SUPERVISED LEARNING WITH CLASSIFICATION AND UNSUPERVISED

Michael Brosnan

CIS 435 PRACTICAL DATA SCIENCE USING MACHINE LEARNING

PROF KAKADE

Introduction

Breast cancer is a serious and unfortunately very common diagnosis in the United States and in the world. About one in eight women in the United States will develop breast cancer in their lifetime, and the diagnosis rate has increased from about 100 cases per 100,000 women in 1975 to about 120 cases per 100,000 women in 2018 (*Breast cancer statistics* 2022). While great advances have been made in breast cancer research in the past few decades, the issue still confounds physicians; and while mortality rates have decreased since 1975, about 25 cases per 100,000 women still end up in death (*Breast cancer statistics* 2022). The key to breast cancer survival is early detection and intervention, which can reduce the mortality rate and keep women healthier longer. As researchers continue to learn more about the disease and look to find answers, machine learning has become a viable option to help identify cancer at early stages. With increased usage of Electronic Medical records and a more openness to data sharing, data surround breast cancer has become more readily available. Physicians, scientists, and engineers are using this data and machine learning algorithms to better identify cancer in patients and help them determine the appropriate next steps once the cancer is detected.

Business Problem

The business problem that we have been presented, is to use the data provided to build a machine learning algorithm, or set of algorithms, to predict whether a person's cancer is benign or malignant. This will involve taking the data from the data set, running it through our algorithms, and producing a result that a provider can use to help in their analysis of the patient. The algorithm will work as a good reference for providers, specifically in cases where the provider disagrees with what the algorithm has decided. This should result in the provider double checking their initial analysis while also understanding why the algorithm decided the opposite of the provider. This will allow the provider to understand the reasoning of the algorithm and make the best decision will all the different information he or she has been provided.

The data that will be analyzed is a set of 569 individual and unique cases with certain attributes and a diagnosis. There are 23 attributes for each row, ranging from case ID to certain items that describe the cancer. It will then finish with the diagnosis column, which is a binary zero or one field. These will be the characteristics that we will explore and used to determine if a cancer is benign or malignant.

To accomplish our goal of providing a model that can identify if a cancer is benign or malignant, we will be following the Cross Industry Standard Process for Data Mining, or CRISP-DM. We will be going through the phases of Business Understanding, Data Understanding, Data Preparation, Modeling, Evaluation, and Deployment (Rodrigues, 2020). We began in this section with Business Understanding and will touch on the other items as we go, with the result, or Deployment, being the model. During the modeling and evaluation phases, we will use Jupyter Notebook to develop, train, and test different versions of our models. Then we will evaluate the

success of those models against one another, determining which is best to use for the business problem we have been presented.

Machine Learning Applications

Machine learning has been influential in many industries; but there might not be a more interesting industry than healthcare. While there are many use cases where machine learning can be, and is, implemented, there are also major concerns about privacy when it comes to health information: often there is human life at stake, and an error from a machine learning algorithm could cause harm. While these concerns are ever-present, machine learning has been hugely influential in healthcare.

One of the more interesting applications of machine learning in healthcare is predicting the likelihood of a particular patient being admitted to the inpatient floor from the Emergency Department. While many people think machine learning is all about finding disease or the likelihood of disease, an algorithm that can predict the likelihood of an inpatient admission can be very valuable as well. Researchers used serum levels of Urea, Creatinine, Lactate Dehydrogenase, Creatine Kinase, C-Reactive Protein, Complete Blood Count with differential, Activated Partial Thromboplastin Time, D Dimer, International Normalized Ratio, age, gender, triage disposition to ED unit and ambulance utilization as independent variables to investigate their impact on hospital admissions. They then made the score attributed to each patient available to physicians as well as the transport and bed planning teams. This allowed those teams to better anticipate and allocate beds in the inpatient departments; which allowed for faster transfer times for patient from the Emergency Department to inpatient floors, allowing them to receive a higher level of care quicker. This also opened Emergency Department beds quicker, allowing more patients to be seen (Feretzakis et al., 2021).

