HW7

The code for calculations and generating the tree is in the Jupyter notebook in the codes directory.

Q1-1:

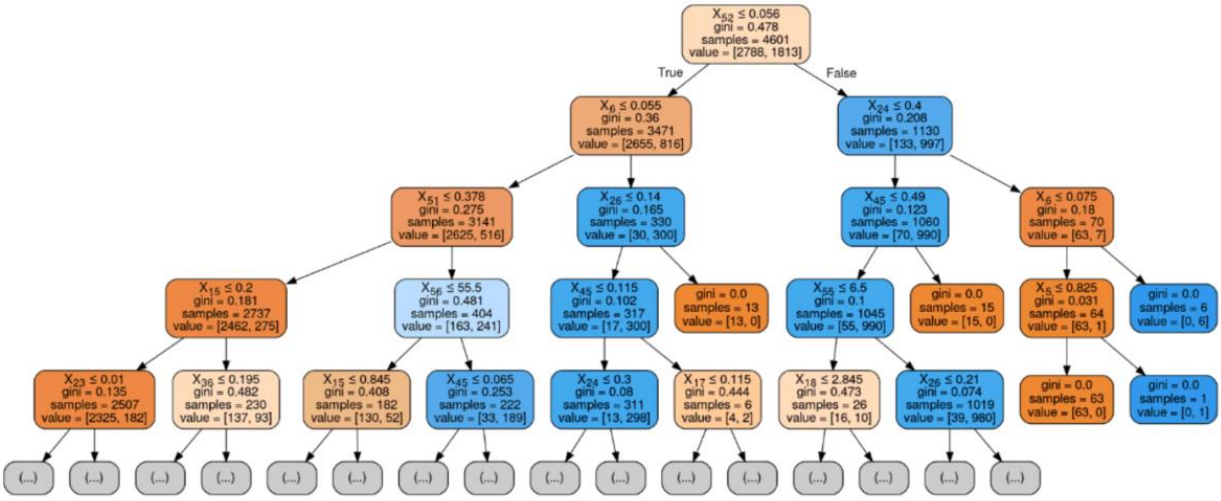
There are 1813 spam emails and 2788 not spam emails.

Q1-2:

The fitted decision tree model is:

```
DecisionTreeClassifier(ccp_alpha=0.0, class_weight=None, criterion='gini',
                       max_depth=None, max_features=None, max_leaf_nodes=None,
                       min_impurity_decrease=0.0, min_impurity_split=None,
                       min_samples_leaf=1, min_samples_split=2,
                       min_weight_fraction_leaf=0.0, presort='deprecated',
                       random_state=None, splitter='best')
```

I used two libraries to generate the graphics for the tree. Both were too large to fit here in the report. I used portion of the tree for our report here. In the Jupyter, I have full tree with both libraries.



Q1-3:

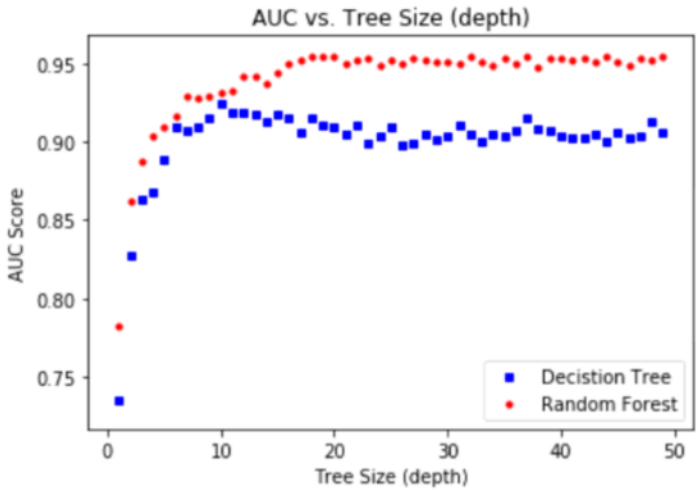
For this section, I calculated the AUC for both Random Forest Tree and Decision Tree for various tree size (depth). The fitted decision tree is as in Q1-2 and the random forest is:

```
RandomForestClassifier(bootstrap=True, ccp_alpha=0.0, class_weight=None,
                       criterion='gini', max_depth=None, max_features='auto',
                       max_leaf_nodes=None, max_samples=None,
                       min_impurity_decrease=0.0, min_impurity_split=None,
                       min_samples_leaf=1, min_samples_split=2,
                       min_weight_fraction_leaf=0.0, n_estimators=100,
                       n_jobs=None, oob_score=False, random_state=None,
                       verbose=0, warm_start=False)
```

The result is shown the figure in the next page.

