1 Random forrest for email spam classifier (30 points)

Your task for this question is to build a spam classifier using the UCR email spma dataset `https://archive.ics.uci.edu/ml/datasets/Spambase` came from the postmaster and individuals who had filed spam. The collection of non-spam e-mails came from filed work and personal e-mails, and hence the word 'george' and the area code '650' are indicators of non-spam. These are useful when constructing a personalized spam filter.

One would either have to blind such non-spam indicators or get a very wide collection of non-spam to generate a general purpose spam filter. Load the data.

1. (5 points) How many instances of spam versus regular emails are there in the data? How many data points there are? How many features there are?

   Note: there may be some missing values, you can just fill in zero.

Answer:

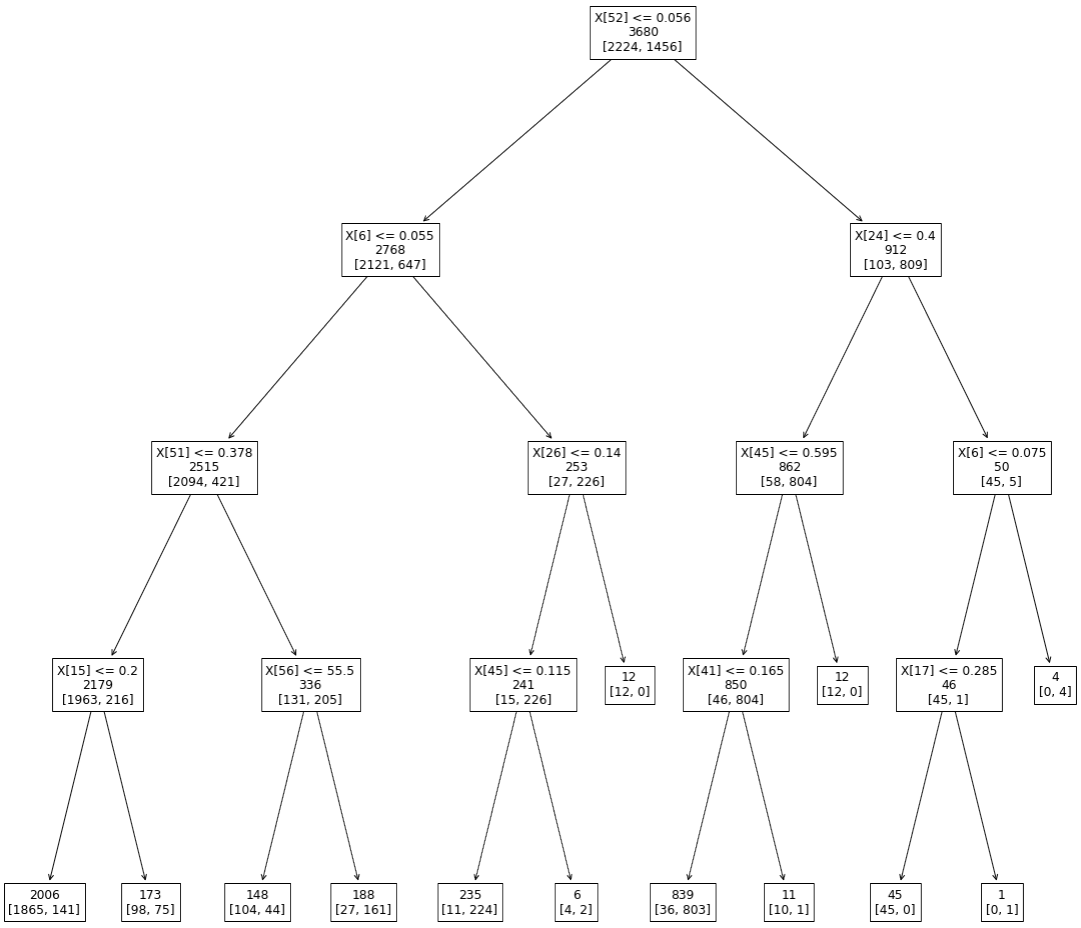
There are 1813 spam emails versus 2788 regular emails, 4601 data points in total and 58 features.

