Assigned: 03/5/2021
Due: Fri. 03/19/2021, 11:59pm

This project will cover topics of classification.

Submission Requirements:

You will again have a final document submission that includes text responses to questions, tables, R/Matlab/Python code used to calculate answers, and figures.

Formatting of submissions will follow the same approaches as those used in prior assignments. Namely: Matlab + markup, Matlab LiveScript, Rmd, Sweave, Jupyter notebook, Colab notebook. Any other packages or tools, outside those listed in the assignments, examples, or those for reproducible research should be cleared by Dr. Brown before use in your submission.

For this project you are allowed to work in **groups of up to 3**. All students must sign-up on Canvas into a group (e.g., P4-Rho, P4-Tau, etc.).

Name your main submission files as *P4_GroupName*, create a zip-file called *Project4_GroupName.zip* and submit on Canvas. For example, if I was using R, and member of GroupChi, I would submit either:

- *P4_GroupChi.Rmd*, *P4_GroupChi.pdf*, or
- *P4_GroupChi.Snw*, *P4_GroupChi.pdf*

Questions:

1. (1 point) List your project GroupName and all group members names.

2. Movie Hits

   For this problem you will use a data set of 7,398 movie titles. This dataset of the movies and the associated properties has been collected from The Movie Database (TMDB) - https://www.themoviedb.org/. You will use this data set with the goal to **predict whether a movie is a hit** (that is, it is profitable with revenue exceeding five times the budget).

   The data set variables consist of the following:

   - `original_title` and `imdb_id` - variables that are to be ignored for prediction, but here to recognize the individual samples
   - `in_collection` - a binary feature that is `1` when a movie is part of a collection, is a sequel or has a sequel, and `0` when a movie is not part of a collection
   - `Action`, `Adventure`, ..., `Western` - a series of binary features encoding the genre of a movie (note, a movie may be categorized by more than one genre).
   - `has_homepage` - a binary feature on whether a movie has a dedicated webpage for the film
   - `is_en_orig_lang` - a binary feature indicating whether English is the original language of the film
   - `Warner Bros.`, `Universal Pictures`, ..., `Miramax Flims` - a series of binary features encoding the production companies associated with a movie (note, a movie can have several production companies)
   - `number_companies` - values representing the number of production companies associated with a film
   - `United State of America`, `United Kingdom`, ..., `Hong Kong` - a series of binary features encoding the production countries associated with a movie,

- `English`, `Francais`, ..., `Arabic` - a series of binary features encoding the spoken languages in a movie
- `cast_size`, `cast_female_count`, and `cast_male_count` - number of cast members listed and counts of the gender of the cast members
- `1920s`, `1930s`, ..., `2010s` - a series of binary features encoding the decade in which the film was released
- `Jan`, `Feb`, ..., `Dec` - a series of binary features encoding the month in which the film was released
- `Hit` - the label/class to be predicted, i.e., whether the movie was profitable making 5 times its budget.

(a) Load in the `hit-movies` data. You will not use the `original_title` and `imdb_id` variables for prediction.

(b) (5 points) Prepare the data for a 10-fold cross-validation design, using the *do-it-yourself* approach Ensure that each split of the data has a balanced distribution of class labels. Use min-max scaling to ensure all variables fall between values of $[0, 1]$.

Note: the next parts of the question do not require separate cells, in fact, the outer cross-validation loop should be run once and inside the loop each classifier can be learned. You should call out each type of learner using comments, e.g., # Q2c - Knn, followed by the Knn code, then # Q2d - Naive Bayes, followed by the NB code, etc.

(c) (9 points) Use kNN to predict whether a movie is a hit. Estimate the generalization performance over the folds, report the mean accuracy, F1-measure, and AUC on the testing data for k values of 3, 9, and 15.

(d) (6 points) Use decision trees to predict whether a movie is a hit. Estimate the generalization performance over the folds, report the mean accuracy, F1-measure, and AUC on the testing data. Show the results for two different sized trees (consider different amounts of pruning).
Because of the imbalanced data, in order to get trees beyond decision stumps weight the data based on the proportion of classes, e.g., weights of positive cases = 6, weights of negative cases = 1. Helpful functions: R - `weights` parameter of `rpart` function, Python - `class_weight` parameter of `DecisionTreeClassifier`

(e) (4 points) Use a Naive Bayes classifier to predict whether a movie is a hit. Report the mean accuracy, F1-measure, and AUC on the testing data over the folds.

(f) Perform a second layer of cross-validation ($k = 5$), an inner loop, to estimate the parameters of the following classifiers. The inner loop of the cross-validation can make use of the methods of grid search to select the best parameterization of the following classifiers. Or, you may elect to use the *do-it-yourself* approach with a nested loop.
Functions: Python - `GridSearchCV` from `sklearn.model_selection`, R - `trainControl` from `caret` or `GridSearchCV` from `superml`

i. (24 points) Learn support vector machine (SVM) models to predict whether a movie is a hit. You will consider multiple classifiers using both the RBF kernel (with default values) and polynomial kernel with degree 2, 3, and 4. Consider values for cost penalty parameter of $\{0.01, 0.1, 1\}$. Report the best parameter values (kernel + cost) for each outer fold (selected by AUC).
Functions: Matlab - `fitcsvm`, Python - `SVC` from `sklearn.svm`, R - `svm` from `e1071`, or use of `kernlab`

ii. (18 points) Use Random Forests to predict whether a movie is a hit. Consider multiple random forests with the number of trees in the forest to be $\{25, 50, 100\}$ and the

maximum number of features to be $\{6, 10, 14\}$. Report the best parameter values (number of trees + features) for each outer fold (selected by AUC).
Functions: Matlab - `TreeBagger`, Python - `RandomForestClassifier` from `sklearn.ensemble`, R - `randomForest` from `randomForest`

iii. (8 points) Use AdaBoost to predict whether a movie is a hit. Consider boosting methods with the number of decision stumps of $\{25, 50\}$. Report the best parameter values (number of stumps) for each outer fold (selected by AUC).
Functions: Matlab - `fitcensemble`, Python - `AdaBoostClassifier` from `sklearn.ensemble`, R - `boosting` from `adabag`

For the best parameters of each classifier, estimate the generalization performance over the folds, report the mean accuracy, F1-measure, and AUC on the testing data.

The performance results can be reported as a table/matrix with rows consisting of classifier type: KNN3, KNN9, KNN15, DT1, DT2, NB, best SVM, best RF, best AdaBoost, and the columns report the performance metrics: accuracy, F1-measure, and AUC.