The ultimate accuracy of random forest (96%) is higher than decision tree (92%). The results show that while random forest accuracy increased by increasing the tree size (depth) up to 20, from that size, the algorithm started to flatten out the accuracy did not change (or in case of decision tree the accuracy worsen then flatten out). The decision tree's accuracy increased up to depth increase to 10 and from there, the accuracy decreased.

HW7



Q2-1:

The fitted linear regression model and the **fitting error** is as below:

```
LinearRegression(copy_X=True, fit_intercept=True, n_jobs=None, normalize=False)
The intercept [7.38412739]
The slope [[0.02128314]]
The root mean square error 3.243561781943323
The r2 0.6861858695095129
```

Q2-2:

The strategy is to first evaluate the Ridge regression model with only lambda = 0. Eliminating lambda will make the model, a typical polynomial regression model. I will test various polynomial degrees (2 to 10). We will pick the best fit data (least RMSE). The expectation is to witness the RMSE goes down by increasing the polynomial degree and then at some point the fit get worsen. The strategy to select lambda for ridge regression is explained in the **next question**.

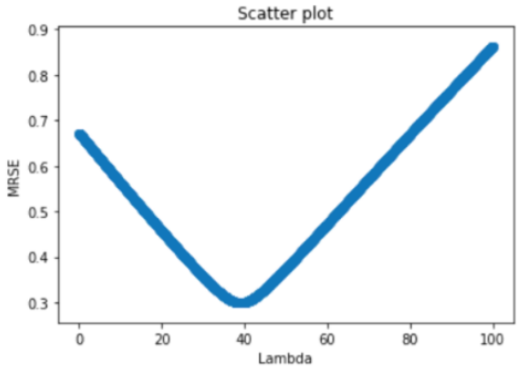
The best polynomial degree for my model is degree 5.

```
Ridge(alpha=0, copy_X=True, fit_intercept=True, max_iter=None, normalize=False,
      random_state=None, solver='auto', tol=0.001)

best polynomial degree 5
best RMSE 0.3225830042568235
```

Q2-3:

For this section, we used the 5 KFold technique. And to define the best lambda we did 10000 iteration by starting lambda with 0.01 and increase lambda with iteration by 0.01. I got the result below:



HW7

This result shows the best lambda is around 40 for the polynomial model of degree 5. The model (polynomial degree 5 and ridge regression lambda 39.03) predicted the coefficient for 400 as 17.43 while the linear regression predicted 15.90 as below.

```
model = LinearRegression()
model.fit(X, y)
y_pred = model.predict([[400]])
print(y_pred)
```

```
[[15.89738509]]
```

```
polynomial_features= PolynomialFeatures(degree=bestDegree)
X_poly = polynomial_features.fit_transform(X)

model = Ridge(alpha=39.03)
model.fit(X_poly, y)
y_pred = model.predict(polynomial_features.fit_transform([[400]]))
print(y_pred)
```

```
[[17.42891061]]
```

By checking the scatter plot for coefficient and temperature, the results for polynomial is more accurate compare to linear regression.

HW7

Q3-1 The regulized linear regression is.

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

$$\simeq \arg\min_{\beta} \left\{ \sum_{i=1}^{n} (y_i - x_i^T \beta)^2 + \lambda \sum_{j=1}^{n} \beta_j^2 \right\}$$

the criterion to be minimzed can be written:

$$f(\beta, \lambda) = (y - X\beta)^T (y - X\beta) + \lambda \beta^T \beta$$

To minimize, we take the derivative of $f(\beta, \lambda)$ by β:

$$\frac{\partial f(\beta,\lambda)}{\partial \beta} = \frac{\partial}{\partial \beta} (y^2 - 2X\beta y + X^2\beta^2 + \lambda\beta^2)$$

$$= 2X^2\beta - 2Xy + 2\lambda\beta$$

$$= 2(X^TX)\beta - 2X^Ty + 2\lambda I\beta = 0$$

$$\Rightarrow 2(X^TX)\beta + 2\lambda I\beta = 2X^Ty$$

$$\Rightarrow (X^TX)\beta + \lambda I\beta = X^Ty$$

$$\Rightarrow \hat{\beta}_{ridge} = \beta = (X^TX + \lambda I)^{-1} X^Ty$$

closed form

let assume $\hat{\beta}_{ridge}$ has Gaussian distribution then to define the distribution we need the mean and variance $N(\mu, \delta^2)$

