Data 650 – Fall 2019

Assignment 2 – SparkML Logistic Regression

Kenneth Lulie

Professor Purushothman

University of Maryland University College

October 20, 2019

**Assignment 2: Part One**

1. **Discuss the importance of data exploration and visualization prior to running the logistic regression method.**

Data exploration and visualization prior to running any machine learning method is essential (Han, 2011). By exploring the data we are able to identify potential issues, remedy them before the model is trained, and pick the correct model for the characteristics of the data. Additionally, it helps us become familiar with the data which is always essential to any applied purpose of data science. Visualization in particular is useful because even an untrained human eye can pick up on patterns when they are displayed visually, but might miss them or get overwhelmed when presented the same information numerically.

1. **What do the Titanic data exploration results reveal about the relationships between the likelihood of survival and passenger data?**

The data exploration showed a clear relationship between Passenger class and Gender with the likelihood of survival of 1st class passengers being much more likely to survive than 3rd class passengers and with women being much more likely to survive than men. The impact of age on survival is harder to interpret but it seems like there is a possible relationship between very young children and very old adults being less likely to survive. This would indicate a non-linear relationship for Age that Logistic Regression may not be able to handle without additional data processing such as bucketing (Han, 2011).

1. **Discuss the logistic regression method results, including the classification accuracy for training and test set.**

The results of the train and test set can be seen in the chart below. The accuracy of the model was 82% on the train dataset and 78% on the Test dataset indicating a drop in accuracy of 4%, likely due to over-fitting.

On the surface, this may seem like a reasonably accurate model based on a relatively small dataset. However, especially in situations like this where the target variable is unbalanced, it’s important to compare how the model performed against simply predicting the outcome based on the total distribution, and on just one variable (Han, 2011). In this case, if we simply predict that all passengers died we would get a 58.5% accuracy. Further, if we look at just the gender of the passenger and predict that all women live and all men die, we get an accuracy of around 77%. This is essentially the same performance the model had on the Test dataset.

By reviewing the precision and recall, we can see that the model is not simply using such an un-sophisticated method as predicting solely based on gender. The precision and recall on the test dataset are both 72.1% indicating that the model identified successfully 72.1% of all cases where the passenger survived, and that 72.1% of all predictions where the passenger survived was correct. Further, the F1 measure was 72.1%. The F score is a weighted average of precision and recall, and takes into account both false positives and false negatives. It is more useful for measuring performance on imbalanced datasets. When compared to the Train set, we see that that the F1 measure dropped by about 6.5%, showing another measure of over-fitting.

The ROC curve also shows that while the model can capture about 60% of the positive observations with a very small false positive rate, and 80% with a modest false positive rate, the false positive rate dramatically increases when trying to capture the last 20%. Essentially, this is saying that you could identify about 60% of the passengers who survived with a high level of confidence that you are identifying very few false positives, but after that the performance drops off dramatically.

+----------------+---+---+

|label\_prediction|0.0|1.0|

+----------------+---+---+

| 1.0| 17| 44|

| 0.0| 77| 17|

+----------------+---+---+

|  |  |  |
| --- | --- | --- |
| Titanic Logistic Regression Results | | |
|  | **Train** | **Test** |
| Accuracy | 82% | 78% |
| Error | 17.60% | 21.90% |
| Precision | 79.40% | 72.10% |
| Recall | 78.00% | 72.10% |
| F1 Measure | 0.788 | 0.7213 |
| Area Under PR |  | 67.50% |
| Area Under ROC |  | 0.77 |

1. **Is logistic regression suitable for this problem? Why or Why not?**

Logistic regression is a standard classification suitable when the predictor variables have a linear relationship with the target variable (Han, 2011). In this case however, it does not seem to give powerful results. Without further refinement it is unknown if logistic regression is capable of increasing its performance for this application, but as mentioned above the performance on the test set is comparable to simply using the gender to predict survival. Processing the Age variable to be categorical through bucketing could increase accuracy if there is a non-linear relationship such as very young and very old being more likely to perish.

It is difficult task to determine how accurate any algorithm would be considering the data set is only approximately 800 observations with three predictor variables each. Without exploring other algorithms for this application it is not able to be determined if logistic regression is unsuitable or if there just isn’t enough data to build a strong predictive model.

1. **What alternative machine learning methods could be suitable for this problem? Consider at least 2 alternative methods.**

Other alternative machine learning methods that could be suitable for this problem include Naïve Bayes, Gradient Boosted Decision Forests, and a standard classification Neural Net (Han, 2011).

