PubMed

December 5, 2023

This document contains steps to calculate AI model accuarcies on PubMedQa dataset

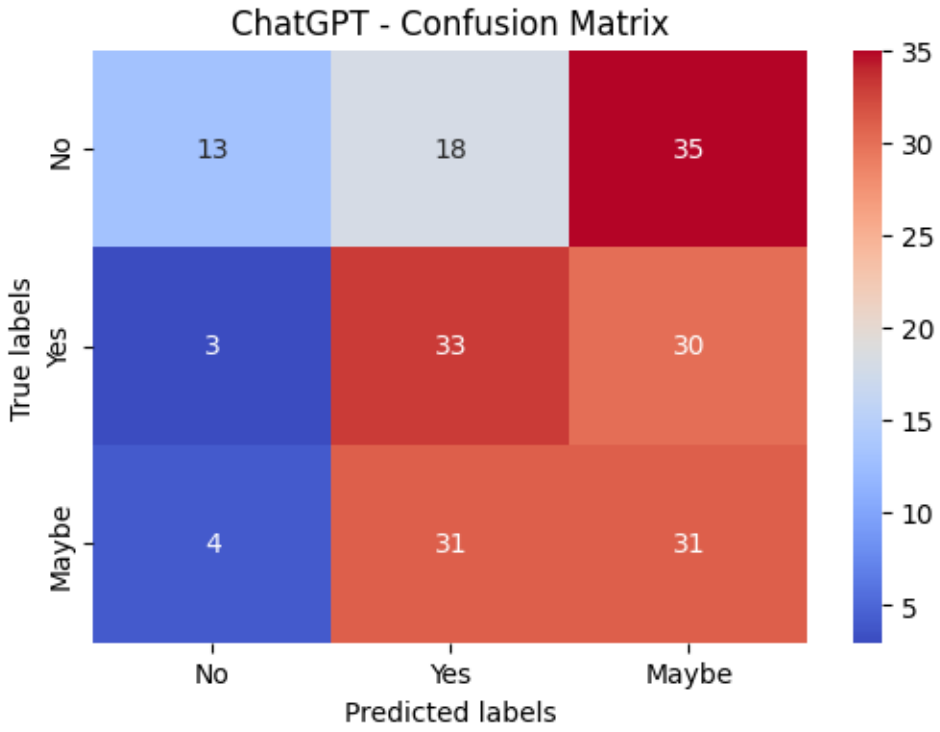
[ ]:
```
#calculating metrices
import pandas as pd
from sklearn.metrics import classification_report, confusion_matrix

# Read the Excel file
df = pd.read_excel('/content/balancedPubMedQA.xlsx')

actual_labels = df['answer']
Chatgpt = df['Chatgpt']
BioGpt  = df['BioGpt']
MetaAI = df['MetaAI']

print("Confusion Matrix of Chatgpt")
print(confusion_matrix(actual_labels, Chatgpt))

# calculating results for ChatGPT prinitng accuracies
report_Chatgpt = classification_report(actual_labels, Chatgpt)
print(report_Chatgpt)

# Confusion Matrxi BioGPT
print("Confusion Matrix of BioGPT")
print(confusion_matrix(actual_labels, BioGpt))

#calculating results for BioGPT prinitng accuracies
report_BioGpt = classification_report(actual_labels, BioGpt)
print(report_BioGpt)

#Confusion Matrix MetaAI
print("Confusion Matrix of MetaAI")
print(confusion_matrix(actual_labels, MetaAI))

# calculating results for MetaAI prinitng accuracies
report_MetaAI = classification_report(actual_labels, MetaAI)
print(report_MetaAI)
```

```
Confusion Matrix of Chatgpt
```

