Project 1: Developing a Spam Detection Model

Start Assignment

- **Due** Nov 27 by 9:59pm **Points** 100 **Submitting** a text entry box, a website url, or a file upload

Background

Let's say you work at an Internet Service Provider (ISP) and you've been tasked with improving the email filtering system for its customers. You've been provided with a dataset that contains information about emails, with two possible classifications: spam and not spam. The ISP wants you to take this dataset and develop a supervised machine learning model that will accurately detect spam emails, so it can filter them out of its customers' inboxes.

What You're Creating

You will be creating two classification models to fit the provided data, and evaluate which model is more accurate at detecting spam. The models you'll create will be a logistic regression model and a random forest model.

Files

Project 1 Starter Code
Links to an external site.

Instructions

This challenge consists of the following subsections:

- Split the Data into Training and Testing Sets
- Scale the Features
- Create a Logistic Regression Model
- Create a Random Forest Model
- Evaluate the Models

Split the Data into Training and Testing Sets

Open the starter code notebook and then use it to complete the following steps.

1. Read the data from https://static.bc-edx.com/mbc/ai/m4/datasets/spam-data.csvLinks to an external site.
   into a Pandas DataFrame.
2. In the appropriate markdown cell, make a prediction as to which model you expect to do better.
3. Create the labels set (y) from the “spam” column, and then create the features (X) DataFrame from the remaining columns.

NOTE
A value of 0 in the “spam” column means that the message is legitimate. A value of 1 means that the message has been classified as spam.

4. Check the balance of the labels variable (y) by using the `value_counts` function.
5. Split the data into training and testing datasets by using `train_test_split`.

Scale the Features

1. Create an instance of `StandardScaler`.
2. Fit the Standard Scaler with the training data.
3. Scale the training and testing features DataFrames using the transform function.

** if we do not scale the data first, getting the error

```
/Users/angikar.sarkar/anaconda3/envs/dev/lib/
python3.10/site-packages/sklearn/linear_model/
_logistic.py:460: ConvergenceWarning: lbfgs failed to
converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale
the data as shown in:
    https://scikit-learn.org/stable/modules/
preprocessing.html
Please also refer to the documentation for alternative
solver options:
```

```
    https://scikit-learn.org/stable/modules/
linear_model.html#logistic-regression
  n_iter_i = _check_optimize_result(
```

Create a Logistic Regression Model

Employ your knowledge of logistic regression to complete the following steps:

1. Fit a logistic regression model by using the scaled training data (`X_train_scaled` and `y_train`). Set the `random_state` argument to `1`.
2. Save the predictions on the testing data labels by using the testing feature data (`X_test_scaled`) and the fitted model.
3. Evaluate the model's performance by calculating the accuracy score of the model.

Create a Random Forest Model

Employ your knowledge of the random forest classifier to complete the following steps:

1. Fit a random forest classifier model by using the scaled training data (`X_train_scaled` and `y_train`).
2. Save the predictions on the testing data labels by using the testing feature data (`X_test_scaled`) and the fitted model.
3. Evaluate the model's performance by calculating the accuracy score of the model.

Evaluate the Models

In the appropriate markdown cell, answer the following questions:

1. Which model performed better?
2. How does that compare to your prediction?

Requirements

To receive all points, your Jupyter notebook file must have all of the following:

Split the Data into Training and Testing Sets (30 points)

- There is a prediction about which model you expect to do better. (5 points)
- The labels set (`y`) is created from the “spam” column. (5 points)
- The features DataFrame (`X`) is created from the remaining columns. (5 points)
- The `value_counts` function is used to check the balance of the labels variable (`y`). (5 points)
- The data is correctly split into training and testing datasets by using `train_test_split`. (10 points)

Scale the Features (20 points)

- An instance of `StandardScaler` is created. (5 points)
- The Standard Scaler instance is fit with the training data. (5 points)

- The training features DataFrame is scaled using the transform function. (5 points)
- The testing features DataFrame is scaled using the transform function. (5 points)

Create a Logistic Regression Model (20 points)

- A logistic regression model is created with a `random_state` of `1`. (5 points)
- The logistic regression model is fitted to the scaled training data (`X_train_scaled` and `y_train`). (5 points)
- Predictions are made for the testing data labels by using the testing feature data (`X_test_scaled`) and the fitted model and saved to a variable. (5 points)
- The model's performance is evaluated by calculating the accuracy score of the model with the `accuracy_score` function. (5 points)

Create a Random Forest Model (20 points)

- A random forest model is created with a `random_state` of `1`. (5 points)
- The random forest model is fitted to the scaled training data (`X_train_scaled` and `y_train`). (5 points)

- Predictions are made for the testing data labels by using the testing feature data (`X_test_scaled`) and the fitted model and saved to a variable. (5 points)
- The model's performance is evaluated by calculating the accuracy score of the model with the `accuracy_score` function. (5 points)

Evaluate the Models (10 points)

The following questions are answered accurately:

- Which model performed better? (5 points)
- How does that compare to your prediction? (5 points)

Grading

This assignment will be evaluated against the requirements and assigned a grade according to the following table:

Grade	Points
A (+/-)	90
B (+/-)	80–89
C (+/-)	70–79
D (+/-)	60–69
F (+/-)	< 60

Submission

Make sure to submit your work by the assignment due date. To do so, click Submit, then upload your project files. If you have any problems uploading your files, you may also provide a link to folder within Google Drive, Dropbox, or a similar service. Set the sharing permissions so that anyone with the link can view your files.
Comments are disabled for graded submissions in Bootcamp Spot. If you have questions about your feedback, please notify your instructional staff or your Student Success Advisor.

Reference

Hopkins, M., Reeber, E., Forman, G. & Suermondt, J. 1999. *Spambase* [Dataset]. UCI Machine Learning Repository. Available: https://archive-beta.ics.uci.edu/dataset/94/spambase
Links to an external site.
 [2023, April 28]. https://doi.org/10.24432/C53G6X
Links to an external site.

.