Naïve Bayes in particular is likely well suited as it performs well in cases where the data sets aren’t large. Additionally, as Naïve Bayes assumes each variable is as important as another and each variable is independent to each other variable it is well suited for a small amount of hand picked variables such as the data used for this analysis. Further, as the model is simple it avoids over-fitting which is a significant risk if a more complicated method is used to predict on this relatively small dataset.

Gradient Boosted Decision Trees are suitable for either regression or classification and have been recently touted on their performance for most applications. Gradient Boosted Decision Trees work as an ensemble model of decision tree, which combine to make a strong model out of weaker models. They are generally a strong performer for many applications due to the ability to deliver powerful results with minimal data-processing. However, with only three variables used in the model, we are likely to experience over-fitting.

Last, we could also use a simple neural network. If the neural network is kept to a small size with only one or two hidden layers it could likely avoid over-fitting. Additionally, Neural Networks are capable of handling non-linear relationships which could be of value getting more prediction out of the age variable.

**Assignment 2: Part Two**

1. **Define the purpose of the study and the target variable. Which variables are used as predictors?**

The purpose of the study is to help identify risk factors before a child is born that the child could be born with a low birth weight. Doctors could then use these risk factors to help identify at risk pregnancies and intervene early enough to improve the outcome. Further we could educate the public or create policies to change behavior in mothers to create more positive outcomes. The target variable is if the baby was born with a low or non-low birth weight with the cutoff being 2,500 grams or around 5.5 pounds.

The predictor variables available in the dataset are the mother’s age in years, the mother’s race, the mother’s smoking status during pregnancy, the mother’s history of premature labor, the mother’s history of hypertension, the mother’s Uterine Irritability status, and the mother’s number of physician visits during the first trimester.

1. **Interpret the data exploration and visualization results. What did you learn about the low birth weight data from data exploration, including possible relationships between predictors and the target variable?**

The data exploration and visualization showed that Age, Race, Smoking Status and Number of Physician Visits appeared to be correlated to birth-weight outcome. Specifically, Blacks were most likely to have underweight births, followed by mothers of ‘other’ races, followed by whites. Smokers during pregnancy were more likely to have low-weight births than non-smokers, and women who had 1 or 2 visits to the doctor during the first trimester appeared to be less likely to have an underweight birth than women who had 0 or more than 3. The review of Age showed a different distribution of women who had normal birth-weight outcomes than those who had low-birth weight outcomes.

Additionally, the target variable is imbalanced with 59 low-birth weight outcomes to 130 normal birth weight outcomes.

It’s important to note that the Age variable and the Doctor Visits variables appear to be non-linear in their relationships, and as such logistic regression may have trouble capturing the relationship without further data-preprocessing (Han, 2011).

The other variables included of History of Premature Labor, History of Hyper Tension and Presence of Uterine Irritability were not included in the dataset due to the imbalanced nature of these variables in the dataset, specifically very few of the observations were not null. This is an issue for this data exploration as we are trying to gather insight into risk factors, and the low number of observations affected means that the correlation could simply be random chance rather than an actual casual relationship.

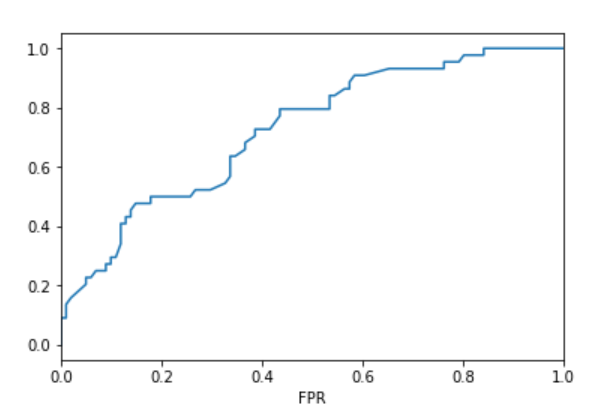
1. **Discuss the method results, including the classification accuracy for training set and test set and model evaluation metrics (precision, recall, ROC curve area).**

The performance statistics of the model on the train and test data are presented in the table below. The most important statistic to review to indicate performance on an imbalanced dataset is the F1 Measure, which is a weighted measure of precision and recall (Han, 2011). The F1 measure is a good indication of how the model performs against a strategy of guessing the most likely outcome for every observation. Simply reviewing accuracy would not quickly allow a review of this comparison. Based on this metric we can say that the model performed very poorly on the train set, and even worse on the test set at F1 measures of 34.92% and 25% respectively.