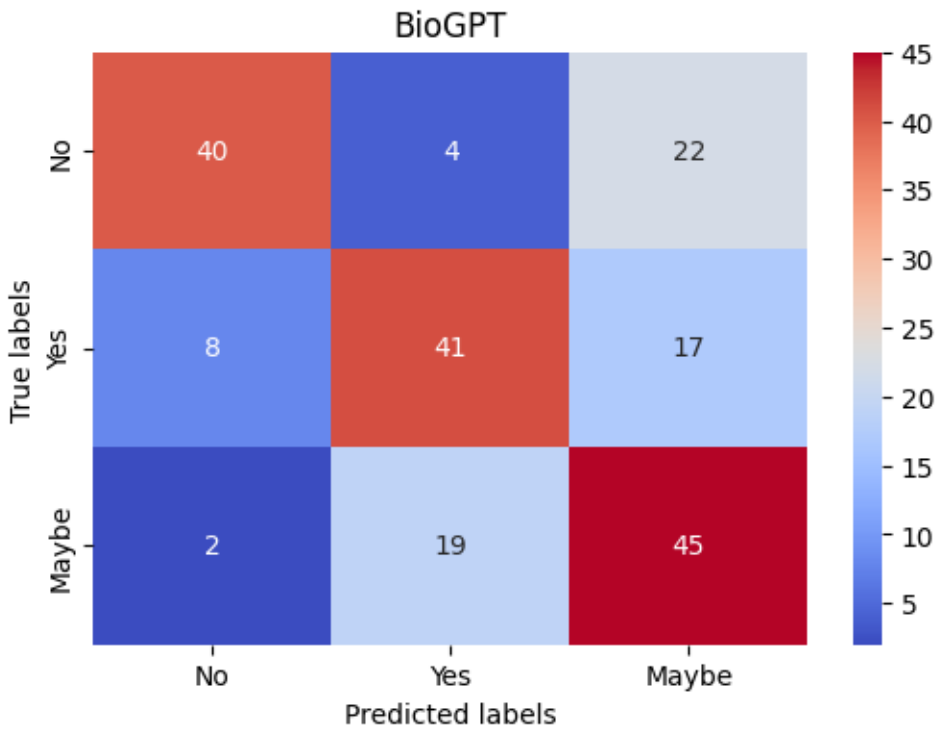
```
[[13 18 35]
 [ 3 33 30]
 [ 4 31 31]]
              precision    recall  f1-score   support

           0       0.65      0.20      0.30        66
           1       0.40      0.50      0.45        66
           2       0.32      0.47      0.38        66

    accuracy                           0.39       198
   macro avg       0.46      0.39      0.38       198
weighted avg       0.46      0.39      0.38       198

Confusion Matrix of BioGPT
[[40  4 22]
 [ 8 41 17]
 [ 2 19 45]]
              precision    recall  f1-score   support

           0       0.80      0.61      0.69        66
           1       0.64      0.62      0.63        66
           2       0.54      0.68      0.60        66

    accuracy                           0.64       198
   macro avg       0.66      0.64      0.64       198
weighted avg       0.66      0.64      0.64       198

Confusion Matrix of MetaAI
[[20 32 14]
 [ 4 52 10]
 [15 35 16]]
              precision    recall  f1-score   support

           0       0.51      0.30      0.38        66
           1       0.44      0.79      0.56        66
           2       0.40      0.24      0.30        66

    accuracy                           0.44       198
   macro avg       0.45      0.44      0.42       198
weighted avg       0.45      0.44      0.42       198
```

[ ]:
```python
#printing Chatgpt confusion Matrix
import seaborn as sns
import matplotlib.pyplot as plt
import numpy as np
```

```
# confusion matrix
conf_matrix = np.array([[13, 18, 35],
                        [3, 33, 30],
                        [4, 31, 31]])

# Labels for the classes
classes = ['No', 'Yes', 'Maybe']

# Plotting the confusion matrix using a heatmap
plt.figure(figsize=(6, 4))
sns.heatmap(conf_matrix, annot=True, cmap='coolwarm', fmt='d')

# Setting labels for x and y axes
plt.xlabel('Predicted labels')
plt.ylabel('True labels')

# Setting x and y axis ticks using the class labels
plt.xticks(np.arange(len(classes)) + 0.5, classes)
plt.yticks(np.arange(len(classes)) + 0.5, classes)

plt.title('ChatGPT - Confusion Matrix')

plt.show()
```

