**Credit Risk Analysis Report**

**Overview of the Analysis**

Credit risk poses a classification problem that’s inherently imbalanced. This is because healthy loans easily outnumber risky loans. We will use various techniques to train and evaluate models with imbalanced classes and use a dataset of historical lending activity from a peer-to-peer lending services company to build a model that can identify the creditworthiness of borrowers.

This report consists of the following analysis:

* Split the Data into Training and Testing Sets
* Create a Logistic Regression Model with the Original Data
* Predict a Logistic Regression Model with Resampled Training Data

First, we take lending data and create labels set into two classes - 0 (healthy loan) and 1 (high-risk loan). You will notice credit risk is imbalanced, with 0 (healthy loan) easily outnumbering 1 (high-risk loan).

75036 of 0 (healthy loan) vs 2500 of 1 (high-risk loan).

A picture containing text

Description automatically generated

So, we split the data into training and testing sets. We create a Logistic Regression Model with original data, then we predict a Logistic Regression Model with resampled training data.

Logo

Description automatically generated with medium confidence

After we achieved balanced data, we can now perform analysis. We were able to get an improved performance with resampled training data to predict high-risk loans.

**Steps**

## Split the Data into Training and Testing Sets

* Read the lending\_data.csv lending data into a Pandas DataFrame.
* Create the labels set (y) from the “loan\_status” column, and then create the features (X) DataFrame from the remaining columns.
* Check the balance of the labels variable (y) by using the value\_counts function.
* Split the data into training and testing datasets by using train\_test\_split.

## Create a Logistic Regression Model with the Original Data

* Fit a logistic regression model by using the original training data (X\_train and y\_train)
* Save the predictions on the testing data labels by using the testing feature data (X\_test) and the fitted model
* Evaluate the model’s performance by calculating the accuracy score of the model, generate a confusion matrix and print the classification report

## Predict a Logistic Regression Model with Resampled Training Data

* Use the RandomOverSampler module from the imbalanced-learn library to resample the data. Confirm that the labels have an equal number of data points for healthy and high-risk loans.
* Use the LogisticRegression classifier and the resampled data to fit the model and make predictions
* Evaluate the model’s performance by calculating the accuracy score of the model, generate a confusion matrix and print the classification report

**Results**

Original Data - Logistic Regression with imbalanced training data.

* + Precision for 0 (healthy loan): 1.00
  + Recall for 0 (healthy loan): 0.99 (99% accurate in predicting healthy loans as healthy)
  + Precision for 1 (high-risk loan): 0.85 (85% accurate in the high-risk loans predictions)
  + Recall for 1 (high-risk loan): 0.91 (91% accurate in predicting high-risk loans as high risk)

A picture containing table

Description automatically generated

Resampled Data - Logistic with balanced training data

Table

Description automatically generated with low confidence

* + Precision for 0 (healthy loan): 1.00
  + Recall for 0 (healthy loan): 0.99 (99% accurate in predicting healthy loans as healthy)
  + Precision for 1 (high-risk loan): 0.84 (84% accurate in the high-risk loans prediction)
  + Recall for 1 (high-risk loan): 0.99 (99% accurate in predicting high-risk loans as high risk))

**Summary**

Oversampled data performs better than the original data, even though there’s a slight drop on precision for 1 (high-risk loan) 85% vs 84% from the original data, but the recall for 1 (high-risk loan) increased 8% from 91% to 99%. The cost of associated with approving high-risk loans outweighs the loss of potential profit associated with not approving healthy loans. Therefore, I would recommend the model using resampled data.