Module 17 Challenge

New Attempt

Due May 2 by 12:59am **Points** 100 **Submitting** a text entry box or a website url

Background

Jill commends you for all your hard work. Piece by piece, you've been building up your skills in data preparation, statistical reasoning, and machine learning. You are now ready to apply machine learning to solve a real-world challenge: credit card risk.

Credit risk is an inherently unbalanced classification problem, as good loans easily outnumber risky loans. Therefore, you'll need to employ different techniques to train and evaluate models with unbalanced classes. Jill asks you to use `imbalanced-learn` and `scikit-learn` libraries to build and evaluate models using resampling.

Using the credit card credit dataset from LendingClub, a peer-to-peer lending services company, you'll oversample the data using the `RandomOverSampler` and `SMOTE` algorithms, and undersample the data using the `ClusterCentroids` algorithm. Then, you'll use a combinatorial approach of over- and undersampling using the `SMOTEENN` algorithm. Next, you'll compare two new machine learning models that reduce bias, `BalancedRandomForestClassifier` and `EasyEnsembleClassifier`, to predict credit risk. Once you're done, you'll evaluate the performance of these models and make a written recommendation on whether they should be used to predict credit risk.

What You're Creating

This new assignment consists of three technical analysis deliverables and a written report. You will submit the following:

- Deliverable 1: Use Resampling Models to Predict Credit Risk
- Deliverable 2: Use the SMOTEENN Algorithm to Predict Credit Risk
- Deliverable 3: Use Ensemble Classifiers to Predict Credit Risk
- Deliverable 4: A Written Report on the Credit Risk Analysis (README.md)

Files

Use the following link to download the **Module-17-Challenge-Resources.zip (https://2u-data-curriculum-team.s3.amazonaws.com/dataviz-online/module_17/Module-17-Challenge-Resources.zip)** file that includes the `LoanStats_2019Q1.csv` dataset and two starter code files: `credit_risk_resampling_starter_code.ipynb` and `credit_risk_ensemble_starter_code.ipynb`.

Before You Start

Create a new GitHub repository entitled "Credit_Risk_Analysis" and initialize the repository with a README.

Deliverable 1: Use Resampling Models to Predict Credit Risk (30 points)

Deliverable 1 Instructions

Using your knowledge of the `imbalanced-learn` and `scikit-learn` libraries, you'll evaluate three machine learning models by using resampling to determine which is better at predicting credit risk. First, you'll use the oversampling `RandomOverSampler` and `SMOTE` algorithms, and then you'll use the undersampling `ClusterCentroids` algorithm. Using these algorithms, you'll resample the dataset, view the count of the target classes, train a logistic regression classifier, calculate the balanced accuracy score, generate a confusion matrix, and generate a classification report.

REWIND

For this deliverable, you've already done the following in this module:

- **Lesson 17.2.3:** Split the data into training and testing sets
- **Lesson 17.3.1:** Perform logistic regression
- **Lesson 17.4.1:** Calculate accuracy, precision, and sensitivity
- **Lesson 17.4.2:** Create a confusion matrix
- **Lesson 17.10.1:** Use the `RandomOverSampler` and `SMOTE` algorithms to resample a dataset
- **Lesson 17.10.2:** Use the `ClusterCentroids` algorithm to resample a dataset

Follow the instructions below and use the `credit_risk_resampling_starter_code.ipynb` file to complete Deliverable 1.

Open the `credit_risk_resampling_starter_code.ipynb` file, rename it `credit_risk_resampling.ipynb`, and save it to your Credit_Risk_Analysis folder.

Using the information we've provided in the starter code, create your training and target variables by completing the following steps:

- Create the training variables by converting the string values into numerical ones using the `get_dummies()` method.
- Create the target variables.
- Check the balance of the target variables.

Next, begin resampling the training data. First, use the oversampling `RandomOverSampler` and `SMOTE` algorithms to resample the data, then use the undersampling `ClusterCentroids` algorithm to resample the data. For each resampling algorithm, do the following:

- Use the `LogisticRegression` classifier to make predictions and evaluate the model's performance.
- Calculate the accuracy score of the model.
- Generate a confusion matrix.
- Print out the imbalanced classification report.

Save your `credit_risk_resampling.ipynb` file to your Credit_Risk_Analysis folder.

