SML312 — Research Projects in Data Science: Mini Project #2

08/11/22

Jonathan Hanke

Justin H. Cai

Problem 1

(a). See code.

(b). As part of the EDA, I viewed the given description of the dataset, summary statistics of features, the distribution of the features, and features based on the corresponding target value.
In terms of distributions, I noticed that the distribution of the feature variables mostly followed either an approximately normal distribution or a right skewed distribution, where the 10 mean features seemed split between these two kinds of distributions. The distributions of the standard errors tended to be right skewed, which aligned with my expectations. The "worst" value distributions tended to align with the mean feature distributions, which aligned with my expectations.
In terms of features relative to the target, I noticed that mean radius, mean perimeter, mean area, and mean concave points tended to have a target value of 0 at higher values and a target value of 1 at their lower values. The rest of the mean variable features either exhibited this relationship to a much lesser degree or didn't show a clear relationship between its values and the target value. The error variables tended to have more outliers; perimeter error, area error, smoothness error, and concavity error are good examples. The "worst" value distributions, like before, tended to exhibit the same relationship with the target value as the mean variable features.
The most important features seem to be mean radius, mean perimeter, mean area, and mean concave points, along with their corresponding "worst" value variable features. This is because from the EDA, these features show the strongest correlation with the target variable, in this case a negative correlation–the larger values tend to have a target value of 0, and the smaller values tend to have a target value of 1. The relationships between the other mean and "worst" features with the target variable appear much less clear, and the error feature variables either have too many outliers or don't exhibit a clear relationship with the target variable, making them seem less important.
There are 357 tumors classified as benign and 212 classified as malignant as stated in the description, indicating a balanced dataset (i.e. no one category of tumor has disproportionately more than the other).

(c). See code.

(d). I chose to use the k-fold cross validation score (includes both accuracy and standard deviation) as my primary metric to measure "goodness", although I included other metrics such as accuracy, precision, recall, and normalized confusion matrices to provide even more context into the performance of the model. There are several reasons I chose to use the k-fold cross validation score. First, between accuracy, precision, and recall, I felt accuracy was the most important metric because I felt false positives and false negatives were approximately equally bad, and more importantly, the dataset is balanced (as seen in the EDA). If a benign tumor is misclassified as malignant, the treatment can potentially cause a lot of cell damage and negative health side effects as well as waste resources, while if a malignant tumor is misclassified as benign, the treatment could be insufficient and the tumor can spread. Having a balanced dataset is important if we want to use accuracy as a metric because if there are far more tumors classified as benign than malignant or vice versa, the accuracy score could be high but can hide a model's difficulty in classifying the minority kind of tumor. However, the generic accuracy metric does not account for overfitting, and a model with a high accuracy may not generalize well to real-world datasets. Thus, I chose k-fold cross validation, which splits the data into k folds, uses $k-1$ of them for training, and tested on the remaining data, and measures accuracy k times, once for each choice of fold for testing data. This ensures the most possible data is used for training, giving a more accurate model overall, and promotes generalizability of the model. I used $k = 10$ folds to achieve a balance bias and variation, and also because it is a pretty common and standard value used by data scientists. The average accuracy value between each of the k iterations is chosen as the overall accuracy, and the standard deviation provides a sense of how varied the accuracies are between folds. Thus,

Problem 1 continued on next page...

higher accuracy and lower standard deviation are optimal.

(e). See code.

(f). The random forest model had the highest cross-validation accuracy score at 0.960 on the training data, although none of the other models had accuracies below 0.93. Random forest also had one of the smallest standard deviation values of 0.0276, which is hardly worse than the smallest of the models at 0.218. Since all the models had very low standard deviations, I felt that discriminating between such slight differences was not as important as the values showed the models did not vary much between the folds, which is good. Thus, I considered random forest to perform best on the training data because it had the highest accuracy. In general, the random forest model performed very well, achieving an accuracy of 0.963 and a cross validation standard deviation of 0.033.

(g). The random forest model had the highest cross-validation accuracy score at 0.983 on the testing data, although none of the other models had accuracies below 0.89. Random forest also had the smallest standard deviation values of 0.035. Thus, I considered random forest to perform best on the training data because it had both the highest accuracy and the lowest standard deviation. In general, the random forest model performed very well, achieving an accuracy of 0.963 and a cross validation standard deviation of 0.033.