A second application of machine learning in healthcare is clinician decision support using a model to predict a patient's risk of in hospital mortality. Like the previous example, the model takes inputs from the patients Electronic Medical Record (EMR) and puts that into an algorithm to determine a patient's risk of dying during the visit. One of the interesting problems the developers of this algorithm had was the lack of data from people who left the hospital. They had to work with local outside organizations to get information regarding if patients the algorithm had identified as risky had in fact passed after they left the hospital. These inputs were crucial when training the model. This risk score could be provided to physicians to help them bring together several different variables together to make a better assessment of a patient's overall risk. One of the keys to providing these scores is making it actionable, as just providing a score does not help the providers. They should be able to see the inputs that resulted in the score so they can make the appropriate adjustments to a patients care to help that score go down, and the patient's health improve (Kent, 2020).

The final example is using deep machine learning to view CT scans and determine several different factors, including binary classification, multi-class classification, and segmentation of different parts of the image, which in our example is of the chest. This group of

data scientists used convolutional neutral networks to classify different CT images to the appropriate categories. This is a valuable tool because it allows for the algorithm to quickly run through CT images to determine certain characteristics of the image. Allowing the algorithm to categorize the different images frees providers to view more specific cases or to concentrate more on cases that are more in question and require more time with the individual patients (Draelos, 2020).

There are almost endless opportunities for machine learning when it comes to the healthcare sector. The application must be nonintrusive and ultimately extremely reliable because this is often a person life at stake. If you trust a machine learning algorithm to categorize if a patient has a disease or not, it needs to be extremely accurate, because too many false negatives or false positives can cause a huge issue. In other industries, a machine learning algorithm with a 70% success rate can be fine and usable, but in healthcare, you often need the algorithm to be correct almost 100% of the time, which is an almost impossible task.

Machine Learning Algorithms

To accomplish the task described in the business problem four algorithms were used- three that were supervised learning algorithms, one that was unsupervised. The Logisitic Regression, Decision Tree, and the Naïve bayes were supervised, while the KMeans was unsupervised. Each of these algorithms preforms differently with the data provided and produce different results. We will go over the advantages and disadvantages of each model in this section, then we will go into detail about how each model looked at the data and their overall performance.

The first machine learning algorithm used in this assignment was a logistic regression algorithm, which is a supervised learning technique used to determine results that are binary, like True/False and Yes/No. The algorithm analyzes the relationship between the different variables and the ultimate result and discovers what the linear relationship is between the items in the dataset, then uses a Sigmoid function to create non-linearity (*Sklearn.linear_model.logisticregression*). While a linear regression model fits a straight line to the data, a logistic regression model used an S-shaped curve to determine the result of the binary question. A logistic regression model is easy to implement and interpret, and can extend to multiple classes, which would be multinomial regression. It preforms particularly well with simple datasets that are linearly separable. It is also less likely to overfit the model, which should lead to more accurate results. The biggest limitation with a logistic regression model is that it assumes that the relationship between the dependent and independent variables is linear, and if that is not the case the results of the model will not be effective. The dependent variable in the logistic regression model is bound to a discrete number set because the model can only predict discrete functions. While we listed the simplicity of the model as a positive, it comes back as a negative if you are working with a complex data set with complex relationships, as this model is not adept at handling those complex sets (*Advantages and disadvantages of logistic regression* 2020).

The next algorithm used was a Decision Tree algorithm, "which is a non-parametric supervised learning method used for classification and regression (*1.10. decision trees*)." Decision trees are easy to use and easy to understand, especially because they can be visualized. Not much data preparation is required, and it can operate with both numerical and categorical values, which is especially key because many other algorithms and models are not capable of handling both types of values. Another advantage is the fact that a decision tree is not a black box- it can be reviewed and explained. Similarly, it is possible to validate the model using statistical tests, so it is possible to easily understand how well the model is performing and if any training has worked to improve it. Decision trees are often susceptible to over fitting, as it is easy to create decision tress that are overly complex, which will generalize the data. Outliers in the data can also cause issues for decision trees, as they will try to account for those outliers, which could alter the result of the tree negatively. This along with the fact that some classes of the tree may dominate the tree, which will create a bias in the analysis (*1.10. decision trees*).