Deliverable 1 Requirements

You will earn a perfect score for Deliverable 1 by completing all requirements below:

- For all three algorithms, the following have been completed:
  - An accuracy score for the model is calculated **(7.5 pt)**
  - A confusion matrix has been generated **(7.5 pt)**
  - An imbalanced classification report has been generated **(15 pt)**

Deliverable 2: Use the SMOTEENN algorithm to Predict Credit Risk (15 points)

Deliverable 2 Instructions

Using your knowledge of the `imbalanced-learn` and `scikit-learn` libraries, you'll use a combinatorial approach of over- and undersampling with the `SMOTEENN` algorithm to determine if the results from the combinatorial approach are better at predicting credit risk than the resampling algorithms from Deliverable 1. Using the `SMOTEENN` algorithm, you'll resample the dataset, view the count of the target classes, train a logistic regression classifier, calculate the balanced accuracy score, generate a confusion matrix, and generate a classification report.

REWIND

For this deliverable, you've already done the following in this module:

- **Lesson 17.3.1:** Perform logistic regression
- **Lesson 17.4.1:** Calculate accuracy, precision, and sensitivity
- **Lesson 17.4.2:** Create a confusion matrix
- **Lesson 17.10.3:** Use the `SMOTEENN` algorithm to resample a dataset

Follow the instructions below and use the information in the `credit_risk_resampling_starter_code.ipynb` file to complete Deliverable 2.

1. Continue using your `credit_risk_resampling.ipynb` file where you have already created your training and target variables.
2. Using the information we have provided in the starter code, resample the training data using the `SMOTEENN` algorithm.
3. After the data is resampled, use the `LogisticRegression` classifier to make predictions and evaluate the model's performance.
4. Calculate the accuracy score of the model, generate a confusion matrix, and then print out the imbalanced classification report.

Save your `credit_risk_resampling.ipynb` file to your Credit_Risk_Analysis folder.

Deliverable 2 Requirements

You will earn a perfect score for Deliverable 2 by completing all requirements below:

- The combinatorial `SMOTEENN` algorithm does the following:
    - An accuracy score for the model is calculated **(5 pt)**
    - A confusion matrix has been generated **(5 pt)**
    - An imbalanced classification report has been generated **(5 pt)**

Deliverable 3: Use Ensemble Classifiers to Predict Credit Risk (25 points)

Deliverable 3 Instructions

Using your knowledge of the `imblearn.ensemble` library, you'll train and compare two different ensemble classifiers, `BalancedRandomForestClassifier` and `EasyEnsembleClassifier`, to predict credit risk and evaluate each model. Using both algorithms, you'll resample the dataset, view the count of the target classes, train the ensemble classifier, calculate the balanced accuracy score, generate a confusion matrix, and generate a classification report.

REWIND

For this deliverable, you've already done the following in this module:

- **Lesson 17.2.3:** Split the data into training and testing sets
- **Lesson 17.3.1:** Perform logistic regression
- **Lesson 17.4.1:** Calculate accuracy, precision, and sensitivity
- **Lesson 17.4.2:** Create a confusion matrix

- **Lesson 17.9.2:** Understand adaptive boosting

Follow the instructions below and use the information in the `credit_risk_resampling_starter_code.ipynb` file to complete Deliverable 3.

1. Open the `credit_risk_ensemble_starter_code.ipynb` file, rename it `credit_risk_ensemble.ipynb`, and save it to your Credit_Risk_Analysis folder.
2. Using the information we have provided in the starter code, create your training and target variables by completing the following:
   - Create the training variables by converting the string values into numerical ones using the `get_dummies()` method.
   - Create the target variables.
   - Check the balance of the target variables.
3. Resample the training data using the `BalancedRandomForestClassifier` algorithm with 100 estimators.
   - Consult the following **Random Forest documentation** (https://imbalanced-learn.org/stable/references/generated/imblearn.ensemble.BalancedRandomForestClassifier.html) for an example.
4. After the data is resampled, calculate the accuracy score of the model, generate a confusion matrix, and then print out the imbalanced classification report.
5. Print the feature importance sorted in descending order (from most to least important feature), along with the feature score.
6. Next, resample the training data using the `EasyEnsembleClassifier` algorithm with 100 estimators.
   - Consult the following **Easy Ensemble documentation** (https://imbalanced-learn.org/stable/references/generated/imblearn.ensemble.EasyEnsembleClassifier.html) for an example.
7. After the data is resampled, calculate the accuracy score of the model, generate a confusion matrix, and then print out the imbalanced classification report.

Save your `credit_risk_ensemble.ipynb` file to your Credit_Risk_Analysis folder.

