Potential Models for Multiclass Classification

Unsurprisingly predicting the cause of the fire will be more complicated than a simple binary classification. This database classifies the cause of a fire into 13 classes: Debris burning, lightning, arson, equipment use, campfires, children, smoking, railroad, power line, fireworks, structure, miscellaneous and undefined. Our analysis tests numerous models' ability to accurately predict the cause of the fires. This document discusses the models we consider, their assumptions, tuning parameters (if applicable), advantages and disadvantages.

Top Results:

Decision Trees

We start with the discussion of decision trees because they are the foundation for the next two methods and it is vital that the reader understands this method first. The decision tree algorithm has gained popularity for its ability to fit complex datasets, (1). One can think of a flow chart where as the model takes a data point through it follows a series of binary questions until it results in a selected class, (1). There is a large chance for overfiting/bias when using this method, (1). Some methods use techniques to control over fitting such as in our next two models, one of which ended up with the leading predictive capability, (9).

Random Forest

This is one of the most powerful machine learning methods, (1). Think of random forests as developing multiple trees which all make a recommendation for a particular class selection for a data point, (1). Then, the class with the most votes is what the data point is where the data point is assigned, (1). A more statistical explanation is simply this is a bagging method made for decision trees, (1).

Ensemble Extra Trees (Leading in predicative capability)

This is a particular type of random forest method, (1). Utilizing the ensemble method of either extra trees or random forest attempts to add some additional randomness within the steps of developing the individual trees to reduce the variance. Of course, this can become an issue since it increases bias. The use of averaging can help control the amount of over-fitting, (9).

Less Favorable:

QDA

Quadratic Discriminant Analysis, QDA, is a simple model to fit with no parameters that require tuning. It is within the same family of models as LDA but instead of using a linear decision boundary it uses a quadratic one, (6). As opposed to some of the other methods we worked with it naturally is capable of handling multiclass

classification, (6) as opposed to some models that have to use one versus the rest procedures, which can become lengthy.

Logistic

This is a commonly used classification algorithm and it in many cases is one of the first models students will be introduced to for classification. Typically this model is a decent candidate for binary classification. It can be expanded to a multi-classification problem by using one versus the rest methods for each of the n classes. This method fits a binary classification model for each of the predictors, (4). A simple way to think about one versus the rest is that for each class there is a logistic model that predicts whether a data point is in this class or in one of the other classes.

Ridge Classifier

This model develops a classifier based on the theory of ridge regression. It also uses one versus the rest methods for the n classes when it is applied to multiclass classification problems, (5). The parameters will be optimized if you using the CV version of the function. It is important to note that this method assumes that the data is centered.

Bayesian Classification

This model is relatively simple. It uses Bayesian statistical techniques to determine probabilities for each data point indicating the likelihood that it is in a particular class. It is then sorted into the class with the highest probability of being the correct match. This method performs very well in spam mail classification, (7).

Sources

1. Géron, Aurélien. *Hands-on Machine Learning with Scikit-Learn and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*. OReilly, 2018.
2. "Sklearn.tree.DecisionTreeClassifier¶." *Scikit-Learn*, scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html.
3. Scikit-learn: Machine Learning in Python, Pedregosa *et al.*, JMLR 12, pp. 2825-2830, 2011.
4. "Sklearn.linear_model.LogisticRegression¶." *Http://Scikit-Learn.org/*, scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html.
5. "Sklearn.linear_model.RidgeClassifier¶." *Scikit-Learn*, scikit-learn.org/stable/modules/generated/sklearn.linear_model.RidgeClassifier.html.
6. "1.2. Linear and Quadratic Discriminant Analysis¶." *Scikit-Learn* , scikit-learn.org/stable/modules/lda_qda.html.
7. "1.9. Naive Bayes¶." *Scikit-Learn*, scikit-learn.org/stable/modules/naive_bayes.html.

8. "3.2.4.3.1. Sklearn.ensemble.RandomForestClassifier¶." *Scikit-Learn*, scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html.
9. "3.2.4.3.3. Sklearn.ensemble.ExtraTreesClassifier¶." *Scikit-Learn*, scikit-learn.org/stable/modules/generated/sklearn.ensemble.ExtraTreesClassifier.html.