2. (10 points) Build a classification tree model (also known as the CART model). In Python, this can be done using sklearn.tree.DecisionTreeClassifier. In our answer, you should report the tree models fitted similar to what is shown in the "Random forest" lecture, Page 16, the tree plot. In Python, getting this plot can be done using sklearn.tree.plot_tree function.

Answer:

After splitting data into 80% training data and 20% testing data, I fitted training data in a classification trees model. When max depth was not limited, the tree was big, and plot was hard to read. I tried different max depth and decided to use max depth equal to 4 to have both a good tree plot and a good accuracy (threshold = 0.5).

Max depth	2	4	6	8	Unlimited
Model accuracy	88.0%	91.0%	92.0%	92.0%	91.0%



3. (15 points) Also build a random forrest model. In Python, this can be done using sklearn.ensemble.RandomForestClassifier.

   Now partition the data to use the first 80% for training and the remaining 20% for testing. Your task is to compare and report the AUC for your classification tree and random forest models on testing data, respectively. To report your results, please try different tree sizes. Plot the curve of AUC versus Tree Size, similar to Page 15 of the Lecture Slides on "Random Forest".

   Background information: In classification problem, we use AUC (Area Under The Curve) as a performance measure. It is one of the most important evaluation metrics for checking any classification model?s performance. ROC (Receiver Operating Characteristics) curve measures classification accuracy at various thresholds settings. AUC measures the total area under the ROC curve. Higher the AUC, better the model is at distinguishing the two classes. If you want to read a bit more about AUC curve, check out this link `https://towardsdatascience.com/understanding-auc-roc-curve-68b2303cc9c5` For instance, in Python, this can be done using sklearn.metrics.roc_auc_score and you will have to figure out the details.

Answer:

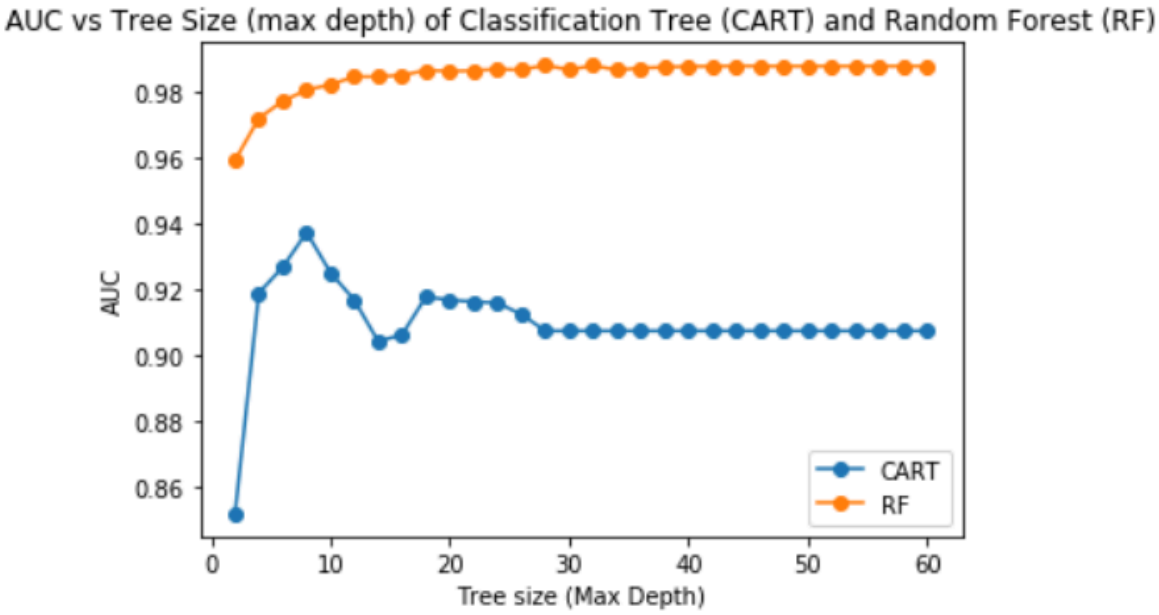
Using the same training data and test data from the above, random forest was fitted (tree number = 100) and accuracy was 96.0% (threshold = 0.5). Because accuracy could change depending on different thresholds, it's better to use AUC to compare different models.