Deliverable 3 Requirements

You will earn a perfect score for Deliverable 3 by completing all requirements below:

- The `BalancedRandomForestClassifier` algorithm does the following:
  - An accuracy score for the model is calculated **(2.5 pt)**
  - A confusion matrix has been generated **(2.5 pt)**
  - An imbalanced classification report has been generated **(5 pt)**

- The features are sorted in descending order by feature importance **(5 pt)**

- The `EasyEnsembleClassifier` algorithm does the following:
    - An accuracy score of the model is calculated **(2.5 pt)**
    - A confusion matrix has been generated **(2.5 pt)**
    - An imbalanced classification report has been generated **(5 pt)**

Deliverable 4: Written Report on the Credit Risk Analysis (30 points)

Deliverable 4 Instructions

For this deliverable, you'll write a brief summary and analysis of the performance of all the machine learning models used in this Challenge.

The report should contain the following:

1. **Overview of the analysis:** Explain the purpose of this analysis.
2. **Results:** Using bulleted lists, describe the balanced accuracy scores and the precision and recall scores of all six machine learning models. Use screenshots of your outputs to support your results.
3. **Summary:** Summarize the results of the machine learning models, and include a recommendation on the model to use, if any. If you do not recommend any of the models, justify your reasoning.

Deliverable 4 Requirements

Structure, Organization, and Formatting (6 points)

The written analysis has the following structure, organization, and formatting:

- There is a title, and there are multiple sections **(2 pt)**
- Each section has a heading and subheading **(2 pt)**
- Links to images are working, and code is formatted and displayed correctly **(2 pt)**.

Analysis (24 points)

The written analysis has the following:

- Overview of the loan prediction risk analysis:
    - The purpose of this analysis is well defined **(4 pt)**

- Results:
    - There is a bulleted list that describes the balanced accuracy score and the precision and recall scores of all six machine learning models **(15 pt)**

- Summary:
    - There is a summary of the results **(2 pt)**
    - There is a recommendation on which model to use, or there is no recommendation with a justification **(3 pt)**

Submission

Once you're ready to submit, make sure to check your work against the rubric to ensure you are meeting the requirements for this Challenge one final time. It's easy to overlook items when you're in the zone!

As a reminder, the deliverables for this Challenge are as follows:

- Deliverable 1: Use Resampling Models to Predict Credit Risk
- Deliverable 2: Use the SMOTEENN algorithm to Predict Credit Risk
- Deliverable 3: Use Ensemble Classifiers to Predict Credit Risk
- Deliverable 4: A Written Report on the Credit Risk Analysis (README.md)

Upload the following to your Credit_Risk_Analysis GitHub repository:

- Your `credit_risk_resampling.ipynb` file.
- Your `credit_risk_ensemble.ipynb` file.
- An updated README.md that has your written analysis.

To submit your challenge assignment for grading in Bootcamp Spot, click Start Assignment, click the Website URL tab, then provide the URL of your Credit_Risk_Analysis GitHub repository, and then click Submit. Comments are disabled for graded submissions in BootCampSpot. If you have questions about your feedback, please notify your instructional staff or the Student Success Manager. If you would like to resubmit your work for an improved grade, you can use the **Re-Submit Assignment** button to upload new links. You may resubmit up to 3 times for a total of 4 submissions.

IMPORTANT

Once you receive feedback on your Challenge, make any suggested updates or adjustments to your work. Then, add this week's Challenge to your professional portfolio.

NOTE

You are allowed to miss up to two Challenge assignments and still earn your certificate. If you complete all Challenge assignments, your lowest two grades will be dropped. If you wish to skip this assignment, click Next, and move on to the next Module.

