The affiliated senior high school of National Taiwan Normal University independent study diary

Group: information technology

Teacher: Jason

Class:1612

Number:05

Name: Wu, Tai-Cheng

All code can be found on Github. This file was originally a Markdown file, however, due to the restriction rule made by the school, it's converted to pdf format, which might result in some formatting problem. To access the original file. visit the original file on Github.

11/9

Machine learning introduction

Today, our teacher taught us about machine learning, ML.

11/16

House Value prediction

Today, our teacher taught us how to predict house prices using linear regression and also how to fill in the missing value

```
import seaborn as sns
import pandas as pd
import numpy as np
import random
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_squared_error
from sklearn.linear_model import LinearRegression
```

this code demonstrates the package we need to import for our data analysis

```
df=pd.read_csv("dataset/train_data_titanic.csv")
```

this code shows us how to import a dataset as Dataframe in pandas

11/23

Titanic prediction

Today, our teacher taught us how to do the classic question - titanic. We use logistic regression to classify two category

```
df.drop(['Name','Ticket'],axis=1,inplace=
        True)
```

this code will drop a column that we dont want, usually, containing text type data or too many missing values.

```
betterdf= pd.DataFrame(KNNImputer().fit_transform(kdf))
```

this code shows us to fill missing value using KNN.

```
model = LogisticRegression(max_iter=1000)
```

this code is setting the model to logistic regression and the maximum iteration value is 1000. If we set max_iter too little, its performance might weaken, if we set it too high, it might be a waste of time.

```
accuracy = accuracy_score(y_test, y_pred)
```

This is the function by which we evaluate our model's performance.

11/30

Kaggle

Today, our teacher taught us how to upload our prediction to Kaggle, a web platform that has a lot of models and datasets

```
k=10000
lines=np.random.rand(k)*100
grades=np.zeros(k)
for i in range(k):
    grades[i]=math.floor(lines[i]/2+2*np.random.normal())
df=pd.DataFrame(data={'line':lines,'grade':grades})
```

This code is generating a set of data that are similar to what has happened for our school English essay writing exam.

```
plt.scatter(sx, sy, color='black',marker='.')
plt.plot(sx, syy, color='blue', linewidth=3)
plt.xlabel('Line')
plt.ylabel('Grade')
plt.title('Best Fit Line in Linear Regression')
plt.show()
```

This is the code we use to show the figure and evaluate using our own vision.

```
sumbitdf=pd.DataFrame(columns=['PassengerId','Survived'])
sumbitdf['PassengerId']=range(892,1310)
sumbitdf['Survived']=pred
for i in range(0,418):
  sumbitdf['Survived'][i]=round(sumbitdf['Survived'][i])
sumbitdf.to_csv('for_submission_2023113004.csv', index=False)
```

This is the code we use to submit our data to Kaggle.

12/7

Loan prediction

Today, our teacher taught us how to predict whether the client will return the money in time.

```
table = df.pivot_table(
    values="LoanAmount", index="Self_Employed",
    columns="Education", aggfunc=np.median
)
```

This code will generate a table to account for different types of values.

```
joblib.dump(modelk, "2023121401.pkl")
```

This code is used to output our trained model.

```
modelk,
r,a,f1=modelfun(RandomForestClassifier(n_estimators=20),df,var_mod,'Loan_St
atus')
```

This code will call the function we wrote and use the random forest classifier to classify the data we give which is df.