$$\mathcal{E}[\hat{\beta}_{ridge}] = \mathcal{E}[(X^TX + \lambda I)^{-1} X^Ty] = (X^TX + \lambda I)^{-1} X^T \mathcal{E}[y]$$

$$= (X^TX + \lambda I)^{-1} X^TX\beta$$

$\mathcal{E}[y] = \mathcal{E}[X^T\beta + \varepsilon] = X^T\beta + 0$ ①

Gaussian noise with 0 as mean

$X^TX^T = X^{T^2} = X^TX$ ②

HW7

$$\text{Var}[\hat{\beta}_{Ridge}] = \text{Var}[(X^TX+\lambda I)^{-1}X^Ty]$$

$$= (X^TX+\lambda I)^{-1\times 2}\, X^{T^2}\, \text{Var}[y]$$

$$= (X^TX+\lambda I)^{-1}X^T\,\text{Var}[y]\,(X^TX+\lambda I)^{-1}X^T$$

$$= (X^TX+\lambda I)^{-1}X^T\sigma^2(X^TX+\lambda I)^{-1}X^T$$

$$= \sigma^2(X^TX+\lambda I)^{-1}X^TX\,(X^TX+\lambda I)^{-1}$$

$\text{Var}[ax] = a^2\,\text{Var}[x]$

the distribution of $\hat{\beta}_{Ridge}$ will be $\mathcal{N}(\mathcal{E}[\hat{\beta}_{Ridge}], \text{Var}[\hat{\beta}_{Ridge}])$

Q3-2: $\text{Bias}(\hat{y}) = \mathcal{E}[X^T\hat{\beta}_{Ridge}] - X^T\beta$

$$= X^T\mathcal{E}[\hat{\beta}_{Ridge}] - X^T\beta$$

$$= X^T(\mathcal{E}[\hat{\beta}_{Ridge}] - \beta)$$

$$= X^T((X^TX+\lambda I)^{-1}X^TX\beta - \beta)$$

$$= X^T((X^TX+\lambda I)^{-1}(X^TX+\lambda I-\lambda I)\beta - \beta)$$

$$= X^T(I-\lambda(X^TX+\lambda I)^{-1}\beta - \beta)$$

$$= X^T(\beta-\lambda(X^TX+\lambda I)^{-1}\beta - \beta)$$

$$= -X^T\lambda(X^TX+\lambda I)^{-1}\beta$$

$\text{Bias} = \mathcal{E}[\hat{y}] - y$

$\beta = \beta^*$ is considered constant

HW7

Q3-3

$$E\left[\left(X^T\beta_{Ridge} - E\left[X^T\hat{\beta}_{Ridge}\right]\right)^2\right] = Var\left(X^T\hat{\beta}_{Ridge}\right)$$

$$= X^T Var\left(\hat{\beta}_{Ridge}\right) X =$$

$$= X^T\left(\delta^2 (X^TX+\lambda I)^{-1} X^TX (X^TX+\lambda I)^{-1}\right) X$$

$Var(X) = E[(X - E[X])^2]$

$Var(aX) = a^2 Var(X)$

according Q3-1

Q3-4: $MSE\left(\hat{\beta}_{Ridge}\right) = Var\left[X^T\hat{\beta}_{Ridg}\right] + bias\left[X^T\hat{\beta}_{Ridg}\right]^2$

Results from b & c

- when $\lambda \to \infty$

$$Bias\left[X^T\hat{\beta}_{Ridge}\right] \underset{\lambda\to\infty}{=} -X\lambda (X^TX + \lambda I)^{-1}\beta$$

$$\underset{\lambda\to\infty}{=} \frac{-X\lambda\beta}{X^TX + \lambda I} \simeq \frac{-X\beta}{I} \quad (1)$$

this shows that MSE will depend on the coefficent β.

$$Var\left[X^T\hat{\beta}_{Ridg}\right] \underset{\lambda\to\infty}{=} 0 \quad (2)$$

all λ are in the denominator

$\lim_{x\to\infty} \frac{1}{x} = 0$ $\lim_{x\to 0} \frac{1}{x} = \infty$

(1), (2) => when λ increase, Bias increases and Variance decreases

- when $\lambda \longrightarrow 0$

$$Bias\left[X^T\hat{\beta}_{Ridg}\right] \underset{\lambda\to 0}{=} 0 \quad (1), \quad Var\left[X^T\hat{\beta}_{Ridge}\right] \underset{\lambda\to 0}{=} \infty \quad (2)$$

(1), (2) => when λ decreases, Bias decreases and Variance increases