**Dear all students,**

**Follow the instructions carefully, please:**

* Kindly, you have 120 minutes to answer the following three questions, and **10 minutes of upload** time (not extendable).
* For the questions that you need to answer in the text, add your answer in this Word file and upload it into Moodle. For the questions that you need to write code, upload your Python file to Moodle. Rename both files using your English name or student ID.
* You can use Slides and class code examples.
* **Do not** use .rar or zip to upload answers (you will lose a mark if compressed); you must upload your **Python** code and **Word** file in Moodle.
* No need to mention that using the Internet or communicating during the exam is prohibited.
  + Cell phones are only in the pocket all the time (you must disconnect your computer from any wifi or internet network during the exam).

**Good Luck.**

--------------------------------------------------------------------------------------------------

English Name: Hsin Yu, Chen

Chinese name: 陳昕佑

Student ID: 61375017H

--------------------------------------------------------------------------------------------------

**Questions:**

1. (a) Which model is the best among the four models in Figure 1, and explain what we understand from the slope error (SE) of these four models? (b) Explain if we can check the Null hypothesis with the given slope error (SE) in these models or not, and why?

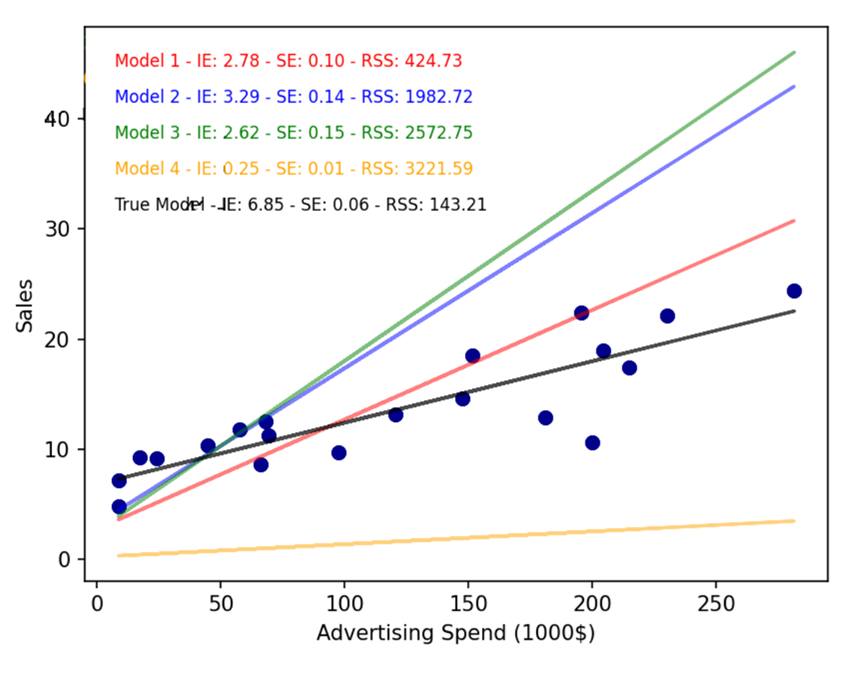


Figure 1

Answer:

1. Model 1 is the best model which intercept error and slope error minimize the RSS to the smallest. If the slope error is closer to the true model’s slope error, then the model would fit more to the true model, but intercept error control up and down in the figure, so it’s not only depends on slope error to fit the true model.
2. Yes, we can check the null hypothesis with SE as know as . If , then the model from can be simplified to .
3. (a) What are the differences between K-means, K Nearest Neighbor (KNN), and K-medoid algorithms? (b) If we apply a 4\*4 filter and apply a Sliding Window with Stride 1 on an image of 9\*9 size, what is the size of the output node? Also, what is the output if we also apply a Maxpooling of 3 × 3 filter with a stride of three on that output?

