DS5220: Supervised Machine Learning and Learning Theory Fall 2022

Problem Set 3

Instructor: Hongyang Ryan Zhang *Due:* ***November 4, 2022, 11:59pm***

Instructions:

- You are expected to write up the solution on your own. Discussions and collaborations are encouraged; remember to mention any fellow students you discussed with when you turn in the solution.
- There are up to three late days for all the problem sets and project submissions. Use them wisely. After that, the grade depreciates by 20% for every extra day. Late submissions are considered case by case. Please reach out to the instructor if you cannot meet the deadline.
- Submit your written solutions to Gradescope and upload your code to Canvas. You are recommended to write up the solution in LaTeX.

Problem 1 (20 points) We will now perform cross-validation on a simulated data set.

```
numpy.random.seed(123)
x = numpy.random.normal(0, 1, (200))
y = x + 2 * x**2 - 2 * x**3 + numpy.random.normal(0, 1, (200))
```

(a) **[7 points]** Perform best subset selection in order to choose the best model containing the polynomial features up to degree 10: $X, X^2, \cdots, X^{10}$. What is the best model obtained according to C_p (AIC), BIC, and adjusted R^2? Show some plots to provide evidence for your answer, and report the coefficients of the best model obtained. [Hint: Write a recursion to enumerate over all possible subsets of $\{X, X^2, \ldots, X^{10}\}$.]

(b) **[7 points]** Perform subset selection using forward stepwise selection. How does your answer compare to the results in (a)? [Hint: Write a (double) for loop to implement the forward stepwise rule.]

(c) **[6 points]** Fit a linear regression with lasso regularization model to the simulated data set, again using $X, X^2, \cdots, X^{10}$ as the predictors. Use cross-validation to select the optimal value of λ. Create plots of the cross-validation error as a function of λ. Report the coefficient estimates using the optimal λ on the entire data, and discuss the results obtained. [Hint: You may find `sklearn.linear_model.Lasso` useful.]

Problem 2 (25 points) This question is based on the `Medical Insurance Cost` data set. This data set has statistics about the medical insurance charges based on a set of factors such as region, number of children and other features. You can find the description of this data set at here. We will now try to predict the medical insurance charges using all variables.

(a) **[2 points]** Split the data into a training set and a test set with 80% observations randomly assigned to the training set and the rest 20% observations assigned to the test set.

(b) **[3 points]** Fit a linear model using least squares on the training set, and report the test error obtained.

(c) **[5 points]** Fit a ridge regression model on the training set, with λ chosen by cross-validation. Report the test error obtained. [Hint: You may find `sklearn.linear_model.Ridge` useful.]

(d) **[5 points]** Fit a lasso regression model on the training set, with λ chosen by cross-validation. Report the test error obtained, along with the number of non-zero coefficient estimates. [Hint: You may find `sklearn.linear_model.Lasso()`.]

(e) **[5 points]** Fit a principal component regression model on the training set, with M chosen by cross-validation. Report the test error obtained, along with the value of M selected by cross-validation. [Hint: First apply `sklearn.decomposition.PCA` to the data set, then fit a linear regression model.]

(f) **[5 points]** Fit a partial least squares model on the training set, with M chosen by cross-validation. Report the test error obtained, along with the value of M selected by cross-validation. [Hint: You may find `sklearn.cross_decomposition.PLSRegression` useful.]

Problem 3 (25 points) This question is based on the College data set. This data set has statistics for a large number of US Colleges from the 1995 issue of US News and World Report. You can find the description of this data set at `https://rdrr.io/cran/ISLR/man/College.html`. Let us first create a variable of acceptance rate, Accept.Rate, that is the number of applications accepted (Accept) divided by the number of applications received (Apps). We will now try to predict the acceptance rate using all variables other than Accept and Apps. We can remove Accept and Apps from the data frame. The data set can be downloaded here.

(a) **[0 points]** Split the data into a training set and a test set with 80% observations in the training set and 20% observations in the test set.

(b) **[5 points]** Fit a regression tree to the training set. Plot the tree, and interpret the results. What test MSE do you obtain? (Hint: You may find `DecisionTreeRegressor()` in the `sklearn.tree` library, `plot_tree()` in `sklearn.tree()` useful.)

(c) **[10 points]** Use cross-validation in order to determine the optimal depth of tree `max_depth`. Similarly, select the minimum number of samples for a split `min_samples_split` and for a leaf `min_samples_leaf` using cross-validation. Does selecting these parameters in the regression tree improve the test MSE? Plot the tree again and compare it to the plot from Step (b).