From the picture below, I summarize the results as follows:

1. Random forest AUC increases as tree size increases and then stays the same. AUC curve becomes flat when max depth is over 12, and AUC converges at the value of 0.9878.

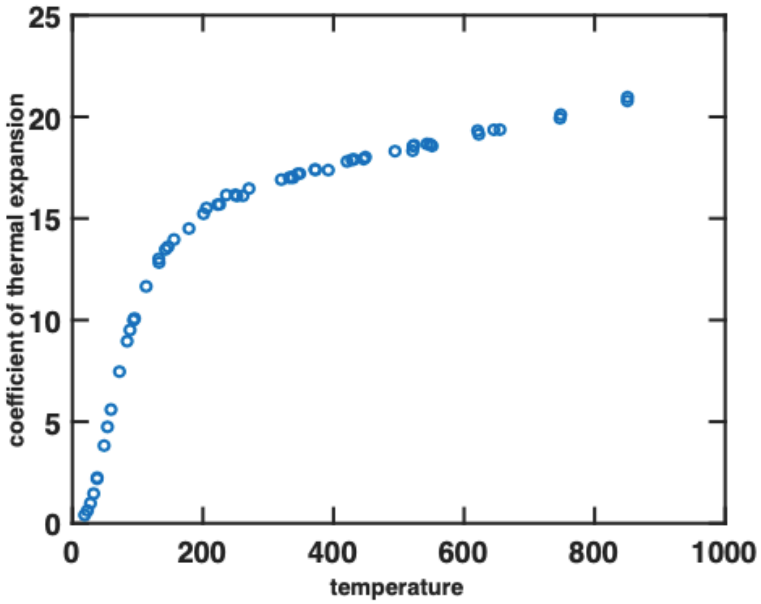
2. As tree size increases, CART AUC increases, decreases, increases, decreases and then stays the same. CART shows the highest AUC when max depth is 8 and becomes overfitted when max depth is over 8. AUC curve becomes flat when max depth is over 28, and AUC converges at the value of 0.9073.

3. Random forest shows a better performance than CART no matter what max depth is chosen.



2 Nonlinear regression and cross-validation (30 points)

The coefficient of thermal expansion y changes with temperature x. An experiment to relate y to x was done. Temperature was measured in degrees Kelvin. (The Kelvin temperature is the Celcius temperature plus 273.15). The raw data file is copper-new.txt.



1. (10 points) Perform linear regression on the data. Report the fitted model and the fitting error.

Answer:

Using the whole dataset to fit the model, I got the fitted model as:

$$y = 0.02128314x + 7.38412739$$

The mean squared error based on the whole dataset was 12.6052.

(I also fitted the linear regression model on training data as follows:

After splitting data into 80% training data and 20% testing data, I fitted training data in a linear regression model.

The fitted model has the form as:

$$y = 0.02169922x + 7.44064023$$

The mean squared error based on test data is 12.9869.)

2. (10 points) Perform nonlinear regression with polynomial regression function up to degree $n = 10$ and use ridge regression (see Lecture Slides for "Bias-Variance Tradeoff"). Write down your formulation and strategy for doing this, the form of the ridge regression.

Answer:

Strategy:

a. Split data into 80% training data and 20% testing data;

b. Transform training data and testing data to degree 1 to 10;

b. In each degree, perform ridge regression using alpha/lambda values in the range of $1x10^{-7}$ and 1.