Interestingly, a system of just predicting the most likely outcome, normal birth-weight outcome, would have earned a 64% accuracy on the test set out performing the model. When we review the very low percentages of precision (37.5%) and recall (18.75%), we can see that the model is following a strategy of over-predicting the normal outcome. Reviewing the numbers further, the model results show that of the 8 births predicted by the model to be low weight, only 3 of them actually were.

|  |  |  |
| --- | --- | --- |
| Low Birth Weight Logistic Regression Results | | |
|  | **Train** | **Test** |
| Accuracy | 71.70% | 60.00% |
| Error | 28.27% | 40.00% |
| Precision | 57.80% | 37.50% |
| Recall | 25.00% | 18.75% |
| F1 Measure | 34.92% | 25.00% |
| Area Under PR |  | 36.71% |
| Area Under ROC |  | 50.75% |

The review of the ROC curve below also shows that the model is only reliable for the 15% of positive predictions it is most confident about. Again, this is a strong signal that this model did not capture the relationships properly (Han, 2011).



1. **Is the logistic regression method suitable for this study? Why or why not?**

Logistic regression method does not appear suitable for this study. Specifically, some of the variables appear to have non-linear relationships such as Age and number of Doctor Visits. Further, the number of observations to train off of are fairly low when the total data set is only 189 observations, especially when those observations at imbalanced in the target variable outcome.

Further exacerbating this issue, only four variables were chosen to be predictors, with the remaining being discarded due to the very imbalanced distribution in those variables.

1. **How would you improve the accuracy of your model?**

The most important approach to increase the accuracy of this model would be to bucket the age variable, and convert the doctor visit variable to hot encoding. This would allow the logistic regression to more properly model the non-linear relationship of these variables.

The other important way to increase the accuracy would be to drastically increase the number of observations, and increase the number of variables. At 189 observations with only four predictor variables each there is much room for improvement (Han, 2011). We could also change the parameters of the model, and the number of iterations spent training.

1. **Discuss at least 2 alternative machine learning methods that could be suitable for this problem and explain why?**

As mentioned in Part 1, Naïve Bayes would likely perform better than logistic regression for datasets where the number of observations is low and the number of features is low (Han, 2011). As there just might not be enough data to create a model that will perform well by learning the relationships, Naïve Bayes might perform better by using the prior probabilities to predict the outcomes. Additionally, Naïve Bayes is robust against over-fitting due to the nature of the algorithm using prior probabilities rather than a complicated model to predict outcomes.

Again, as mentioned in Part 1 Gradient Boosted Decision Trees could also be suitable for this issue. However, the risk of over-fitting would be extremely high, and the number of branches and depth of the trees would need to be aggressively trimmed to avoid over-fitting. However, if more data became available it would be a great choice for its ability to tolerate little data preprocessing and still understand complex non-linear relationships.

A Support Vector Machine could also potentially perform well on this limited dataset for classification. In particular, the way SVMs create hyper-planes with different kernels could assist it in dealing with non-linear relationships. Further, with only four predictor variables we should be able to avoid over-fitting.

**Assignment 2: Part Three**

1. **What is overfitting? What is the impact of overfitting on model performance? Discuss at least 2 approaches to avoid overfitting the model.**

One way to understand overfitting in an accessible definition is that it occurs when a model ‘memorizes’ the data rather than actually learning the relationships between the features and the target variable that allow it to correctly predict on data the model hasn’t seen before (Han, 2011). Overfitting is shown when the model has a higher accuracy on the training data set than on the test data set. In fact, the main reason test data sets are standard is to allow testing for overfitting that occurred training the model.

For example, decision trees when given too many branches and allowed to grow too deep are notorious for their ability to over-fit. If enough complexity is allowed in the model, many datasets will get a 100% accuracy on the training set, and then a terrible performance on the test data set. One way to control overfitting in decision trees is to limit the number of branches, the maximum depth, and to set a minimum number of observations in each end node.

To avoid overfitting it is important to manage the relationship between the amount of data available to train, and the complexity of the algorithm training on the data. If you can increase the amount of data available to train on, it will usually help with overfitting. On the other hand you can also control the complexity of the algorithm training on the data. You can either pick simpler models which have less ability to over-fit such as Naïve Bayes or carefully tune the parameters of more complicated algorithms such as reducing the number of hidden layers in a neural network.

