

To,

Codistan Ventures

From,

Muhammad Huzaifa

17th February 2023

Subject: AI Python Internship Task

Git Rep: <https://github.com/huzaifah-here/Codistan>

Task 1: <https://github.com/huzaifah-here/Codistan/blob/master/task1.py>

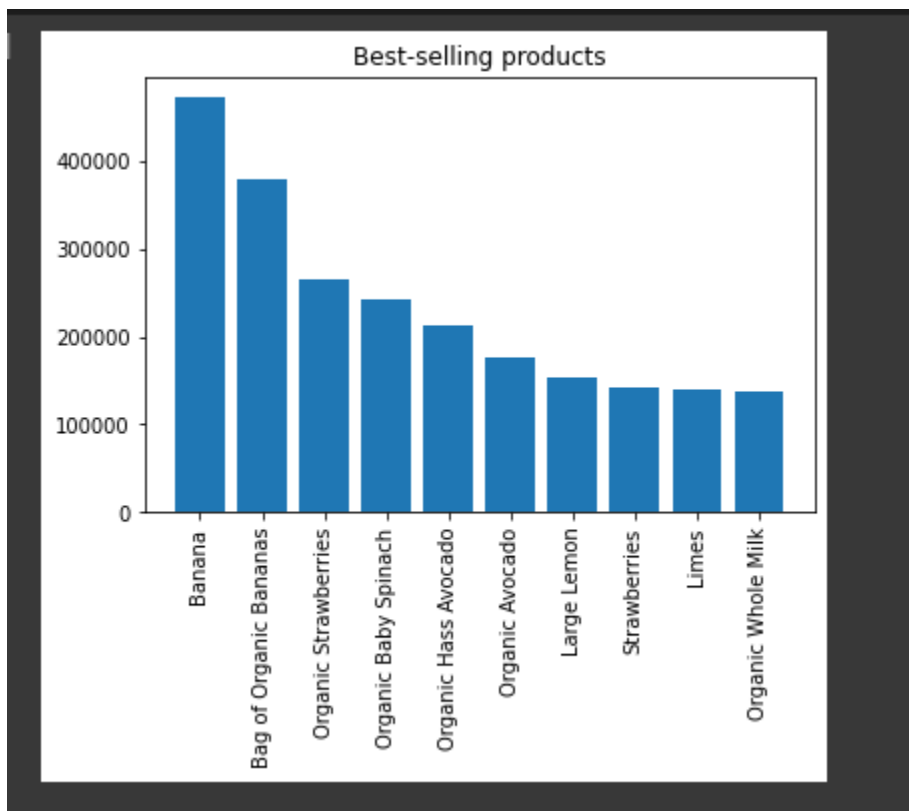
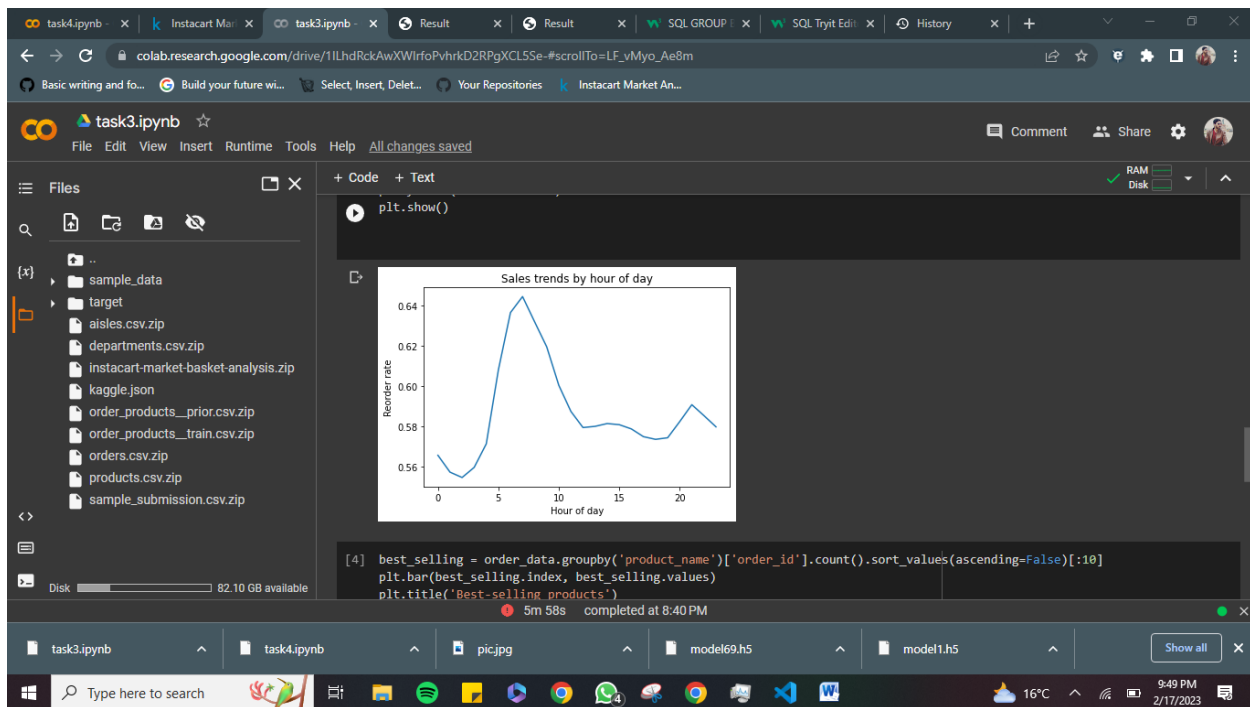
- Clone the function
- Reverse the arguments when the function call
- Return reverse arguments output

