal7/credit-card-customers>
2. <https://towardsdatascience.com/machine-learning-with-r-churn-prediction-be5d2b4dba93>
3. <https://www.valuepenguin.com/credit-cards/credit-card-age-restrictions>
4. <https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.StratifiedKFold.html>
5. <https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html>
6. <https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html>
7. <https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html>
8. <https://scikit-learn.org/stable/modules/generated/sklearn.pipeline.make_pipeline.html>
9. <https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.OneHotEncoder.html>
10. <https://scikit-learn.org/stable/modules/generated/sklearn.compose.make_column_transformer.html>
11. <https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.StandardScaler.html>