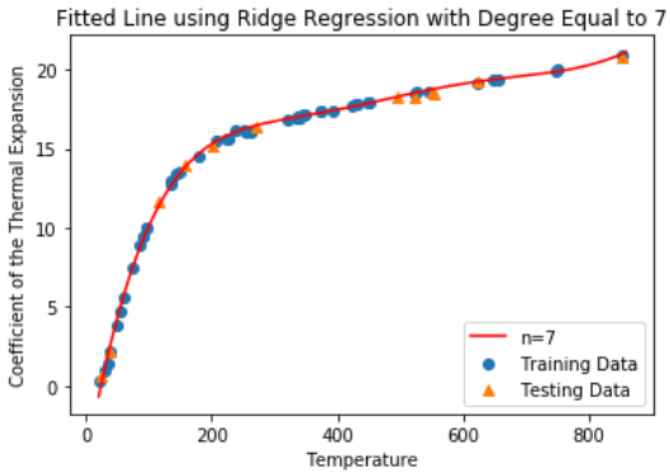
c. Find the minimum MSE value, and corresponding degree and alpha/lambda.

The MSE results are summarized as the following table.

	alpha = 1e-07	alpha = 1e-06	alpha = 1e-05	alpha = 0.0001	alpha = 0.001	alpha = 0.01	alpha = 0.1	alpha = 1
n = 1	12.986921	12.986917	12.986872	12.986421	12.981946	12.940448	12.797905	18.549388
n = 2	6.053214	6.053165	6.052685	6.047945	6.006842	6.036358	10.669938	17.825714
n = 3	1.432399	1.431693	1.424946	1.386166	2.165050	6.486727	9.500859	18.134711
n = 4	0.209639	0.211184	0.251931	0.936517	2.107008	5.457523	9.622696	18.171859
n = 5	0.067269	0.076455	0.294080	0.613599	2.123238	4.801282	9.858848	18.033499
n = 6	0.071654	0.074917	0.194059	0.646225	1.902647	4.670356	9.913851	17.853795
n = 7	0.065745	0.081531	0.139995	0.661760	1.727718	4.712986	9.842143	17.695252
n = 8	0.067009	0.074750	0.139557	0.605100	1.669262	4.754189	9.735746	17.575090
n = 9	0.072190	0.070965	0.145676	0.538546	1.675436	4.752623	9.644951	17.491131
n = 10	0.075334	0.074542	0.144303	0.497688	1.694644	4.717016	9.584486	17.435402

Minimum MSE is 0.065745 when n = 7 and alpha/lambda =$1x10^{-7}$, and the corresponding model formula and graph are as follows:

$$y = -4.84658185 + 0x^0 + 2.26915319x10^{-1}x^1 - 9.38197233x10^{-4}x^2 + 1.77300773x10^{-6}x^3 - 9.86069778x10^{-10}x^4 - 1.34916889x10^{-12}x^5 + 2.08095957x10^{-15}x^6 - 7.78443447x10^{-19}x^7$$



The ridge regression has the following form:

$$\theta^r = \text{argmin}_\theta L(\theta) = \frac{1}{m}\sum_{i=1}^{m}(y^i - \theta^T x^i)^2 + \lambda\|\theta\|^2$$

where x has m data points, $x = (1, x, x^2, \ldots, x^n)^T$

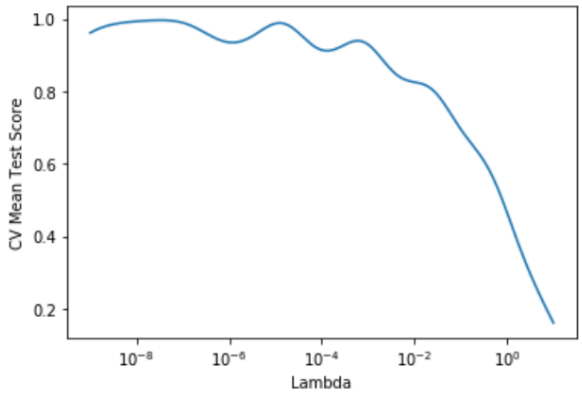
$\theta = (\theta_0, \theta_1, \theta_2, \ldots, \theta_n)^T$, λ is also ~~alpha~~ here alpha ~~values~~.

