Midterm Project

Performance of explainer in different text classification models

Zexin Ren

March 22, 2022

Outline

Introduction

Explainer

"Why Should I Trust You?": Explaining the Predictions of Any Classifier

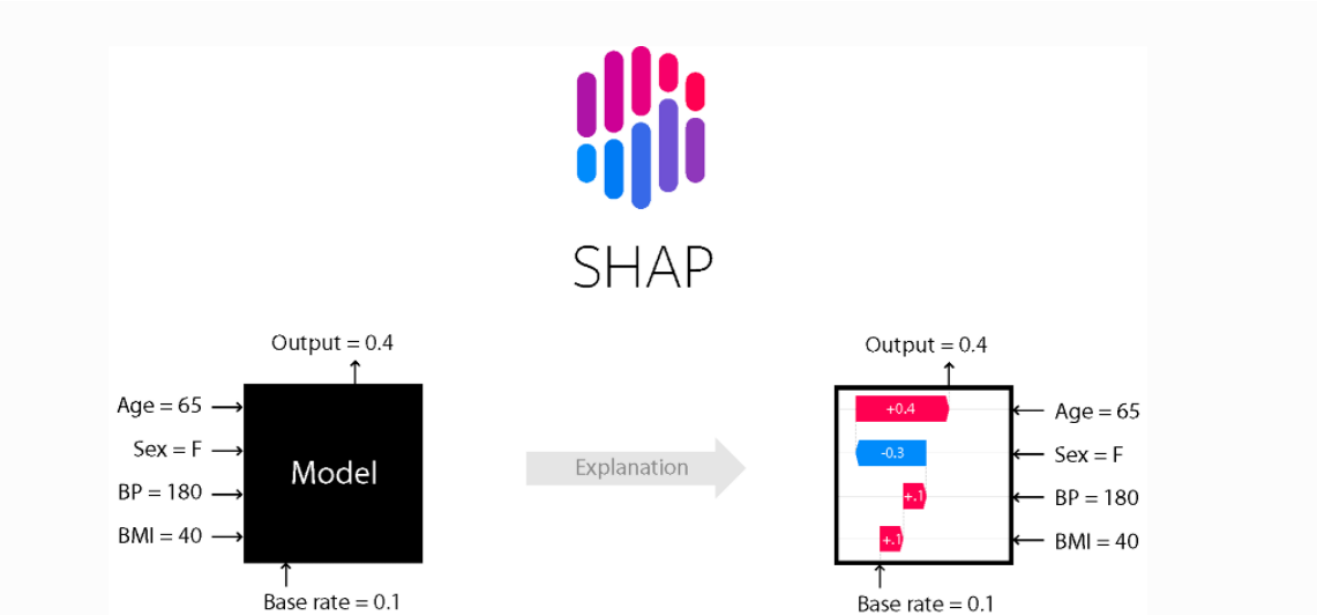


Figure: An example of SHAP explainer

Introduction

Model and Dataset

	Model 1	Model 2
Num. of Labels	2	5
Model Name	distilbert-base-uncased	distilbert-base-uncased
Tokenizer Name	distilbert-base-uncased	distilbert-base-uncased
Dataset	Clinical Statement	Medical abstracts
Test Accuracy	85.5%	77%

Method

Salience Analysis

Figure: A Text Example

Method

Top K Mask

How to test this result?

