Lovedeep Bajaj

Which Individual Is Going to Default on Their Next Month Payment

**Abstract:**

Is this individual going to default on their payment next month? This is the question a lot of credit card companies have and that was the question I tried to solve using this dataset. The number of people who will default are not going to be as high as those who will not. This was the reasoning behind why I focused on the sensitivity score of the pipeline instead of the accuracy. The way I tried to solve this question is by figuring out which classification technique should be used in the pipeline to ensure the sensitivity was as high as possible. When I was done with my analysis on the dataset, I concluded that the random forest was the best at classifying the individuals.

**Related Work(s):**

The dataset was obtained from the Kaggle website and can also be found on the UCI Machine Learning Repository. On the Kaggle website, one can see that people have analyzed the dataset using R and python. I am trying a different approach and using pyspark to determine the results it will yield. Also, almost all of the individuals that I saw analyze the dataset did not use a pipeline to obtain a true workflow. They were primarily focused on finding the best model to use on the data that was in-front of them. However, I am interested in having a pipeline in which new individuals are added to the dataset without having to redo all of the normalization /standardization. There would be less of a hassle, since the normalization/standardization would be built into the pipeline.

**Data:**

The dataset itself originally contained 25 attributes and among them was an ID feature that I removed. Next, I wanted to examine how many duplicates were present in the dataset before removing them. If dropping the duplicates led to 5% or higher of data, then I would not drop them. The dataset only contained 35 rows of duplicates, which is less than 1% of the data, allowing me to drop those values. After removing all of the duplicates, I discovered there was no missing data in any of the attributes. Once all of this was done, I was able to go through each of the features one by one and determine if the values they contained made sense.

The first feature that I reviewed was the age feature to ensure that the minimum age in the dataset is 18 or older and that the maximum age is less than or equal to 100. When I analyzed the minimum age, it was 21, which is larger than 18. The maximum age is 79, which is less than 100. Thus, I concluded that there were no outliers in the age feature. The next feature within the dataset was the limit balance. For this feature, I verified that no one had a limit of 0 or a negative value. When I obtained the minimum limit balance, I examined it to be $10,000 dollars, which is not negative or 0. After determining that the limit balance feature contained appropriate values, I moved onto the sex feature. For this attribute, I made sure that the only values it contained were male (1) and female (2). I replaced 1 with 0 and replaced 2 with 1 to represent males and females, respectively. After replacing the values, I focused on the education feature. For this attribute, I checked to see that it only contained the values 1, 2, 3, 4, 5, and 6. When I examined the distinct values in this feature, I found that there were some individuals that had 0 for their education. Before removing all of these individuals, I wanted to determine how many I would be removing. If more than 1% is removed, it could be a documentation error, in which they forgot to include 0 as a value for education. There were 14 people who had 0 filled in for this feature and I removed them all, since it was less than 1% of the data within this dataset. Similarly, the same issue arose for the marriage feature. There were some individuals that had 0 for this feature, with less than 1%, so I removed them, as well. The last couple of features I looked over were PAY\_1 – PAY\_6 to make sure they only contained the values -2 till 9. The remaining features were not reviewed because they involved payments that could be either positive or negative.

After removing the potential outliers that I deemed to be actual outliers, create visualizations were created for the features in order to better understand the distribution of values. Below are some images of the visualizations that I created.

**Figure 1.Chart, bar chart

Description automatically generated**

Figure 1. Represents the limit balance range of the individuals in the dataset. One can determine that most of the individuals fall in-between $100,000 till $300,000.

**Figure 2.**

**Chart, bar chart

Description automatically generated**

Figure 2. Represents the number of males and females in the dataset, including the ones who defaulted. The dataset contains more females than males.

**Figure 3.**

**Chart, bar chart

Description automatically generated**

Figure 3. Represents the education levels of the people in the dataset. Most of the people who defaulted in this dataset have an university level education.

**Figure 4.**

**Chart, bar chart

Description automatically generated**

Figure 4. Represents marriage status. Most of the people in this dataset are single, and the number of married people who defaulted is almost the same as those who are single.

**Figure 5.**

**Chart, bar chart

Description automatically generated**

Figure 5. Represents the age range of the individuals. Most of the people in this dataset are between 21 and 40. The age range that is most likely to default are those between 21 and 30.

**Figure 6.**

**Chart, bar chart

Description automatically generated**

Figure 6. Represents the previous payments made during September and August of 2005.

**Methodology:**

I stored 70% of the dataset for the training and 30% for the testing. Nex, done, I began creating the pipeline stages. The first stage was established by assembling all the features that would help me determine if the individual would default on their next month payment inside of a vector assembler. The feature that was created from this vector assembler was called unscaled features. Afterwards, the pipeline moves onto the second stage, which involves scaling the data by utilizing a standard scaler to make sure the values are all within a similar range. The last stage, which is also the stage that I experimented with the most, was the estimator stage. For this I stage, I was trying to choose between using logistic regression or random forest to classify the individuals. In order to determine the one that yielded the best pipeline, I tested both of them.

The logistic regression was tested first. Adding the logistic regression to the pipeline, training the pipeline, and testing it out on the testing set led to an accuracy and sensitivity of 81% and 24%, respectively. After discovering the base values for the pipeline, the base logistic regression was used without conducting any hyperparameter tuning. This was followed by some hyperparameter tuning to examine if changing the C parameter would allow for a better result. However, the accuracy and sensitivity did not change, possibly due to a need for improvement by obtaining more data. The second test was with the random forest for the estimator stage. By doing so, I discovered that the pipeline performed the same as the logistic regression. The accuracy and sensitivity were 81 % and 24%, respectively. Later, I tried to see which number of trees would yield the best results. Utilizing 7 trees would improve the accuracy to 82% and the sensitivity to 33.9%. But, if 13 trees were used, there would be a sensitivity of 34.4% and an accuracy of 81%.

**Results:**

I decided that the best classification technique to use during the estimator stage is the random forest with 13 trees. I chose 13 trees instead of 7 trees due to my main focus on a higher sensitivity value than a higher accuracy value. This is because the dataset is imbalanced, and I am more curious about those who will default.

**References:**

1. <https://www.kaggle.com/uciml/default-of-credit-card-clients-dataset>
2. <https://archive.ics.uci.edu/ml/datasets/default+of+credit+card+clients>
3. <https://spark.apache.org/docs/latest/ml-classification-regression.html>
4. <https://spark.apache.org/docs/latest/ml-pipeline.html>
5. <https://spark.apache.org/docs/latest/ml-tuning.html>
6. <https://spark.apache.org/docs/latest/ml-features>
7. <http://spark.apache.org/docs/latest/api/python/pyspark.mllib.html?highlight=standardscaler>
8. <https://matplotlib.org/>
9. <https://seaborn.pydata.org/generated/seaborn.countplot.html>