(h). The lowest accuracy value between the models trained on the training data was 0.932, and the lowest accuracy value between the models trained on the testing data was 0.895. The majority of models had accuracy scores above 0.93 on both the training and testing data. In addition, the highest standard deviation value across all models between both testing and training data was 0.085, indicating there is not much variation between the individual accuracy values across each fold iteration, implying better generalizability and not overlearning of any one particular set of folds or data. Because the accuracies are consistently high and the standard deviations are consistently low across both the testing and training data, I do not consider any of the models to have overfit.

Problem 2

(a). I chose to use the permutation feature importance method to determine which features are most important. The permutation feature importance is calculated as the decrease in model accuracy when a given feature's values are randomly shuffled, thus corrupting the dataset. If the model's accuracy shows a large decrease when a given feature is shuffled, then we can infer that the model is highly dependent on that feature, and conclude that feature is important. While decision tree and random forest models provide a feature importance attribute based on mean decrease in impurity, these importances can be high even for features that don't predict the target variable well because they are derived from training dataset statistics, a limitation that permutation importance avoids.
From the bar graphs, we see that feature 22 (worst texture) has the highest importance across three models, while feature 23 (worst perimeter) has the highest importance across two, and feature 27 (worst concavity) on one model. In addition, feature 13 (perimeter error) maintains relatively high importance across 3 models despite not being the highest of any. Thus, I consider these features the most important. Which features were important were not very consistent across models. The worst perimeter feature had the highest importance among the non-linear and non-tree based models of k-nearest neighbors and Naive Bayes, while worst texture performed well among logistic regression, support vector machine, and decision tree models. This does not align well with my predictions from part 1(b), as the only important feature matching with my prediction from 1(b) is worst perimeter. The difference may be due to the fact that just because a feature displays a relationship with the target value doesn't mean it is a great predictor of the target value.

(b). I consider the k-nearest neighbors model the most explainable primarily around its simplicity and ease of understanding both its conclusions and how it arrives at the conclusions. I feel that it is easiest for someone without a technical background in statistics, ML, or math to understand how the model works–by classifying a value based on the values and classifications of the nearest k values to it. It is both relatively easy to understand the notion of "nearest neighbors" and how the k-nearest neighbors model classifies values based on the values of a majority of its nearest neighbors. On the other hand, the Naive Bayes classifier requires some working knowledge of probability and Bayes' theorem, SVM requires some understanding of planar or higher dimensional spaces and optimization, logistic regression requires familiarity with the logarithm function and regression, and decision trees and random forests, although simple in principle, can be hard to understand how decision rules are calculated.

(c). It is extremely important for doctors to be able to understand exactly how a model classifying tumors works, the conclusions made by the model, the parameters/features the model is using, and any shortcomings of the model. An explainable model for classifying tumors goes a long way in ensuring doctors can properly interpret results, modify the model, and consequently provide the best possible treatment to patients.
Understanding which features are most important to a classification model are a big component in helping a doctor understand how a model works. The doctor can use this knowledge to evaluate whether the model is too biased towards one or several features, and use his/her own medical knowledge to appropriately tune which features are most important to use. Without an understanding of which features are most important to a model, a doctor is more likely to be skeptical of a model, or the model may not align with established medical science without the doctor's knowledge.

Problem 3

(a). See code.

(b). See code. The area under the curve is 0.99, which is very close to the highest possible value of 1, indicating that the support vector machine model is very good at distinguishing between benign and malignant tumors.

(c). See code. My metric of goodness is the k-fold cross validation accuracy as explained in part 1(d). Mathematically, accuracy is given by

$$\frac{\text{TPR} + \text{TNR}}{\text{TPR} + \text{FPR} + \text{TNR} + \text{FNR}}$$

and the k-fold cross validation accuracy is just the mean accuracy over each of the k folds. The ROC curve plots FPR against TPR, so in order to maximize accuracy, we want to maximize TPR and minimize FPR. In other words, we want to maximize $\text{TPR} - \text{FPR}$. The threshold that achieves the maximum of $\text{TPR} - \text{FPR}$ is 0.6409.

Acknowledgements

Honor Statement

This problem set represents my own work in accordance with University regulations.
— Justin Cai, 08/11/22

Collaborators

None