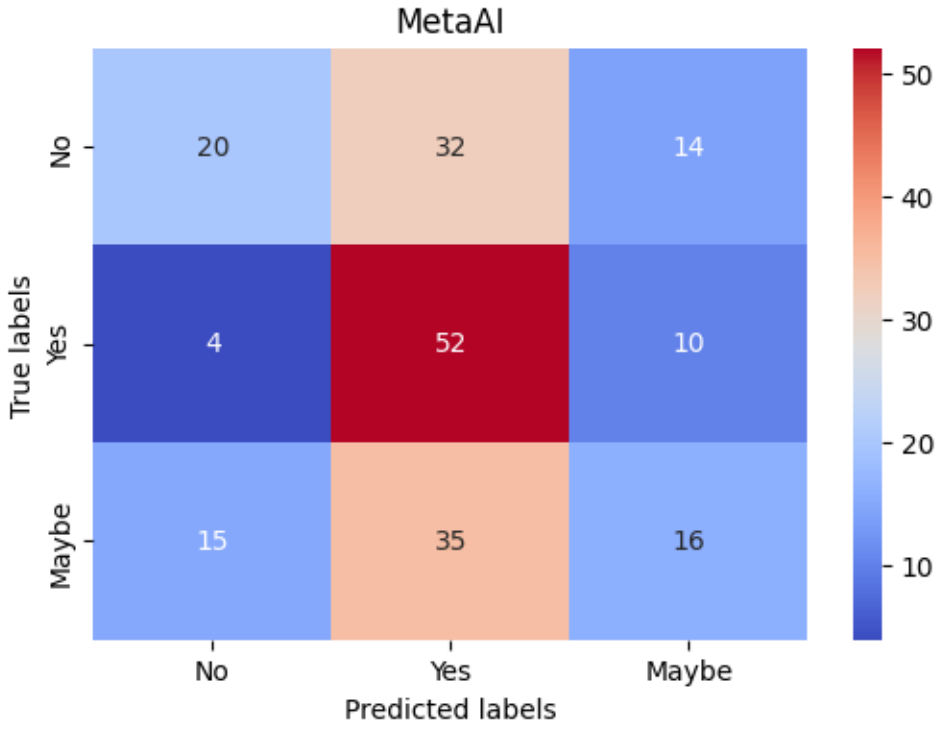


[ ]:
```
#printing BioGPT confusion Matrix
import seaborn as sns
import matplotlib.pyplot as plt
import numpy as np

# confusion matrix
conf_matrix = np.array([[40, 4, 22],
                        [8, 41, 17],
                        [2, 19, 45]])


# Labels for the classes
classes = ['No', 'Yes', 'Maybe']

# Plotting the confusion matrix using a heatmap
plt.figure(figsize=(6, 4))
sns.heatmap(conf_matrix, annot=True, cmap='coolwarm', fmt='d')

# Setting labels for x and y axes
plt.xlabel('Predicted labels')
plt.ylabel('True labels')

# Setting x and y axis ticks using the class labels
plt.xticks(np.arange(len(classes)) + 0.5, classes)
plt.yticks(np.arange(len(classes)) + 0.5, classes)
plt.title('BioGPT')
plt.show()
```



