CSC 2621- Data Science Kedziora

Lab Week #12 Leigh Goetsch

04/15/2024 04/08/2024

Lab 9: Classification

Data Description

Dataset Overview

We were provided with two CSV files for this lab: wdbc.data.csv and wdbc.names. The wdbd.names file contains the key for the wdbc.data.csv file. It also contains information on the dataset, such as the title, past usage, results, and a description of the dataset.

The wdbc.data.csv file contains 32 columns of data. The first column is the ID number, the second column is the diagnosis (M = malignant, B = benign), and the remaining colums are sets of ten real-valued features computed for each cell nucleus. The features are as follows: radius, texture, perimeter, area, smoothness, compactness, concavity, concave points, symmetry, and fractal dimension. The mean, standard error, and "worst" or largest (mean of the three largest values) of these features were computed for each image, resulting in 30 features. For instance, field 3 is Mean Radius, field 13 is Radius SE, field 23 is Worst Radius. [1]

Loading the Dataset

I loaded the wdbc.data.csv file into a pandas dataframe and displayed the first few rows of the dataset. Its target variable is diagnosis, representing whether the breast tissue is malignant or benign. [2]

Next I plotted the distribution of the target variable to determine if there is a class imbalance problem in the dataset. The distribution of the target variable is as follows: 357 benign and 212 malignant. The proportion of benign to malignant is 62.74% to 37.26%. This indicates that there is a class imbalance problem in the dataset since the proportion of benign to malignant is not equal. [3]

Feature Analysis

The dataset constains 30 independent features and 1 dependent feature. After running T-tests on the features, I found that 5 features were not statistically significant. [4] The total number of instances in the dataset is 569.[5] This means that the dataset has more feature-space relative to the number of instances, which could lead to the curse of dimensionality issue. This issue can cause the model to overfit the training data.[6]

Missing Values and Outlier Detection

When investigating the dataset for missing values, I found that there were no missing values in the dataset.[7] I also applied an outlier detection method to the dataset. I used the Isolation Forest method to identify any outliers. The Isolation Forest method found 51 outliers. This could impact the classification task by causing the model to overfit the training data.[8]

[1] [Write-up 1] Describe the two CSV files associated with this lab, including the type of information each contains. Detail the specific data each file presents and its relevance to the lab objectives.

[2] [Write-up 2] Identify the target variable in this CSV file. What does this variable represent?

[3] [Write-up 3] Analyze and report the distribution of the target variable. Based on this distribution, discuss whether there is a class imbalance problem in this dataset.

[4] [Write-up 4] Determine and report the total number of predictive features included in the dataset.

[5] [Write-up 5] What is the total number of instances (rows) in the dataset?

[6] [Write-up 6] Evaluate whether the dataset faces a 'curse of dimensionality' issue, considering the number of features relative to the number of instances.

[7] [Write-up 7] Investigate the dataset for any missing values. Provide a summary of any findings.

[8] [Write-up 8] Select one outlier detection method discussed in this course and apply it to the dataset. Discuss whether any outliers were identified. If you find any potential outliers, explain their possible impact on the classification task.

Data Splitting and Preprocessing

Data Splitting Strategies

For any classification task, you can use the hold out method to split the data. Just split the set into a training and testing set, train the model on the training set, and then evaluate it on the testing set. Another option is to use k-fold cross-validation, where the dataset is divided into k subsets. Each subset is used as a validation set once, while the remaining k-1 subsets are used as training sets. The hold-out method is simpler and faster than k-fold cross-validation, but k-fold cross-validation is more reliable and provides a more accurate estimate of the model's performance. This method is especially helpful with small datasets and when you want to reduce variability in performance estimation. [9]

Implementing 5-Fold Cross Validation

For this specific dataset, I would apply 5-fold cross-validation to split the data. Because the binary classes are imbalanced, I would use stratified k-fold cross-validation to make sure each fold has the same proportion of benign and malignant instances.

I would do this by using the StratifiedKFold function from the scikit-learn library. This function splits the dataset into k folds while preserving the proportion of samples for each class. I would use each fold as the validation set one time and the rest of the folds as the training set for that round. I would do this for every fold, making sure that each one is used for training and validation throughout the process. [10]

Feature Normalization and Model Implementation

Feature normalization is scaling the features of the dataset so they have a mean of zero and a standard deviation of one. This is important because it can help the model converge faster and prevent the model from being biased towards features with larger scales. For logistic regression and random forest classifiers, feature normalization is not necessary because these models are not sensitive to the scale of the features. [11] That being said, feature normalization is useful in logistic regression because it will help spped up convergence. Because of this, I will normalize the features in the dataset.

If feature normalization is required, it should be performed after data splitting. This is because normalization should be done on the training set and then applied to the validation set. If normalization is done before data splitting, information from the validation set could leak into the training set, leading to data leakage and inaccurate model evaluation. The timing of normalization affects the training and validation proccesses by ensuring the model is trained on normalized data and evaluated on the same scale, as long as the normalization is done after data splitting so that the

[9] [Write-up 9] Discussion on Splitting Strategies: For any classification task, describe how you would split your data using the hold-out method. Compare and contrast the hold-out method with the k-fold cross-validation method. Specifically, under what circumstances would we prefer to use k-fold cross-validation over the hold-out method?