In this case, when MSE is minimized, $n=7$, λ = alpha = 1×10^{-7}.

3. (5 points) Use 5 fold cross validation to select the optimal regularization parameter λ. Plot the cross validation curve and report the optimal λ.

Answer:

Using GridSearchCV function and 5-fold cross validation, different lambda (200 values in the range between $1x10^{-9}$ and $1x10^{1}$) were tested to give the best lambda as 3.2176×10^{-8}. Cross validation curve is as follows:



4. (5 points) Predict the coefficient at 400 degree Kelvin using both models. Comment on how would you compare the accuracy of predictions.

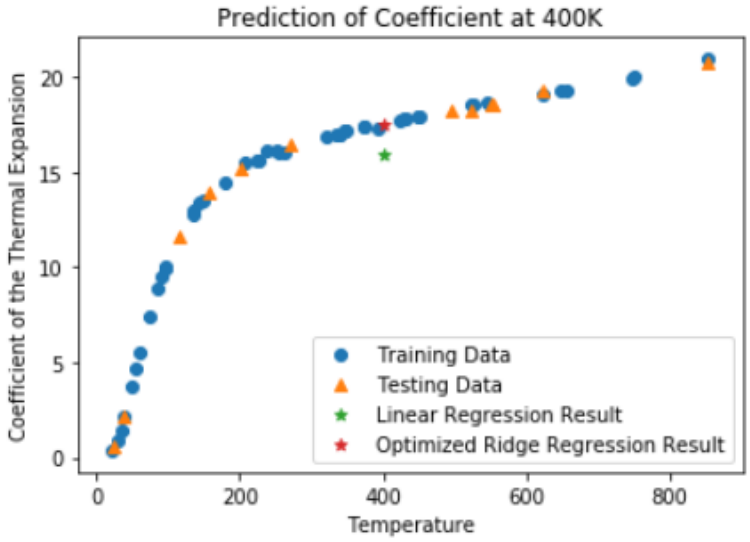
Using linear regression (trained on the whole data set), the predicted coefficient is 15.89738509.

Using optimized ridge regression obtained from part 3, the predicted coefficient is 17.4932931.

I would compare the prediction accuracy using two methods:

a. Quantitatively, compare MSE of two models. MSE of linear regression is 12.61, and MSE of optimized ridge regression is 0.06554. Smaller MSE is better and I prefer to rely on optimized ridge regression's prediction result.

b. Qualitatively, plot and compare results to original data. The plot is as follows, and the prediction result from optimized ridge regression fits the original data trend better.



3 Regression, bias-variance tradeoff (40 points)

Consider a dataset with n data points (x_i, y_i), $x_i \in \mathbb{R}^p$, drawn from the following linear model:

$$y = x^T\beta^* + \epsilon,$$

where ϵ is a Gaussian noise and the star sign is used to differentiate the true parameter from the estimators that will be introduced later. Consider the regularized linear regression as follows:

$$\hat{\beta}(\lambda) = \arg\min_{\beta} \left\{ \frac{1}{n}\sum_{i=1}^{n}(y_i - x_i^T\beta)^2 + \lambda\|\beta\|_2^2 \right\},$$

where $\lambda \geq 0$ is the regularized parameter. Let $X \in \mathbb{R}^{n\times p}$ denote the matrix obtained by stacking x_i^T in each row.

1. (10 points) Find the closed form solution for $\hat{\beta}(\lambda)$ and its distribution.
2. (10 points) Calculate the bias $\mathbb{E}[x^T\hat{\beta}(\lambda)] - x^T\beta^*$ as a function of λ and some fixed test point x.
3. (10 points) Calculate the variance term $\mathbb{E}\left[\left(x^T\hat{\beta}(\lambda) - \mathbb{E}[x^T\hat{\beta}(\lambda)]\right)^2\right]$.
4. (10 points) Use the results from parts (b) and (c) and the bias-variance decomposition to analyze the impact of λ in the squared error. Specifically, which term dominates when λ is small, and large, respectively?