Task 2: <https://github.com/huzaifah-here/Codistan/blob/master/task2.py>

- We want to print company's name Hence Select name
- Name from company c, where c is variable
- It join between the COMPANY and EMPLOYEE tables using the ID column
- Similarly, Salary and Employee
- Show company's name group
- If salary greater or equal to 40,000

Task 3: <https://github.com/huzaifah-here/Codistan/blob/master/task3.ipynb>

- Read_csv() to load dataset
- Group order data, calculate avg, reorder rate of data
- For best selling graph, group order by product name and count number of times it has been ordered
- Matplot is used for plotting graphs



Task 4: <https://github.com/huzaifah-here/Codistan/blob/master/task4.ipynb>

- Used Image generator to prepare image data
- Sklearn lib has been used for splitting 80/20

- Created Model consist on 10 layers
- Compiled and adam optimizer has been used
- For fitting, used 5 iteration coz taking too much time if 25
- Check the model by giving test image print class label after predicting image
- Save the model

```
model = tf.keras.models.Sequential([
    tf.keras.layers.Conv2D(32, (3,3), activation='relu', input_shape=(224,224,3)),
    tf.keras.layers.MaxPooling2D(2,2),
    tf.keras.layers.Conv2D(64, (3,3), activation='relu'),
    tf.keras.layers.MaxPooling2D(2,2),
    tf.keras.layers.Conv2D(128, (3,3), activation='relu'),
    tf.keras.layers.MaxPooling2D(2,2),
    tf.keras.layers.Flatten(),
    tf.keras.layers.Dense(256, activation='relu'),
    tf.keras.layers.Dropout(0.5),
    tf.keras.layers.Dense(5, activation='softmax')
])
```

```
[ ] model.compile(optimizer='adam',
                  loss='categorical_crossentropy',
                  metrics=['accuracy'])
```

-



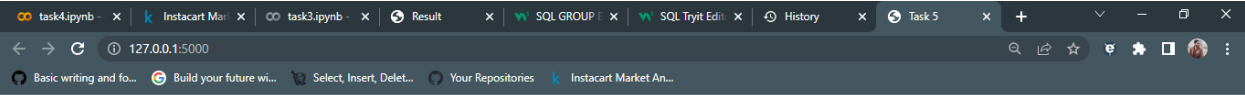
```
1/1 [=====] - 0s 68ms/step  
Predicted class: Rose
```

```
[ ] score = model.evaluate(test_generator, verbose=0)  
print('Test loss:', score[0])  
print('Test accuracy:', score[1])
```

```
Test loss: 0.8876707553863525  
Test accuracy: 0.6693641543388367
```

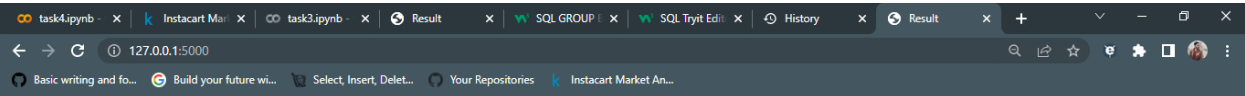
Task 5: <https://github.com/huzaifah-here/Codistan/tree/master/task5>

- Simple html files created
- Load the model
- Index page given with browse button and post button
- Model return integers according to the model training
- Dictionary has been created to show class name
- Result page to show class name of the predicted model



Flower detector

No file chosen



Result: Rose