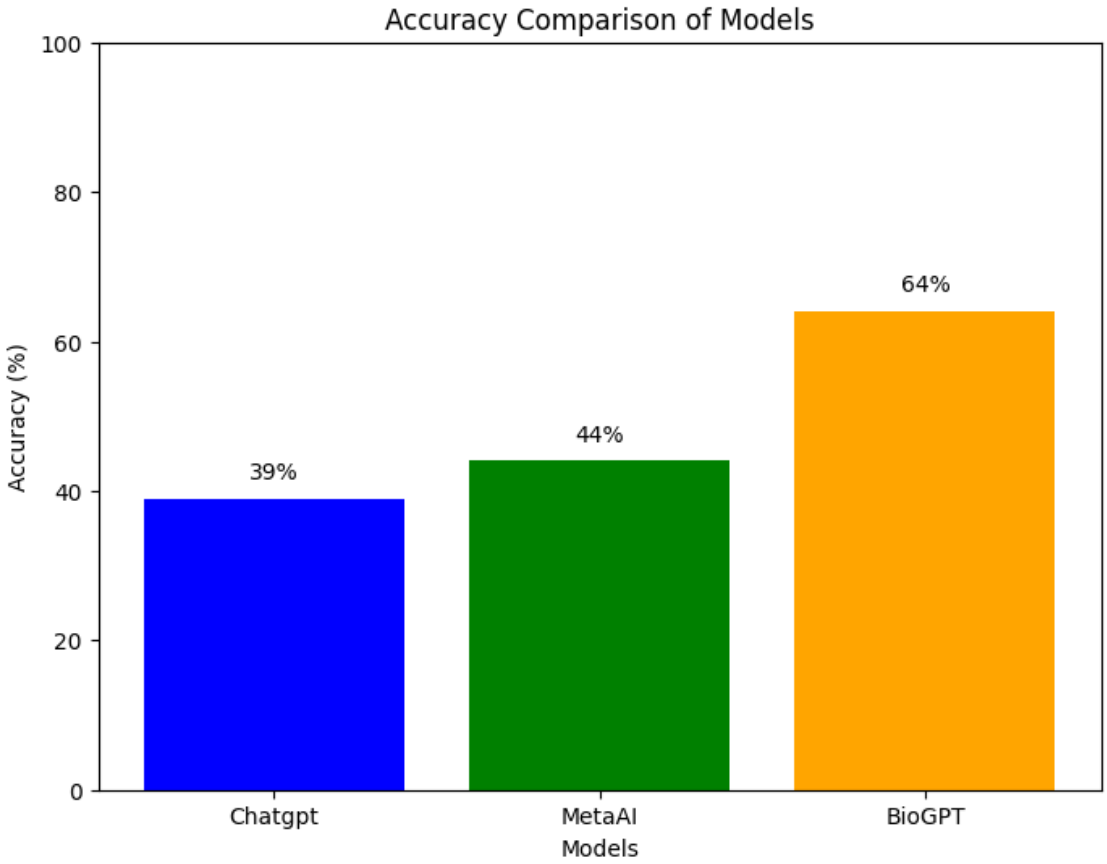
[ ]:
```
#printing MetaAI confusion Matrix
import seaborn as sns
import matplotlib.pyplot as plt
import numpy as np

# Confusion Matrix
conf_matrix = np.array([[20, 32, 14],
                        [4, 52, 10],
                        [15, 35, 16]])

# Labels for the classes
classes = ['No', 'Yes', 'Maybe']

# Plotting the confusion matrix using a heatmap
plt.figure(figsize=(6, 4))
sns.heatmap(conf_matrix, annot=True, cmap='coolwarm', fmt='d')

# Setting labels for x and y axes
plt.xlabel('Predicted labels')
plt.ylabel('True labels')

# Setting x and y axis ticks using the class labels
plt.xticks(np.arange(len(classes)) + 0.5, classes)
```

```
plt.yticks(np.arange(len(classes)) + 0.5, classes)
plt.title('MetaAI')
plt.show()
```



[ ]:
```
#comparing results
import matplotlib.pyplot as plt

models = ['Chatgpt', 'MetaAI', 'BioGPT']
accuracy = [39, 44, 64]  # Accuracy percentages

plt.figure(figsize=(8, 6))
bars = plt.bar(models, accuracy, color=['blue', 'green', 'orange'])

plt.xlabel('Models')
plt.ylabel('Accuracy (%)')
plt.title('Accuracy Comparison of Models')
plt.ylim(0, 100)  # Set the y-axis limit to ensure proper visualization of␣
 ↪percentages

# Adding text on top of each bar
for bar, acc in zip(bars, accuracy):
    plt.text(bar.get_x() + bar.get_width() / 2, bar.get_height() + 2,
             f'{acc}%', ha='center', va='bottom', fontsize=10)
```

```
plt.show()
```



[7]:
```
!jupyter nbconvert --to pdf "/content/drive/MyDrive/Colab Notebooks/PubMed.
↪ipynb"
```

```
[NbConvertApp] Converting notebook /content/drive/MyDrive/Colab
Notebooks/PubMed.ipynb to pdf
[NbConvertApp] Support files will be in PubMed_files/
[NbConvertApp] Making directory ./PubMed_files
[NbConvertApp] Making directory ./PubMed_files
[NbConvertApp] Making directory ./PubMed_files
[NbConvertApp] Making directory ./PubMed_files
[NbConvertApp] Writing 37054 bytes to notebook.tex
[NbConvertApp] Building PDF
[NbConvertApp] Running xelatex 3 times: ['xelatex', 'notebook.tex', '-quiet']
[NbConvertApp] Running bibtex 1 time: ['bibtex', 'notebook']
[NbConvertApp] WARNING | bibtex had problems, most likely because there were no
citations
[NbConvertApp] PDF successfully created
```

```
[NbConvertApp] Writing 110427 bytes to /content/drive/MyDrive/Colab
Notebooks/PubMed.pdf
```

[4]:
```
from google.colab import drive
drive.mount('/content/drive')
```

```
Mounted at /content/drive
```