12/14

Neural networks introduction

Today, our teacher taught us how to use neural networks to predict and categorize.

```
import joblib
```

```
# Load the model
model = joblib.load("2023121401.pkl")

# Collect input from the user
gender = input("Enter gender (0 for Male, 1 for Female): ")
married = input("Are you married? (0 for No, 1 for Yes): ")
dependents = input("Number of dependents: ")
education = input("Education level (0 for Not Graduate, 1 for Graduate): ")
self_employed = input("Are you self-employed? (0 for No, 1 for Yes): ")
property_area = input("Enter property area (0 for Urban, 1 for Semiurban, 2
for Rural): ")

# Convert input to numerical format
input_data = [int(gender), int(married), int(dependents), int(education),
int(self_employed), int(property_area)]

# Make predictions
pred = model.predict([input_data])
print(pred)
```

This is the cli interface for our model. (CLI, command-line interface)

```
@app.route("/mainpage")
def mainpage():
    return render_template("121403.html")
```

This code is the main web interface code.

```
@app.route("/proceed", methods=['GET', 'POST'])
def proceed():
    if(request.method=='GET'):
        return 'f'
    else:
        model=joblib.load("2023121401.pkl")
        inputdata=
[int(request.values['gender']),int(request.values['married']),int(request.v
alues['dependents']),int(request.values['education']),int(request.values['s
elfemp']),int(request.values['pa'])]
        return
(str(model.predict([inputdata]))+str(model.predict_proba([inputdata])))
```

This is the backend of our web interface. RefL 121403.html, this is the awesome html code i wrote.

```
import tensorflow as tf
print(tf.__file__)
```

This will print the location of tensorflow we used. It's useful for debugging env problem.

```
from keras.datasets import mnist
(X_train, y_train), (X_valid, y_valid) = mnist.load_data()
```

This is how we import our pre-processed data.

12/21

MNIST dataset prediction

Today, our teacher taught us how to recognize the hand-written numbers from a famous dataset, MNIST, using neural networks.

```
df=pd.read_csv("dataset/kaggle/train.csv")
xtrain,xtest,ytrain,ytest=train_test_split(df['filepaths'],df['Font'],test_
size=0.3,random_state=83)
xtrainimg =[]
for filepath in xtrain:
    # Open the image file
    img = image_utils.load_img(filepath,color_mode="grayscale",target_size=
(128,128))
    # Convert the image to a NumPy array
    img_array = np.array(img)
    # Append the array to the list
    xtrainimg.append(img_array)
```

This is the code we used to convert the image to the type where our model can processed

```
def replace_letters_with_numbers(input_list):
    result_list = []

    for original_string in input_list:
        modified_string = ""
        for char in original_string:
            if 'a' <= char <= 'z':
                modified_string += str(ord(char) - ord('a') + 11)
            elif 'A' <= char <= 'Z':
                modified_string += str(ord(char) - ord('A') + 11)
            else:
                modified_string += char

        result_list.append(modified_string)

    return result_list
```

This code can convert the letter to numbers we will be ussing as the index in the future.

12/28

Transfer learning

Today, our teacher taught us how to use transfer learning to use the pre-trained model to predict our own data. Also, we start to think about what project we are going to do.

In addition, I also try to do Large Language Model (aka LLM) with GPT-2.

```
keras.mixed_precision.set_global_policy("float32")
preprocessor = keras_nlp.models.GPT2CausalLMPreprocessor.from_preset(
    "gpt2_base_en",
    sequence_length=128,
)
gpt2_lm = keras_nlp.models.GPT2CausalLM.from_preset(
    "gpt2_base_en", preprocessor=preprocessor
)
```

This code will load GPT-2 model given by keras_nlp. The reason to set the precision policy to f32 aka float32, is to ensure accuracy and performance. This code is originally from keras official document about machine learning.

```
with open("usc.txt", 'r', encoding='utf-8') as file:
    paragraphs=[file.read()]
```

We use the United States Code aka USC as our train data. The dataset is not provided to the public, however, it can be found online easily.

1/4

Project preparation

Today, we finished our project and the presentation slide.

We use ssh to access the AI server at our school use ChatGLM-6b as our base model and fine-tune the model with ptunning, which is both fast and effective. It's all about debugging and waiting.

The code can be found on the original creator's repository. For security reasons, the process of debugging is not shown publicly.

1/11

Assessment and presentation

Today, we finished the assessment and presented our project to judges and the chairman from Mitac.