DS5220: Supervised Machine Learning and Learning Theory Fall 2021

Problem Set 2

Instructor: Hongyang R. Zhang *Due:* ***October 27, 2021, 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.
- All homework submissions are subject to the Northeastern University Honor Code.

Problem 1 (20 points) In this problem, you will develop a model to predict whether a given car gets high or low gas mileage based on the `Auto` data set. The `Auto` data set has gas miles per gallon (mpg), horsepower, and other information for several hundred cars. You can find the description of this data set at `https://rdrr.io/cran/ISLR/man/Auto.html`.[1] Note: Some cars are missing the value in `horsepower`; Remove those data points from the data set.

(a) (3 points) Create a binary variable, `mpg01`, that contains a 1 if `mpg` contains a value above its median, and a 0 if `mpg` contains a value below its median. [Hint: You could compute the median using the numpy.median() function and add the `mpg01` column to the data set.]

(b) (3 points) Explore the data graphically in order to investigate the association between `mpg01` and the other features. Which of the other features seem most likely to be useful in predicting `mpg01`? Scatterplots and boxplots may be useful tools to answer this question. Describe your findings. [Hint: You may find `matplotlib.pyplot` helpful.]

(c) (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.

[1] The data set can be downloaded here: `https://www.kaggle.com/ishaanv/ISLR-Auto?select=Auto.csv`.

(d) (3 points) Perform logistic regression on the training data in order to predict `mpg01` using `cylinders`, `weight`, `displacement`, and `horsepower`. What is the test error of the model obtained? [Hint: You may find `sklearn.linear_model.LogisticRegression`, and the functions `fit()` and `predict()` helpful.]

(e) (3 points) Perform LDA on the training data in order to predict `mpg01` using `cylinders`, `weight`, `displacement`, and `horsepower`. What is the test error of the model obtained? [Hint: You may find `sklearn.discriminant_analysis.LinearDiscriminantAnalysis` helpful.]

(f) (3 points) Perform QDA on the training data in order to predict `mpg01` using `cylinders`, `weight`, `displacement`, and `horsepower`. What is the test error of the model obtained? [Hint: You may find `sklearn.discriminant_analysis.QuadraticDiscriminantAnalysis`.]

(g) (3 points) Perform KNN on the training data, with several values of K, in order to predict `mpg01` using `cylinders`, `weight`, `displacement`, and `horsepower`. Report the test errors you observe. Which value of K performs the best for this data set? [Hint: You may find `sklearn.neighbors.KNeighborsClassifier` helpful.]

Problem 2 (10 points) In this problem, we will consider the bootstrap sampling. We will derive the probability that a given data point is part of a bootstrap sampled set. Suppose that we obtain a bootstrap sampled set from a (training data) set of n observations: $x_1, x_2, \ldots, x_n$.

(a) (2 points) Let z_1 be the first bootstrap sample. What is the probability that $z_1 \neq x_1$?

(b) (2 points) Let z_2 be the second bootstrap sample. What is the probability that $z_2 \neq x_1$?

(c) (2 points) For any $n = 1, 2, \ldots$, let z_n be the n-th bootstrap sample. Let $S = \{z_1, z_2, \ldots, z_n\}$ be the set of bootstrap samples. When $n = 100$, what is the probability that x_1 is in S?

(d) (4 points) For an arbitrary n, what is the probability that $x_1 \in S$? Based on this probability, what is the expected number of distinct data points in the set S?

Problem 3 (25 points) This question is based on the `Boston` housing data set. This data set has the information about housing values in 506 suburbs of Boston. You can find the description of this data set at `https://www.cs.toronto.edu/~delve/data/boston/bostonDetail.html`.[2]

(a) (2 points) Based on this data set, provide an estimate for the population mean of `crim` (the crime rate by town). Let's call this estimate $\hat{\mu}$. [Hint: You may find `numpy.mean()` helpful.]

[2] The data set can be downloaded from here: `http://lib.stat.cmu.edu/datasets/boston`.

(b) (2 points) Provide an estimate of the standard error of $\hat{\mu}$. Interpret this result. [Hint: You can compute the standard error of the sample mean by dividing the sample standard deviation by the square root of the number of observations. You may find `numpy.std()` helpful.]

(c) (5 points) Now estimate the standard error of $\hat{\mu}$ using 1,000 bootstrap sampled sets. Let $\hat{\mu}_1, \hat{\mu}_2, \ldots, \hat{\mu}_{1000}$ be the estimated mean from the 1,000 bootstrap sampled sets. Estimate the standard error of $\hat{\mu}$ using these 1,000 values. How does this compare to your answer from (b)? [Hint: You may find `sklearn.utils.resample` helpful. The standard error of $\hat{\mu}$ is the standard deviation of the 1,000 estimated means $\{\hat{\mu}_1, \hat{\mu}_2, \ldots, \hat{\mu}_{1000}\}$ from all the bootstrap sampled sets.]

(d) (4 points) Based on your bootstrap estimate of the standard error from (c), provide a 95% confidence interval for the mean of `crim`. [Hint: You can approximate a 95% confidence interval using the formula $[\hat{\mu} - 2 \cdot \mathrm{se}(\hat{\mu}), \hat{\mu} + 2 \cdot \mathrm{se}(\hat{\mu})]$.]

Then, compare it to the results obtained using `scipy.stats.norm.interval()` (applied to `crim`).

(e) (2 points) Based on this data set, provide an estimate for the first 25% quantile of `crim`. Let's call this quantity $\hat{\mu}_{0.25}$ [You may find `numpy.quantile()` useful.]

(f) (5 points) We would like to estimate the standard error of $\hat{\mu}_{0.25}$. While there is no simple formula to compute the standard error of $\hat{\mu}_{0.25}$, proceed by estimating the standard error of the median using the bootstrap. Compare the standard error to the value of $\hat{\mu}_{0.25}$. Then, comment on your findings. [Hint: Follow the steps in step (c).]

(g) (5 points) Consider a linear regression model to predict `crim` using `rad` (index of accessibility to radial highways). Compute estimates for the standard errors of the intercept β_0 and coefficient β_1 of `rad` in two different ways: (1) using the bootstrap, and (2) using the standard errors provided in the `scipy.stats.linregress()` function. Comment on your findings.

Problem 4 (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 5 (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`.[3] 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.

(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.]

[3] The data set can be downloaded here: `https://www.kaggle.com/ishaanv/ISLR-Auto?select=College.csv`.

(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.]