(d) **[5 points]** Use the bagging approach in order to analyze this data. What test MSE do you obtain? Use the `feature_importances_()` attribute in `DecisionTreeRegressor()` to identify which variables are most important. (Hint: You may find `BaggingRegressor()` in `sklearn.ensemble` library useful.)

(e) **[5 points]** Use random forests to analyze this data. What test MSE do you obtain? Use the `importance()` function to determine which variables are most important. Describe the effect of number of trees m as well as the number of variables considered at each split, on the error rate obtained. (Hint: You may find `RandomForestRegressor()` in the `sklearn.ensemble` library useful. You may use the `n_estimators` parameter to change the number of trees. You may use the `max_features` parameter to change the number of features considered at each split.)

Problem 4 (20 points) In this problem, you will implement the random forest algorithm. The data set for this assignment is the Palmer Archipelago (Antarctica) penguin data from Kaggle repository available at here. Split the original data into 80% for training and 20% for testing (chosen at random).

(a) **[5 points]** Begin by creating a bootstrap sample of size 1000 and then select a subset of p columns. Vary the value of p and report the p that results in the lowest cross-validation error. Now train a decision tree classifier on the bootstrap sample by setting the depth to 6. (Hint: You may want to use `DecisionTreeClassifier()` in `sklearn.tree()` to fit each tree. Refer to class notes and some suggested values to choose the value of p.)

(b) **[5 points]** Repeat the above step to generate $T \in \{1, 50, 100, 150, 200, 300, 400\}$ trees and evaluate on your training set. Combine the predictions from all trees and assign the final class based on a majority vote of the predictions of every tree. In case of ties, assign a class randomly among the ties.

(c) **[5 points]** Report the training and test error, F1 score, and AUC by varying T in the range $\{1, 50, 100, 150, 200, 300, 400\}$.

(d) **[5 points]** Use an existing package to train a Random Forest algorithm with 10, 50, and 100 decision trees. Report similar metrics on both the training and testing sets. Report the top 10 features having the most influence on the model. (Hint: You may find `RandomForestClassifier()` in the `sklearn.ensemble` library useful.)

Problem 5 (10 points) In this problem, we will look inside the neural network architecture for a multi-layer linear neural net and show that it collapses to a single linear layer. Consider a feed forward neural network with one hidden layer as shown below. A linear activation function $\sigma(z) = cz$ is used at the hidden nodes while the output node uses the sigmoid activation function $\sigma'(z) = \dfrac{1}{1+e^{-z}}$ to learn the function for $P(y = 1|x, w)$ where $x = (x_1, x_2)$ and $w = (w_1, w_2, \ldots, w_9)$. See Figure 1 for an illustration.

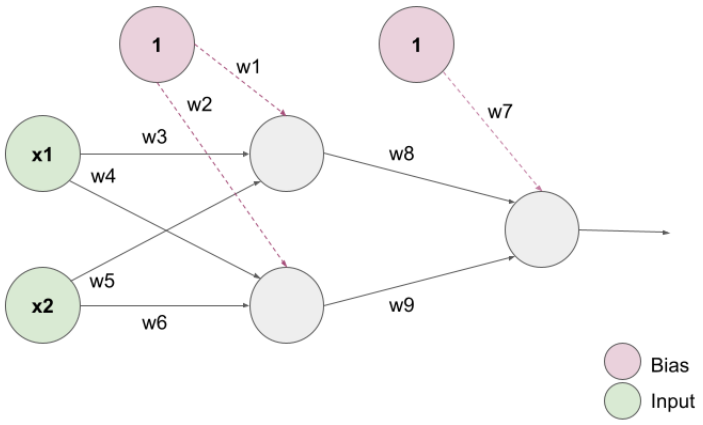

Figure 1: Illustration for the network architecture in Problem 5.

(a) **[5 points]** What is the output $P(y = 1|x, w)$ from the above neural net? Express it in terms of x_i, c and weights w_i. What kind of final classification boundary does this yield?

(b) **[5 points]** Draw a neural net without any hidden nodes such that its output is equivalent to that of the given neural net. Write the weights $\tilde{w}$ of this new neural net in terms of c and w_i.

Is it true that any multi-layered neural network with linear activation functions at hidden layers can be represented as a neural net without any hidden layer? Explain your answer.