[10] [Write-up 10] Description and Implementation: For this specific dataset, explain how you would apply 5-fold cross-validation to split your data. After describing the process, implement this data splitting strategy in code. Ensure your explanation covers how the dataset is divided into folds, how each fold is used for training and validation across the iterations.

[11] [Write-up 11] Normalization Necessity: Discuss whether feature normalization is necessary when using logistic regression and random forest classifiers for this dataset. Explain your reasoning.

model is not exposed to the validation set during training. [12]

Model Implementation, Evaluation, and Analysis

Implementing Models with 5-Fold Cross-Validation

I implemented a logistic regression and random forest classifier using the dataset prepared in Part 2. I employed 5-fold cross-validation to train and validate the models, ensuring each fold served once as the validation set. The parameters for logistic regression and random forest models are as follows:[13]

Logistic Regression:

- Solver: The solver is the algorithm used in the optimization problem. The solver can be set to 'liblinear', 'newton-cg', 'lbfgs', 'sag', or 'saga'. For this problem, I tried lbfgs and liblinear.

Random Forest:

- n_estimators: The number of trees in the forest. The default value is 100. I tried values of 10, 20, 50, and 100.
- max_depth: The maximum depth of the tree. The default value is None. I tried values of None, 10, and 15.

The default parameter performed the best in accuracy(which I used for preliminary comparison), so I used those for the rest of the project.

I used classification_report() from sci-kit learn to calculate the accuracy, sensitivity, specificity, and AUC for each fold. [14] They are the following:

[12] [Write-up 12] Assume normalization is required, discuss whether it should be performed before or after data splitting and why. How does the timing of normalization affect the training and validation processes?

[13] What are the adjustable settings or parameters in the code for logistic regression and random forest models?

[14] [Write-up 13] Recording Performance Metrics: Record accuracy, sensitivity, specificity, and AUC (area under the curve) for both models for each fold.

Model	Fold	Accuracy	Sensitivity	Specificity	AUC
Logistic Regression (lbfgs)					
	1	0.956	0.977	0.944	0.960
	2	0.982	0.953	1.0	0.977
	3	0.991	0.976	1.0	0.988
	4	0.991	0.976	1.0	0.988
	5	0.982	0.976	0.986	0.981
Random Forest (default)					
	1	0.939	0.953	0.930	0.942
	2	0.982	0.953	1.0	0.977
	3	0.947	0.952	0.944	0.948
	4	0.965	0.952	0.972	0.962
	5	0.956	0.905	0.986	0.945

Statistical Analysis of Model Performance

I then calculated the mean and standard deviation of accuracy, sensitivity, and AUC for both models across all folds. [15] The results are as follows:

Logistic Regression (solver lbfgs)

- Accuracy: 0.98 +/- 0.00
- Sensitivity: 0.97 +/- 0.02
- Specificity: 0.99 +/- 0.01
- AUC: 0.98 +/- 0.01

Random Forest

- Accuracy: 0.96 +/- 0.03
- Sensitivity: 0.93 +/- 0.07
- Specificity: 0.98 +/- 0.02
- AUC: 0.95 +/- 0.03

To compare the performance of the two models, I used a paired t-test. I chose the paired t-test because it is used to compare the means of two groups that are related in some way. In this case, the two groups are the logistic regression and random forest models, and they are related because they were trained and tested on the same dataset. I used the ttest_rel() function from the scipy library. The accuracy and AUC were statistically significant, while the sensitivity and specificity were not. [16]

Feature Importance Analysis

Analyzing Feature Importance

To be able to extract the top three most important features from the random forest and logistic regression models, I used the feature_importances_ attribute for the random

[15] Comparative Analysis: Calculate the mean and standard deviation of accuracy, sensitivity, and AUC for both models across all folds.

[16] [Write-up 14] Statistical Testing: Choose an appropriate statistical test to compare the two models. Discuss your choice of test and explain why it is suitable for this analysis. Is there a significant difference in performance (focusing on accuracy, sensitivity, and AUC) between the models? Interpret the results.

forest model and the coef_ attribute for the logistic regression model. I visualized the feature importance for both models using a bar plot, sorting by the feature set (mean, standard error, or worst) and the feature name. I then sorted the features by importance and extracted the top five most important features for each model, and compared the results to see if they agreed on the top three most important features unordered out of their top five. When comparing the top three most important, they only agreed on one feature, 'Worst Radius'. The top three agreed upon features were 'Worst Radius', 'Worst Concave Points', and 'Worst Area'. The top three features for the random forest model were 'Worst Radius', 'Worst Area', and 'Worst Concave Points'. The top three features for the logistic regression model were 'Worst Radius', 'Worst Smoothness', and 'Worst Texture'. [17] The models did not agree on the top three most important features.

I could have also used sci-kit learn's SelectFromModel to select the top three most important features from the random forest and logistic regression models. This method selects the features based on the importance of the model's coefficients. I could have then compared the results to see if they agreed on the top three most important features.

[17] [Write-up 15] Discuss strategies to extract the top three most important features from the random forest and logistic regression models. Implement your strategies in code. For both models, do they agree on the top three most important features?