analyze python Categorical classification task

[paste task ]

the folder structure is

- python code file

- Ecommerce

|-train\_data

|-test\_data

|-test\_data\_hidden

train\_data csv dataset like

[paste column header and first row ]

test\_data csv dataset like

[paste column header and first row ]

test\_data\_hidden csv dataset like

[paste column header and first row ]

make split of 0.8 train and 0.2 in validation give Task wise code

[you will get version 1]

modify code to include # Plot the training and validation accuracy and loss plt.plot(history.history['accuracy'], label='Training Accuracy') plt.plot(history.history['val\_accuracy'], label='Validation Accuracy') plt.legend() plt.title('Training and Validation Accuracy') plt.xlabel('Epoch') plt.ylabel('Accuracy') plt.show() and plt.plot(history.history['loss'], label='Training Loss') plt.plot(history.history['val\_loss'], label='Validation Loss') plt.legend() plt.title('Training and Validation Loss') plt.xlabel('Epoch') plt.ylabel('Loss') plt.show() and from sklearn.metrics import classification\_report # Use the predict() method to get the predicted labels for the test data # Use the model to make predictions on the test data y\_pred\_prob = model.predict(test\_dataset) # Convert the predicted probabilities to class indices y\_pred = np.argmax(y\_pred\_prob, axis=1) # Print the classification report print(classification\_report(y\_test, y\_pred)) and # Save the trained model to a file model.save('vitMixup.h5') from sklearn.metrics import confusion\_matrix # Use the model to make predictions on the test data y\_pred = np.argmax(model.predict(test\_dataset), axis=1) # Create the confusion matrix conf\_mat = confusion\_matrix(y\_test, y\_pred) # Print the confusion matrix print(conf\_mat) and import seaborn as sns import matplotlib.pyplot as plt # Use the model to make predictions on the test data y\_pred = np.argmax(model.predict(test\_dataset), axis=1) # Create the confusion matrix conf\_mat = confusion\_matrix(y\_test, y\_pred) # Create a list of class labels class\_names = ['cat,dog '] # Create a heatmap of the confusion matrix sns.heatmap(conf\_mat, annot=True, fmt='d', cmap='Blues', xticklabels=class\_names, yticklabels=class\_names) # Add labels and title plt.xlabel('Predicted Class') plt.ylabel('True Class') plt.title('Confusion Matrix') # Display the plot plt.show()

[you will get version 2]

use a function based approach use pd.getdumies for onehot encoding sentiment column

[additional]

generate me a readme as code with 3 word title 200 character description Brief about dataset Brief about processing techniques Brief about model architecture brief about training and valuation and space for screenshots as per task name

[you will get your readme for github]