Title: Data7202 A1 Report

Name: Xinqian Wang

ID: s4565489

Tips: the code screenshot will be attached in the Appendix. For code script will also be attached in the folder.

- ## 1.

Question: For the linear regression, make an inference about the coefficients, specifically, comment about the contributions of different advertisement types to sales.

Answer:

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                      y   R-squared:                       0.876
Model:                            OLS   Adj. R-squared:                  0.876
Method:                 Least Squares   F-statistic:                     3516.
Date:                Thu, 18 Mar 2021   Prob (F-statistic):               0.00
Time:                        14:50:07   Log-Likelihood:                -335.10
No. Observations:                1000   AIC:                             676.2
Df Residuals:                     997   BIC:                             690.9
Df Model:                           2
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
radio          0.6519      0.042     15.683      0.000       0.570       0.733
tv             4.4333      0.040    111.214      0.000       4.355       4.512
internet       6.4935      0.043    151.444      0.000       6.409       6.578
==============================================================================
Omnibus:                       22.387   Durbin-Watson:                   2.009
Prob(Omnibus):                  0.000   Jarque-Bera (JB):               19.023
Skew:                          -0.269   Prob(JB):                     7.40e-05
Kurtosis:                       2.591   Cond. No.                         2.71
==============================================================================
```

From the table above we can see from the coefficient that the most relevant advertisement to the sales is the internet, the second is tv and the last is radio.

Question: Use the linear model and the RF (with 500 trees), to make a prediction (using the test set), and report the corresponding mean squared errors.

Answer:

```
************************************************************
Regression mean squared error(testing):  0.10487158803448701

random forest loss(testing):  0.10422862569354253

************************************************************

regression (training) mean squared error:  0.11444251260397326

random forest(training) loss:  0.014170112310322194
************************************************************
```

- **2.**

Question: Is this a good method? Do you expect to obtain the true prediction error? Explain your answer.

Answer:

No, It's not a good method. The model we got based on the question is a biased model with the correlations average not closed to 0. Hence, the samples have already been saw by the predictors. The correct way to do the K fold CV should be some procedures like this below:

1. Divide the samples into K folds randomly.
2. For each fold, finding a subset of good predictors which show strong correlations with the labels by using all the samples
   except in fold k.
3. Using the subset of these predictors to build a multivariate classifier, using all the samples except those in fold k.
4. Using the multivariate classifier to make predictions on the samples in fold k.

- **3.**

Question: For this problem, determine the hypothesis class and state explicitly what is θ and Θ.

Answer:

Normal Distribution

$\mathcal{H} = \{ f(x,\theta) ; \theta \in \Theta$

The hypothesis class is the set of possible classification functions we're considering.

In this case, it should be:

$$f(x,\theta) = f(x;\mu,\sigma) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$$

$\theta = \mu, \sigma$

$$\mu = \frac{1}{n}\sum_{i=1}^{N} X_i$$

$$\sigma = \sqrt{\sum_{i=1}^{N} [X_i-\mu]^2 / N}$$

Θ

$$\boxed{\begin{array}{l} \sigma \in (0,+\infty) \\ \mu \in R \end{array}}$$

- **4.**

Question: Show that the expected value of LossT (g) over the choice of T equals LossD(g).

Answer:

Q4 Define $\mathcal{Z} = \mathcal{X} \times \mathcal{Y}$

$$T = (z_1, \dots, z_m) \in \mathcal{Z}^m$$

$$E_T \, Loss_T(g) = E_T\left[\frac{1}{m}\sum_{i=1}^{m} 1[g(x_i) \neq z_i]\right]$$

$$= \frac{1}{m}\sum_{i=1}^{m} \Pr_{x_i \sim D}[g(x_i) \neq z_i]$$

$$= \frac{1}{m} \cdot m \cdot Loss_D(g)$$

$$= Loss_D(g)$$

- **5.**

Question: Fit these models tot the data and write the corresponding coefficients. Namely, fill the following table:

Answer:

Model	β_0	β_1
Model $_1$	0	0.6
Model $_2$	1.8	0

Question: Consider the squared error loss, the absolute error loss, and the L1.5 loss. Find the average loss for each model. Namely, fill the following table:

Answer:

Model	squared error loss	absolute error loss	$L_{1.5}$ loss
Model $_1$	1.64	1.16	1.36
Model $_2$	0.56	0.64	0.58

Question: Draw a conclusion from the obtained results.

Answer:

Only doing simple linear regression seems not fit the label Y so well. We may need to involve some complexity. We can see from the loss value that model 2 would fit better than model 1.

- **6.**

Question: Load the data-set and replace all categorical values with numbers. (You can use the LabelEncoder object in Python).

Answer:

Attached in the code

Question: Generally, it is better to use OneHotEncoder when dealing with categorical variables. Justify the usage of LabelEncoder in (a).

Answer:

OneHotEncoder can handle the features with 3 or more unique values. But I think the drawbacks would be it will make the number of columns larger.

However, if there exist the features with only 2 unique values, LabelEncoder would be better.

Question: Fit linear regression and report 10-Fold Cross-Validation mean squared error.

Answer:

Attached in the code

- **7.**

Question: Deliver the 95% confidence interval.

Answer:

```
In [10]: import numpy as np
         import math

In [11]: s = np.random.uniform(0, 1, 10000)

In [12]: def function_Fx(x):
             return (x*x + 2*x + 3)**(-1)

In [13]: result = function_Fx(s)

In [14]: result_mean = np.mean(result)
         result_std = np.std(result)
         result_up = result_mean + 1.96*(result_std/100)
         result_down = result_mean - 1.96*(result_std/100)
         print("confidence interval of the function_Fx","[{lower},{upper}]".format(lower=result_down,upper=result_up))

         confidence interval of the function_Fx [0.23998100095187747,0.24186070055522024]
```

Question: Compare the obtained estimation with the true value ` as given in (2).

Answer:

The result calculate by the Crude Monte Carlo Algorithm is 0.24092085075354872.

The real value calculated by the formula is 0.24030098317248838.

The gap between them is just 0.00062.

Appendix

- **1.**

  code

- **2.**

  None

- **3.**

  Image

- **4.**

  Image

- **5.**

  code

- **6.**

```
In [72]: Y = df["Salary"]
         X = df = df.drop(['Salary'], axis=1)
```

(c)

```
In [73]: from sklearn.model_selection import KFold
         from sklearn.metrics import mean_squared_error
         from sklearn.linear_model import LinearRegression
         def Validate(X,Y,model):
             kf = KFold(n_splits=10)
             kf.get_n_splits(X)
             zer_one_err = []
             for train_index, test_index in kf.split(X):
                 #print("TRAIN:", train_index, "TEST:", test_index)
                 X_train, X_test = X.iloc[train_index,:], X.iloc[test_index,:]
                 y_train, y_test = Y.iloc[train_index], Y.iloc[test_index]
                 # fit the model
                 model.fit(X_train,y_train)
                 # predict
                 y_pred = model.predict(X_test)
                 loss = mean_squared_error(y_test,y_pred)
                 zer_one_err.append(loss)
             return np.mean(zer_one_err)

In [74]: model = LinearRegression()
         err = Validate(X,Y,model)

In [75]: err
Out[75]: 116599.01367380242
```

  code

- **7.**

  code