Assignment 2: Decision Tree Classification and Linear Regression

For due date see `https://courses.cs.sfu.ca`

This assignment is to be done individually.

Important Note: The university policy on academic dishonesty (cheating) will be taken very seriously in this course. You may not provide or use any solution, in whole or in part, to or by another student. Any mark in this assignment may be changed on the basis of an oral exam where you need to explain your solution.

You are encouraged to discuss the concepts involved in the questions with other students. If you are in doubt as to what constitutes acceptable discussion, please ask! Further, please take advantage of office hours offered by the instructor and the TA if you are having difficulties with this assignment.

DO NOT:

- Give/receive code or proofs to/from other students
- Use web search to find solutions for assignment

DO:

- Meet with other students to discuss assignment (it is best not to take any notes during such meetings, and to re-work assignment on your own)
- Use online resources (e.g. Wikipedia) to understand the concepts needed to solve the assignment
- Make use of the sample code that we provide on the website. You may have to adapt the code to work on your system. You should be able to explain any piece of code that you submit.

Data Set: The course website specifies the data set, including training data, test data, and preprocessing steps. In addition, different parts of the assignment may require further data preprocessing as specified. The target variables will also be specified for each component.

Theory: Variance and Covariance (10 Marks)

For definitions and notation please refer to the text. We write $var(X)$ for the variance of a single random variable and $cov(X, Y)$ for the covariance of two random variables, such that $var(X) = cov(X, X)$.

1. Show that $var(X) = E(X^2) - [E(X)]^2$.
2. Show that if two random variables X and Y are independent, then their covariance is

zero.

Practice: Decision Tree Learning (15 Marks)

1. Install a package that implements the ID3 decision tree algorithm that we studied in class, for both discrete and continuous input features. We recommend using Weka, see course web page.
2. Apply the ID3 learner to the hockey draft dataset, using *GP > 0* as the target class variable. For data preprocessing, drop the *sum_7yr_GP* column.
3. Show the decision tree learned. Which branch is the most informative, meaning that its leaf has the lowest class entropy? Given your understanding of the domain, do the features on the branch make sense?
4. (Bonus Question) Rerun the Naive Bayes classifier from assignment 1 on the new training and test set for this assignment. Compare the test set accuracy of the decision tree learner to the result of the Naive Bayes classifier.

Theory: Minimum Least Squares Error for Regularized Linear Regression (20 Marks)

Consider least-squares linear regression with L2 regularization as defined in the text.

1. Using the notation of the text, write down the squared-error function, including the regularization term.
2. Show that the weight vector $\boldsymbol{w}^*$ that minimizes this error function is given by $\boldsymbol{w}^* = \left(\lambda \boldsymbol{I} + \boldsymbol{X}^T\boldsymbol{X}\right)^{-1}\boldsymbol{X}^T\boldsymbol{y}$.

Practice: Implement Least Squares Regression (50 Marks)

We will gain practice with linear regression by applying it to predict the number of NHL games that a player will have played after 7 years. So the independent target variable will be *sum_7yr_GP*. We will go through a few typical steps for a regression analysis: data preprocessing, weight learning, and model evaluation.

We can increase the power of linear regression by adding non-linear terms as new derived columns. The functions that give rise to the new columns are called *basis functions*, and the new data matrix that includes the basis functions is called the *design matrix*. A common type of non-linear term to add are products of the original features that combine information from different columns; these are called *interaction terms*.

1. **Data Preprocessing**. In addition to what is specified on the website, apply the following preprocessing steps.

(a) Drop the *GP > 0* column.

(b) The hockey data form a *hybrid* data set, meaning that it mixes discrete with continuous variables. Regression is a technique for continuous input variables. A common approach to using regression with discrete variables is to first convert all discrete variables to binary (Boolean) variables, then use 1 and 0 to represent variables. Boolean variables that are treated as continuous regressors are called "dummy" variables or indicator variables. So the first step is to replace all discrete variables by dummy variables. For example, instead of having a single variable *Country_Group* with three possible values *Canada, USA, Europe*, you should introduce three binary variables *Country_Group = Canada, Country_Group = USA, Country_Group = Europe*. Change the data matrix so all values for the dummy variables are 1 or 0, depending on where the player is from.

(c) **Add quadratic interaction terms.** For each pair of features $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$, in the data matrix (with specified columns dropped), add their product $\boldsymbol{x}_i \cdot \boldsymbol{x}_j$ as a new column. So if the data matrix contains d columns, the expanded *design matrix* $\boldsymbol{D}$ should include $d + \binom{d}{2}$ columns.

(d) **Standardize Predictors.** As discussed in class, in a multivariate regression it is often a good idea to standardize continuous variables to the same scale. There are different methods for doing this. One common way is to change the values of a column in the data matrix like this:

i. Subtract the mean of the column from each entry (**centering**).

ii. Divide each entry by the standard deviation of the column.

Standardize the columns for continuous variables as described (leave alone the discrete columns with dummy variables).

2. **Evaluating a weight vector**. Write code that takes as input a weight vector and outputs the squared-error loss (see text) that results from predicting *sum_7yr_GP* using the weight vector.

3. **Finding a weight vector**. Write code that takes as input a regularization parameter λ and outputs (i) the optimal weight vector as defined in the previous theory exercise (ii) the squared-error loss for this weight vector.

Now you have working code to perform linear regression. The main issue is finding a good value for the regularization parameter λ. Let us try an exponential grid search, as follows.

1. Try the values from the set $\lambda = 0, 0.01, 0.1, 1, 10, 100, 1000$. Make a plot that shows the following. The horizontal axis shows the value of λ on a log scale (this is called a `semilogx` plot). One curve in the plot should show the squared-error loss evaluated by using 10-fold *cross-validation* on the training set. The second curve shows the squared-error loss evaluated by applying the learned weight vector to the *test set.* Put this plot in your report, and note which regularizer value you would choose from the cross-validation, and which regularizer value would give the lowest squared-error on the test set.

2. For the regularizer that you chose as best from cross-validation, inspect the learned weight magnitudes. Are any of the quadratic interaction terms important (i.e. carry significant weight compared to other variables)? The decision tree also captures interactions among predictor variables - how do the decision tree interactions compare to the interaction terms with high weights?

Submitting Your Assignment

You should create a report with the answers to questions and figures described above in PDF format. Make sure it is clear what is shown in each figure. We would also like you to submit your source code.

Submit your assignment using the online assignment submission server at: `https://courses.cs.sfu.ca`.