College Football Spread Prediction ML Project

* Define the problem
  + What is the problem?
    - Want to accurately predict where or not a CFB team will cover the spread in the 2018 season
* Gather data
  + Write scripts to collect spreads, scores, totals, conferences, rankings, ratings from the 2015-2016 season, 2016-2017 season, 2017-2018 season
    - Spreads, scores, and totals from SportBookReviewOnline
      * If not working can use DonBest or ScoresAndOdds
    - Conferences and ratings from Sports-reference
      * Collect all offense and defense ratings
        + Can reduce further down the line
    - More ratings from Massey
* Prepare data
  + Data preprocessing
    - Format data into a single dataframe
      * Decide which features to combine, which instances to delete, etc.
    - Clean data so that it is uniform and consistent
    - Fill in blank data points
    - Create new columns
      * a team covers the spread, total
      * a team wins
      * month the game is played
      * bowl game or the regular season
  + Preliminary data exploration (do before transforming data)
  + Data transformation
    - Change categorical to numeral data
    - Square and cube data
    - Apply square roots
  + Exploration and summarization of data
    - Create univariate plots of each attribute
    - Create bivariate plots of each attribute with every other attribute
    - Create bivariate plots of each attribute with the class variable
  + Split data into train/test/validate
    - Standardize data
    - Normalize data
    - Use PCA to reduce features
      * Visualize k=2 and k=3
      * Save non-transformed data to test models as well
    - Apply mapping from standardized, normalized, and PCA to test and validation sets
* Spot check algorithms
  + Evaluate candidate algorithms
    - Select diverse set of algorithms
      * For this test:
        + KNN, Learning Vector Quantization, Naïve Bayes, Logistic Regression, Linear Discriminant Analysis, CART Decision Trees, C4.5/C5.0 Decision Trees, Back Propagation MLP Classifier, SVMs, Random Forest, Gradient Boost Machines
* Train the models
* Evaluate results
  + Create visualizations for each of the models
  + Use confidence interval, confusion matrix, accuracy logs, gain and lift charts, Kolmogorov-Smirnov charts, chi squared chart, ROC curves, gini coefficients, RMSE, cross validations, predictive power
* Algorithm tuning
  + Use historically effective model parameters
    - Search the space of model parameters
    - Optimize well performing parameters
  + Ensemble methods
    - Use bagging on well performing models
    - Use boosting on well performing models
    - Blend the results of well performing models
  + Model selection
    - Select (5-10) best performing models or model configurations
    - Evaluate well performing models on a holdout validation set
    - Select a small pool (1-3) of well performing models
* Predict 2018-2019 games once we have data
  + Create a script that will output predictions with probabilities for each model
    - Can piggy back off that in a variety of ways such as highest confidence, most model predictions, etc.
  + Create calculations for how much money will be made
    - Can output a win as +$9 and a loss as -$10
    - Can output a win as p\*$10 etc.