Machine Learning for Cities - Homework assignment 2
(Ravi Shroff, 4/18/2016)
DUE on 5/3/2016 by 11:59pm EST

This homework assignment will involve analyzing a dataset of *all* recorded NYPD "Terry" police stops in New York City in the years 2011-2012, as the (compressed) file 'hw_2_stops.csv.zip'. Note that in contrast to the previous assignment and the in-class examples with this dataset, I have provided you with a few additional features and I have not removed missing data. Make sure you only fit models that make sense (e.g. if you are trying to predict arrest, don't use found.weapon as a feature). Also, make sure all plots have titles and axis labels. **You may use any language/software package of your choice to complete this assignment. You may use and modify code from the in-class iPython notebooks as you like. You may turn in an iPython notebook or a typed report (with plots).**

1) Fit a **random forest** to the data and make some plots:
    a) Choose a test/train split (some natural choices: train on 2011, test on 2012, or train on 75% of the data, test on 25% of the data).
    b) Choose a target variable (some natural choices: found.weapon, found.gun, arrested, frisked, searched, summons.issued, found.contraband, any of the "use of force" variables (e.g. force.baton or force.handcuffs, or you can create an aggregate of all the force variables).
    c) [Optional] restrict to a subset of the data. For example, if your outcome measure is found.weapon or found.gun, it may make sense to restrict to stops with suspected.crime=='cpw', or if your outcome measure is found.contraband, it may make sense to restrict to stops with a suspected crime involving criminal sale/possession of "marihuana" and criminal sale/possession of a controlled substance.
    d) Fit the model (with at least 1000 trees), predict *probabilities* for each stop in the test set, and do the following:
        i) Select two categorical features, and make two corresponding plots comparing the average model prediction and empirical outcome for each value of that feature. For example, if a chosen feature is 'precinct', the plot should have a point for each precinct, where the x-value is the average model prediction for all stops in that precinct, and the y-value is the average value of your target variable for all stops in that precinct.
        ii) Write (at least) one paragraph explaining your work in parts a)-d). Make sure to explain your target variable, predictors, how you chose your test/train split, and any other choices you made.
2) Fit a **decision tree** to the data and make a plot:
    a) Choose a test/train split.
    b) Choose a target variable (you may choose the same target as in question 1) or a different one).

c) [Optional] restrict to a subset of the data, as in question 1)
d) For the decision tree package of your choice (e.g. http://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html), choose a parameter governing *model simplicity*. For example, the maximum tree depth or maximum number of leaf nodes. Then, fit your decision tree classifier for different values of this parameter and for each such value, record the corresponding AUC score.
    i) **Make a plot of performance vs. simplicity for different values of the parameter chosen in part d). That is, the x-axis should be parameter value (e.g. tree depth) and the y-axis should be AUC score.**
    ii) **Visualize a simple decision tree (e.g. a "shallow" tree, or a tree with few leaf nodes) classifier and report its performance. You can draw the decision tree by hand or use a graphical representation (e.g. http://scikit-learn.org/stable/modules/generated/sklearn.tree.export_graphviz.html), but make sure it is easy to understand (e.g. the features chosen for each split should be clearly labeled in each internal node, as well as the prediction at each leaf node).**
    iii) **Write (at least) one paragraph explaining your work in parts a)-d). Make sure to explain your target variable, predictors, how you chose your test/train split, and any other choices you made.**