(Hint.) Properties of an affine transformation of a Gaussian random variable will be useful throughout this problem.

Prelequisite equation

if $X \sim N(\mu, \Sigma)$ ..... ①

$Y = AX + B$ .... ②

Then $Y \sim N(A\mu + B, A\Sigma A^T)$ .... ③

where X is a random vector, μ is the mean value vector and Σ is the covariance matrix of X.

Proof: $\bar{y} = E(y) = E(AX+B) = AE(X) + B = A\bar{x} + B$

$\Sigma_y = E\{(y-\bar{y})(y-\bar{y})^T\}$

$= E\{[(AX+B)-(A\bar{x}+B)][(AX+B)-(A\bar{x}+B)]^T\}$

$= E\{[A(x-\bar{x})][A(x-\bar{x})]^T\}$

$= E\{[A(x-\bar{x})](x-\bar{x})^T A^T\}$

$= AE\{(x-\bar{x})(x-\bar{x})^T\}A^T$

$= A\Sigma_x A^T$

Then y is defined as $y \sim N(A\mu + B, A\Sigma A^T)$ ($\mu = \bar{x}$ here)

Part 1. From the slides

$\hat{\beta} = (x^T x + \lambda I)^{-1} x^T y$ .... ④

~~$= (x^T x + \lambda I)^{-1} x^T x \beta^*$~~

Distribution: According to equation ①-③, y is Gaussian and variance is σ_ε^2

$E[\hat{\beta}] = (x^T x + \lambda I)^{-1} x^T x \beta^*$ .... ⑤

$Var[\hat{\beta}] = (x^T x + \lambda I)^{-1} x^T \sigma_\varepsilon^2 [(x^T x + \lambda I)^{-1} x^T]^T$

$= \sigma_\varepsilon^2 (x^T x + \lambda I)^{-1} x^T x [(x^T x + \lambda I)^{-1}]^T$ .... ⑥

So $\hat{\beta} \sim N(⑤, ⑥)$ ..... ⑦

part 2. Bias $= E[x^T\hat{\beta}] - x^T\beta^*$

according to equation ④

$= E[x^T(x^Tx+\lambda I)^{-1}x^Ty] - x^T\beta^* \quad \cdots ⑧$

$\because y \sim N(x^T\beta^*, \sigma_\varepsilon^2)$

$= E[x^T(x^Tx+\lambda I)^{-1}x^Tx\beta^* - x^T\beta^*$

$= x^T(x^Tx+\lambda I)^{-1}x^Tx\beta^* - x^T(x^Tx+\lambda I)^{-1}(x^Tx+\lambda I)\beta^*$

$= x^T(x^Tx+\lambda I)^{-1}(-\lambda I)\beta^*$

$= -\lambda x^T(x^Tx+\lambda I)^{-1}\beta^* \quad \cdots ⑨$

part 3. According to ① - ③, ④

$x^T\hat{\beta} = x^T(x^Tx+\lambda I)^{-1}x^Ty$

$\mathrm{Var}[x^T\hat{\beta}] = x^T(x^Tx+\lambda I)^{-1}x^T\sigma_\varepsilon^2[x^T(x^Tx+\lambda I)^{-1}x^T]^T$

$= \sigma_\varepsilon^2 x^T(x^Tx+\lambda I)^{-1}x^Tx[(x^Tx+\lambda I)^{-1}]^Tx \quad \cdots ⑩$

part 4. Squared error = Bias^2 + Variance + noise

According to ⑨, ⑩

$= [-\lambda x^T(x^Tx+\lambda I)^{-1}\beta^*]^2 + \sigma_\varepsilon^2 x^T(x^Tx+\lambda I)^{-1}x^Tx[(x^Tx+\lambda I)^{-1}]^Tx + \text{noise}$

When λ is small, variance dominates

When λ is large, Bias dominates.