Finally, you can also review the data and ensure that no variables which can be used to identify the specific observation but not likely to have a predictive value is included. Examples include unique identifiers such as social security numbers or telephone numbers.

1. **Discuss 5 (five) key differences between HDFS and Object Storage.**

The Hadoop File System or HDFS is the file system used to store data in Hadoop. The main characteristics of HDFS is that it designed to be a distributed file system meant to run on commodity hardware. It may also be referred to as Block Storage as it breaks larger data objects into smaller blocks. Object Storage instead of splitting a file into blocks stores the file as one Object consisting of the data, the metadata, and the unique identifier (Druva, 2019).

One difference between HDFS and Object Storage is that HDFS allows you to incrementally edit one part of a file easily, as you only have to change the block where the change is taking place. Object Storage instead requires that you edit the entire object, requiring it to be accessed, updated, and then completely rewritten. This means that HDFS can have better performance if you are frequently editing the data.

Another difference is that block storage can be directly accessed by an operating system as a mounted drive, while Object Storage can only do the same with significant performance implications.

A third difference is that due to Object Storage storing a file as one object in its entirety, it is generally more resilient than block storage. Typically an object will be stored in at least three nodes so as long as one remains up the object will remain available. It also simplifies comparing the object for data degradation, and if necessary the other versions can be used to restore the object. However, this does mean that it can use more storage to store the same amount of data.

It is also important to note that there are different use cases for HDFS and Object Storage where one will be superior to the other. Generally, object storage is better for data which will be written once but read many times such as video, images, music, or static web content. However, HDFS is superior for data which will be constantly edited such as transactional data. In fact, Object Storage does not natively support the locking and sharing functions required to maintain a single point of truth for files. It is important to be familiar with both HDFS and Object Storage moving forward, so that the proper tool can be used for the proper purpose.

1. **We may use R, Python, Scala, and Java programming languages with Spark. Discuss the pros and cons of each language.**

R, Python, Scala, and Java are all programming languages widely used, and usable with Spark. Each language has its own set of characteristics and situations where it is more or less suitable (David, 2018).

R is a scripting language that has good support for data science tasks, especially with statistical analysis. The language excels at data exploration, data plotting, and research analysis. R is sometimes also generally considered to be a language easy to learn. It was designed specifically by statisticians for support in statistical analysis. While some people warn against using R in production, it is being used for production purposes more and more as support matures. However, it would be recommended to do research to ensure proper R support for your project before committing to using it in a production capacity. Additionally, if the project is large enough it is likely less suitable than Scala or Java.

Python is sometimes called the second best language at everything. It is a general purpose language, but its ecosystem and support has evolved to be comparable to R for general Data Science tasks while having easier support to be integrated with other systems. It is often used when a model will be put into production for small or medium size projects. Other characteristics include being known for its elegant syntax and readable code, and being a popular language to learn first for aspiring programmers.

Scala is an acronym for ‘Scalable Language’. Scala was specifically designed to be used in large scale productions and is more efficient than code written in R or Python. Specifically, because Scala is a compiled language it is faster than R and Python which need to be interpreted. However, many people prefer the support of R and Python for general data exploration and research. Scala is often the preferred language to use with Spark.

Java shares the same characteristics of Scala but with some differences. This is because Scala is based on Java but was specifically designed for large data science projects using Hadoop. Where Scala is a mixture of object oriented and functional programming, Java is a general purpose object oriented language. Further, Java does not support lazy evaluation or operation overloading.

In summary, many of these languages are capable of the same tasks. However, each one has use cases where it is preferred over languages. Generally, if you are doing a small personal project you would want to use Python, if you are doing quick data analysis on a laptop use R or Python, if you are managing a project many people are working on where speed of development isn’t the priority but stability and scalability are, you would use Java or Scala.

**References**

David, T. B. (2018, February 15). Which Programming Language Is Better: R, Scala, or Python? - DZone Big Data. Retrieved from https://dzone.com/articles/which-programming-language-is-better-r-scala-or-py.

Han, Kamber, and Pei (2011). Data Mining: Concepts and Techniques, Third Edition Retrieved September 14, 2018 from <http://hanj.cs.illinois.edu/cs412/bk3/01.pdf>

Object Storage versus Block Storage: Understanding the Technology Differences. (2019, September 19). Retrieved from https://www.druva.com/blog/object-storage-versus-block-storage-understanding-technology-differences/#targetText=Object storage, by contrast, doesn,into raw blocks of data.&targetText=Another key difference is that,without significant degradation to performance.