The third algorithm used was a Naïve Bayes algorithm, a supervised machine learning technique which uses conditional probability to calculate probability of other variables. This is a commonly taught theorem in statistics and calculus courses. The goal is to calculate the probability of an event occurring given a different event has already occurred. The main assumption of the Naïve Bayes is the assumption of independent predictors, and when this holds true Naïve Bayes outperforms most other models. It performs well even though it requires little training and is easy to implement compared to other models. While we mentioned how well the model performs if the assumption of independence holds true, in real life that assumption is rarely true. Naïve Bayes also suffers from Zero Frequency, which means the model assigned a zero probability to a variable and is unable to make a prediction (Kumar, 2022).

K-means was the final algorithm and model used, which is an unsupervised machine learning technique. This puts the different data points into clusters with like characteristics which can then be analyzed. It does this by starting with a group of randomly selected centroids to start each cluster, then performs repeated calculations to optimize the centroid. This process stops when the centroids have stabilized or when the number of defined clusters has been reached. K-Means is easy to implement, and when large number of variables are present, it is quicker than other clustering techniques. It can also produce tighter clusters and can change the clusters when the centroids are recomputed. Often it can be difficult to predict the number of clusters that will be needed. It is also sensitive to scale in that fact that if you rescale your dataset your results can change completely. This means that extra attention will be needed when scaling your data (Garbade, 2018).

Data Preprocessing Discussion

To better understand the data that was to be used for our machine learning algorithms, we needed to do some preparation and exploration to make sure the data is ready for use. Like most data projects, we imported the necessary python libraries that we would use for the

analysis, then pulled in our data from the bdata.csv file. To begin our exploration, we looked at the columns included in the data sets and what data type those columns were. We then looked at different metrics for the columns including the count, mean, standard deviation, minimum, maximum, and the different quartiles. We also looked to see if there were any empty cells and filled any empty columns with null.

One of the most important things we did while preparing the data was to identify the independent and the dependent variables. With the diagnosis being the end result, we used this as the dependent variable (Y); the other columns being the X variables, or the independent variables. This means that our models would use these inputs and test them against what the dependent variable would be to identify which had the largest influence. This is important because it would help determine how accurate our model was and its efficiency.

We also split the data into a test and a training set. The purpose of this is to have a large set of about 80% of the data to train the model. This will allow the model to learn on as much data at possible, while keeping some data to the side to test the model. We want to test with enough data to get an idea of how good the model is, and that test data cannot be apart of the training data, because that could cause the model to overfit the data, and your test score will not be usable.

Explaining Metrics

There were a few metrics that were used to judge the performance of the different models, specifically the supervised models. After we processed the data and configured our algorithm, the first thing we did was attempt to identify how accurate the model was and what the important features of independent variables were. To do this, we looked for the accuracy of the training set and the test set, to compare how well the model performed on the test and training set. It is not unusual for the model to perform better on the training test compared to the test set, because the training set is usually larger; and with more data the better the model is likely to perform. The test set is smaller and may be susceptible to outliers or larger variation. When we are judging how the model performed, we will use the test set as the performance measure. This is a simple calculation, taking the number of correct predictions divided by the number of total predictions (Mishra, 2020).

The next metric is one of the more important metrics and that is the confusion matrix. The confusion matrix breaks the results of the model down in four different categories: the true positives, the true negatives, false positives, and false negatives. The true positives are the outcomes that the model predicted a Yes outcome, and the true outcome was Yes. The true negative is the same, expect that the model predicted No, and the true result was No. These are accurate predictions made by the model. False positives are cases where the model predicted a Yes result, but the actual result was No, and a false negative are times when the model predicted No and the result was Yes. These are important to understand because they not only look at accuracy, but also what kind of wrong results the model is predicting. It allows you to better break down the results so you can understand where the deficiencies are and

rework your model to adjust for those mistakes (Mishra, 2020). We also see the importance of each feature according to the model. This can help you identify which inputs were most important and were taken into effect when running the model.