Answer:

1. KNN is non-parametric classification in supervised learning and does not build a model during the training phase instead it stores the entire training dataset. K-means is a clustering algorithm used in unsupervised machine learning. K medoids is a variant of K means clustering. The key difference between K means and K medoids is that K medoids uses medoids instead of centroids.
2. 6\*6, 2\*2 after Maxpooling
3. Complete and implement a decision tree model on the following data, in which you should find the best k-fold cross-validation (5-fold) results for one of MSE, R-squared, and F-statistic after the models’ convergence. (You are allowed to use Python libraries). Note that you might need to **tune the tree** (e.g., max\_depth, min\_samples\_leaf to find the parameters that minimize the error, which is the average of test set errors.

Hint: regressor = DecisionTreeRegressor(max\_depth=depth, random\_state=42)

Other hyperparameters:

min\_samples\_split

min\_samples\_leaf

max\_features

import pandas as pd

import matplotlib.pyplot as plt

from mpl\_toolkits.mplot3d import Axes3D

data = pd.DataFrame({

    'SocialMedia': [

        340.1, 154.5, 127.2, 261.5, 290.8, 115.7, 167.5, 230.2, 115.6, 309.8,

        176.1, 324.7, 130.8, 207.5, 314.1, 305.4, 177.8, 391.4, 179.2, 257.3,

        374.11, 169.95, 139.92, 287.65, 319.88, 127.27, 184.25, 253.22, 127.16, 340.78,

        193.71, 357.17, 143.88, 228.25, 345.51, 335.94, 195.58, 430.54, 197.12, 283.03,

        323.095, 146.775, 120.84, 248.425, 276.26, 109.915, 159.125, 218.69, 109.82, 294.31,

        167.295, 308.465, 124.26, 197.125, 298.395, 290.13, 168.91, 372.83, 170.24, 244.435

    ],

    'Billboard': [

        169.2, 145.1, 169.3, 157.5, 157.4, 182.0, 130.5, 118.6, 108.0, 128.2,

        123.2, 103.0, 164.9, 106.2, 145.0, 151.9, 213.0, 154.8, 117.3, 126.1,

        177.66, 152.355, 177.765, 165.375, 165.27, 191.1, 137.025, 124.53, 113.4, 134.61,

        129.36, 108.15, 173.145, 111.51, 152.25, 159.495, 223.65, 162.54, 123.165, 132.405,

        152.28, 130.59, 152.37, 141.75, 141.66, 163.8, 117.45, 106.74, 97.2, 115.38,

        110.88, 92.7, 148.41, 95.58, 130.5, 137.79, 193.05, 140.82, 106.57, 114.645

    ],

    'Sales': [

        29.1, 17.4, 16.3, 25.5, 19.9, 14.2, 18.8, 20.2, 11.8, 18.6,

        16.6, 23.4, 15.2, 15.7, 26.0, 29.4, 19.5, 31.4, 18.3, 21.6,

        30.371015, 17.714363, 17.538445, 27.397403, 20.80007, 14.751211, 20.826538, 21.55986, 13.139398, 19.204558,

        16.979135, 23.812623, 16.345842, 17.244362, 27.167014, 30.092623, 21.099325, 33.133781, 18.751607, 22.305827,

        27.667176, 15.608117, 15.723191, 24.124267, 19.01121, 13.662289, 18.241651, 19.730123, 10.968067, 17.680196,

        15.778138, 21.662496, 14.495963, 14.460843, 24.996702, 27.738864, 18.649246, 29.420281, 17.243353, 20.276602

    ]

})

fig = plt.figure(figsize=(10, 8))

ax = fig.add\_subplot(111, projection='3d')

ax.scatter(data['SocialMedia'], data['Billboard'], data['Sales'], s=60)

ax.set\_xlabel('Social Media Spend')

ax.set\_ylabel('Billboard Spend')

ax.set\_zlabel('Sales')

ax.set\_title('3D Scatter Plot of Advertising Spend vs Sales')

plt.show()

**Note:** Attach the Python code to Moodle (do not compress files as .zip or .rar files)