Module-17 Rubric

Criteria	Ratings					Pts
Deliverable 1: Use Resampling Models to Predict Loan Risk	**30 to >27.0 pts** **Demonstrating Proficiency** ✓There is an accuracy score and confusion matrix for ALL THREE algorithms. ✓A classification report is generated for ALL THREE algorithms.	**27 to >23.0 pts** **Approaching Proficiency** ✓There is an accuracy score and confusion matrix for ALL THREE algorithms. ✓A classification report is generated for TWO of THREE algorithms. ✓Code is written to generate a classification report for the third algorithm.	**23 to >19.0 pts** **Developing Proficiency** ✓There is an accuracy score and confusion matrix for ALL THREE algorithms. ✓A classification report is generated for ONE of THREE algorithms. ✓Code is written to generate a classification report for TWO algorithms, but there are errors.	**19 to >0.0 pts** **Emerging** ✓There is an accuracy score and confusion matrix for ALL THREE algorithms. ✓Code is written to generate a classification report for ONE or more algorithms.	**0 pts** **Incomplete**	30 pts
Deliverable 2: Use the SMOTEENN Algorithm to Predict Loan Risk	**15 to >13.0 pts** **Demonstrating Proficiency** ✓There is an accuracy score for the SMOTEENN algorithm. ✓There is a confusion matrix for the SMOTEENN algorithm. ✓A classification report is generated for the SMOTEENN algorithm.	**13 to >12.0 pts** **Approaching Proficiency** ✓There is an accuracy score for the SMOTEENN algorithm. ✓There is a confusion matrix for the SMOTEENN algorithm. ✓Code is written to generate a classification report for the SMOTEENN algorithm, but there is a minor error.	**12 to >9.0 pts** **Developing Proficiency** ✓There is an accuracy score for the SMOTEENN algorithm. ✓There is a confusion matrix for the SMOTEENN algorithm. ✓Code is written to generate a classification report for the SMOTEENN algorithm.	**9 to >0.0 pts** **Emerging** ✓There is an accuracy score for the SMOTEENN algorithm. ✓Code is written to generate a confusion matrix for the SMOTEENN algorithm. ✓Code is written to generate a classification report for the SMOTEENN algorithm.	**0 pts** **Incomplete**	15 pts
Deliverable 3: Use Ensemble Classifiers to Predict Loan Risk	**25 to >22.0 pts** **Demonstrating Proficiency** ✓There is an accuracy score and confusion matrix for TWO algorithms. ✓A classification report is generated for TWO algorithms. ✓The list of features is sorted in descending order by feature importance.	**22 to >18.0 pts** **Approaching Proficiency** ✓There is an accuracy score and confusion matrix for TWO algorithms. ✓A classification report is generated for TWO algorithms. ✓The list of features is not sorted in descending order by feature importance.	**18 to >16.0 pts** **Developing Proficiency** ✓There is an accuracy score and confusion matrix for TWO algorithms. ✓A classification report is generated for ONE of TWO algorithms. ✓Code is written to generate a classification report for the second algorithm. ✓Code is written that lists the features sorted in descending order	**16 to >0.0 pts** **Emerging** ✓There is an accuracy score and confusion matrix for TWO algorithms. ✓Code is written to generate a classification report for ONE of TWO algorithms. ✓Code is written that lists the features sorted in descending order by feature importance.	**0 pts** **Incomplete**	25 pts

by feature importance.

Criteria	Ratings					Pts
Deliverable 4: Structure, Organization, and Formatting	**6 to >5.0 pts Demonstrating Proficiency** The written analysis has ALL of the following: ✓There is a title, and there are multiple sections. ✓Each section has a heading and subheading. ✓There are images and references to code, and they are formatted and displayed correctly.	**5 to >4.0 pts Approaching Proficiency** The written analysis has ALL of the following: ✓There is a title, and there are multiple sections. ✓Each section has a heading and subheading. ✓There are images and references to code, and they are formatted and displayed correctly, with one or two minor errors.	**4 to >3.0 pts Developing Proficiency** The written analysis has ALL of the following: ✓There is a title, and there are multiple sections. AND ONE of the following: ✓Each section may have a heading and subheading. ✓There are images and references to code, and they are formatted and displayed correctly, with one or two minor errors.	**3 to >0.0 pts Emerging** The written analysis has ALL of the following: ✓There is a title. ✓There may be a subheading for a section. ✓There are no headings for each section, but there are three sections.	**0 pts Incomplete**	6 pts
Deliverable 4: Analysis	**24 to >20.0 pts Demonstrating Proficiency** ✓The purpose is well defined. ✓The balanced accuracy score and the precision and recall scores for ALL SIX algorithms are described. ✓The results are summarized, and there is a recommendation on which model to use or justification.	**20 to >18.0 pts Approaching Proficiency** ✓The purpose is well defined. ✓The balanced accuracy score and the precision and recall scores for FIVE of the SIX algorithms are described. ✓The results are summarized, but the recommendation on which model to use or justification is not clear.	**18 to >16.0 pts Developing Proficiency** ✓The purpose is well defined. ✓The balanced accuracy score and the precision and recall scores for FOUR of the SIX algorithms are described. ✓The results are summarized, but there is no recommendation on which model to use or justification.	**16 to >0.0 pts Emerging** ✓The purpose is well defined. ✓The balanced accuracy score and the precision and recall scores for THREE of the SIX algorithms are described. ✓The results are summarized, but there is no recommendation on which model to use or justification.	**0 pts Incomplete**	24 pts
						Total Points: 100