Once we understand the confusion matrix we moved to precision, recall, F1-score, and support. Precision is the number of true positive predictions made by the model divided by the number of positive predictions made by the model, both true and false. This give you an idea of how accurate the model is when it labels a value as positive. Recall is the number of true positives divided by the number of true positives and false negatives. This gives you a number related to the number of positives predicted to the number of overall positives, showing you how good your model is at predicting the positive outcomes. These two numbers are used in the F1-score. The F1-score is the harmonic mean between the precision and recall. This number shows you how precise and robust your model is, which a higher F1-score the better the model (Mishra, 2020). Finally, the support is the number of occurrences of each class in where Y is true, so in our case the number of diagnoses of 1 and 0 (Mishra, 2020).

The final metric is also a very important metric and that is the receiver operator characteristic (ROC), which shows in graph form the performance of the model against a random guess. The area under the ROC curve (AUC) is an indication of how well the model performed, with an AUC of 1 being perfect. This is another metric to describe how well you model performs and can help you evaluate how ready it is for production.

Unsupervised machine learning is often harder to evaluate. We used an elbow curve to identify the number of clusters that we should use, which was two, and then looked at the centroids of those clusters. Plotting those showed what those clusters looked like on a plot, making them identifiable and how far they are from the center of the cluster. We looked at the inertia of the model to measure how coherent the clusters are (*2.3. clustering*).

Interpreting Results

Overall, when looking at the models we created and scores for the different metrics we calculated, it is important to remember the business problem at hand is to build a model that can predict whether a person's cancer is benign or malignant. While all the models created seem to do a good job of this, we need to determine the best, and then see if that is good enough for the business users to use, or if it needs to improve. Our models were for the most part pretty accurate. The Logistic regression model had a test set accuracy of .9231, meaning it was correct on about 92% of predictions. However, the recall rate of this model for a diagnosis field of "1" was only 89%, which means it missed some positive results, but is was 99% for a diagnosis of "0." It also had a AUC of .988, which means it is a very good model at predicting a result compared to the 50/50 guess.

The decision tree was similarly successful, with the test set accuracy for the model being about 93%, which is slightly better than the logistic regression model. However, the logistic regression model faired better when looking at the recall rate, which shows the Decision Tree

as having recall rate of 89% when looking at a diagnosis of “1” and 95 when looking at a diagnosis of “0.” It also has a AUC of .942, which is lower than the Logistic Regression, showing that it may not be as strong of a model.

The Naïve Bayes was just as accurate as the Decision Tree for the test set at 93%, but it performed worse when it came to recall rate for a diagnosis of “1” with a rate of 85%, but 98% for a diagnosis of “0.” However, it has an AUC of .973, which is better than the decision trees, but less than the logistic regression model. This means it is overall a very good model, like the other two, just slightly below the logistic regression.

The K-Means is somewhat harder to compare to the other models because it does not use the same metrics for testing, instead it used the sum of square errors. We used an elbow graph to show how many clusters would be best, and the result of that was two. When we looked at the actual graph it showed the different clusters but did not provide us with a ton of information regarding accuracy. We also looked at the inertia score, and it was very high, when closer to zero is better. This is most likely because we have a high number of attributes. It is possible that is we limited the number of attributes the score would improve.

Recommended Steps

The recommended steps for this business problem are difficult because of the massive implications that this model will have. In the healthcare industry, people’s lives are at stake, and a predictive model being wrong could have a massive effect on a person’s life. If the model has a false positive result, that person will most likely have more tests done and they will be able to better identify that it was a false positive, but that person will still have the stress of thinking they have cancer. If there is a false negative result, that person could be very ill, but the cancer would not have properly been identified, and that person’s life could be in danger. The supervised models we build were all very accurate in identifying the diagnosis correctly when viewed against other machine learning algorithms, but the fact that a live is at stake means that 92% or 93% accurate might not be good enough.

In terms of next steps, I would recommend that we continue to collect data to create an even larger data set so we can have more information to train the model. We will then continue to optimize the model with the newly collected data and continue to improve the model. While we are doing that, I would recommend that we implement the Logistic Regression model into a shadow environment that receives a data feed from the production environment. It is standard practice for most Electronic Medical Records to have shadow environments and using the Logistic Regression model in this environment will allow us to see its performance on patients in the live system without being in production, and it will allow us to work with the physicians to make sure we find a place for the results of the algorithm in their workflow and that the results are actionable for the physicians. I recommend the Logistic Regression model because while its performance on the test data was one percent worse than the Decision Tree and the Naïve Bayes, it performed better on the recall test and had an AUC closer to one than the other models. The recall rate of .89 for a diagnosis of “1” is somewhat concerning, that is why we would like to continue to gather data and optimize the model. If a model has to be

implemented immediately, I would still recommend the Logistic Regression model, but I would ensure that there is a workflow in place for physicians to validate the results.

References

1.10. decision trees. scikit. (n.d.). Retrieved February 6, 2022, from https://scikit-learn.org/stable/modules/tree.html

2.3. clustering. scikit. (n.d.). Retrieved February 6, 2022, from https://scikit-learn.org/stable/modules/clustering.html

Advantages and disadvantages of logistic regression. GeeksforGeeks. (2020, September 2). Retrieved February 6, 2022, from https://www.geeksforgeeks.org/advantages-and-disadvantages-of-logistic-regression/

Breast cancer statistics. Susan G. Komen®. (2022, January 31). Retrieved February 6, 2022, from https://www.komen.org/breast-cancer/facts-statistics/breast-cancer-statistics

Draelos, R. (2020, August 4). *Chest CT Scan Machine Learning in 5 minutes*. Medium. Retrieved February 6, 2022, from https://towardsdatascience.com/chest-ct-scan-machine-learning-in-5-minutes-ae7613192fdc

Feretzakis, G., Karlis, G., Loupelis, E., Kalles, D., Chatzikyriakou, R., Trakas, N., Karakou, E., Sakagianni, A., Tzelves, L., Petropoulou, S., Tika, A., Dalainas, I., & Kaldis, V. (2021, June 28). *Using machine learning techniques to predict hospital admission at the Emergency Department*. arXiv.org. Retrieved February 6, 2022, from https://arxiv.org/abs/2106.12921

Garbade, D. M. J. (2018, September 12). *Understanding K-means clustering in machine learning*. Medium. Retrieved February 6, 2022, from https://towardsdatascience.com/understanding-k-means-clustering-in-machine-learning-6a6e67336aa1

Kent, J. (2020, March 26). *How machine learning is transforming clinical decision support tools*. HealthITAnalytics. Retrieved February 6, 2022, from https://healthitanalytics.com/features/how-machine-learning-is-transforming-clinical-decision-support-tools

Kumar, N. (n.d.). *Advantages and disadvantages of naive Bayes in machine learning*. Advantages and Disadvantages of Naive Bayes in Machine Learning. Retrieved February 6, 2022, from http://theprofessionalspoint.blogspot.com/2019/03/advantages-and-disadvantages-of-naive.html

Mishra, A. (2020, May 28). *Metrics to evaluate your machine learning algorithm*. Medium. Retrieved February 6, 2022, from https://towardsdatascience.com/metrics-to-evaluate-your-machine-learning-algorithm-f10ba6e38234

Rodrigues, I. (2020, February 20). *CRISP-DM methodology leader in Data Mining and Big Data*. Medium. Retrieved February 6, 2022, from https://towardsdatascience.com/crisp-dm-methodology-leader-in-data-mining-and-big-data-467efd3d3781

Sklearn.linear_model.logisticregression. scikit. (n.d.). Retrieved February 6, 2022, from https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html