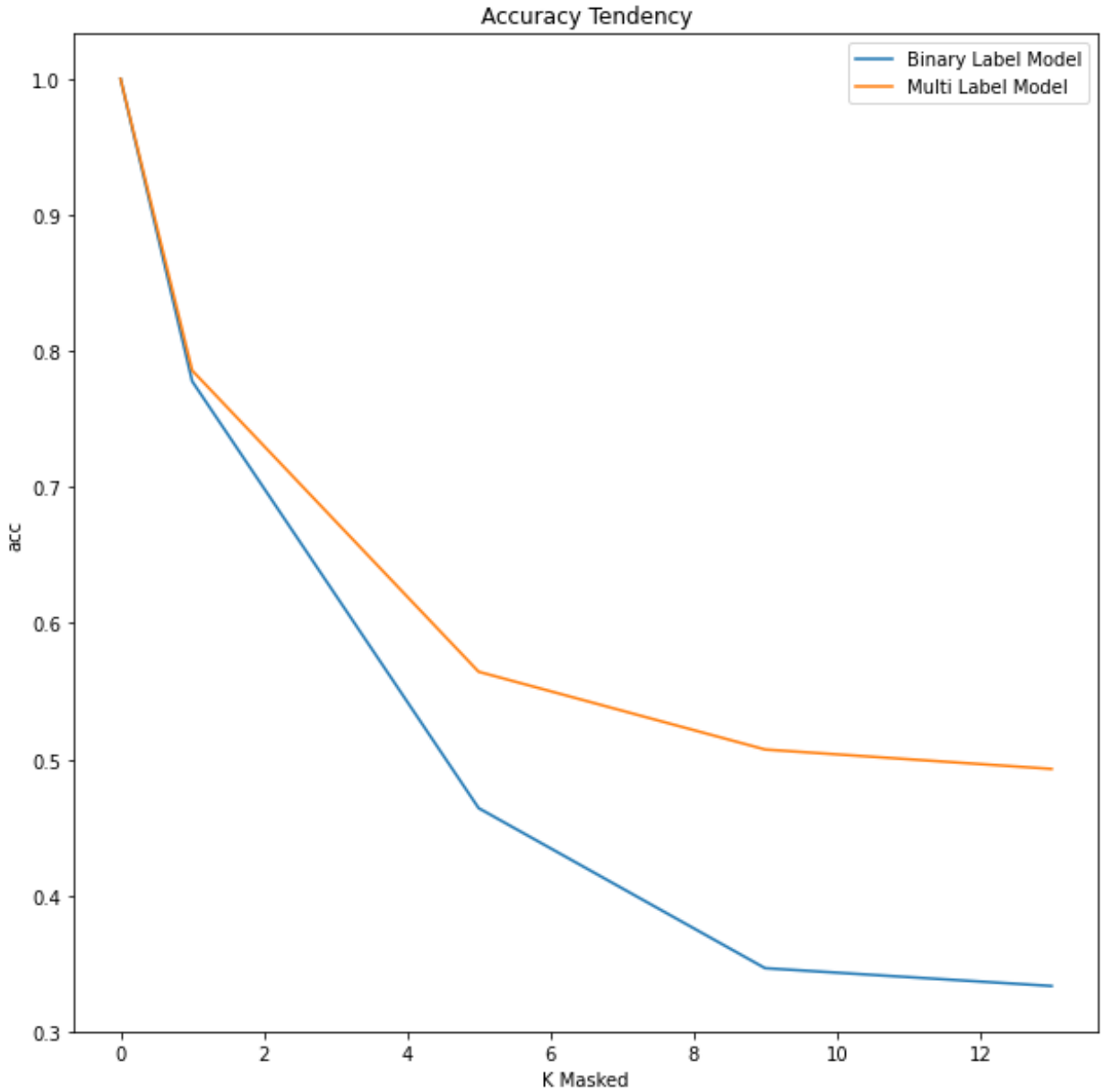
'Right-left disorientation in [UNK] of the [UNK] type. [UNK] [UNK] [UNK] [UNK] -left orientation [UNK] R/L-O) on a confronting subject is more impaired in [UNK] [UNK] dementia [UNK] the Alzheimer type than [UNK] [UNK] [UNK] [UNK] [UNK] infarct [UNK] [UNK] comparable degree of [UNK] . The impairment in R/L-O is independent of aphasia and spatial disorientation.'

Figure: Top K masked

Repeat the same process to all sample on the test set, to see if the accuracy will decrease.

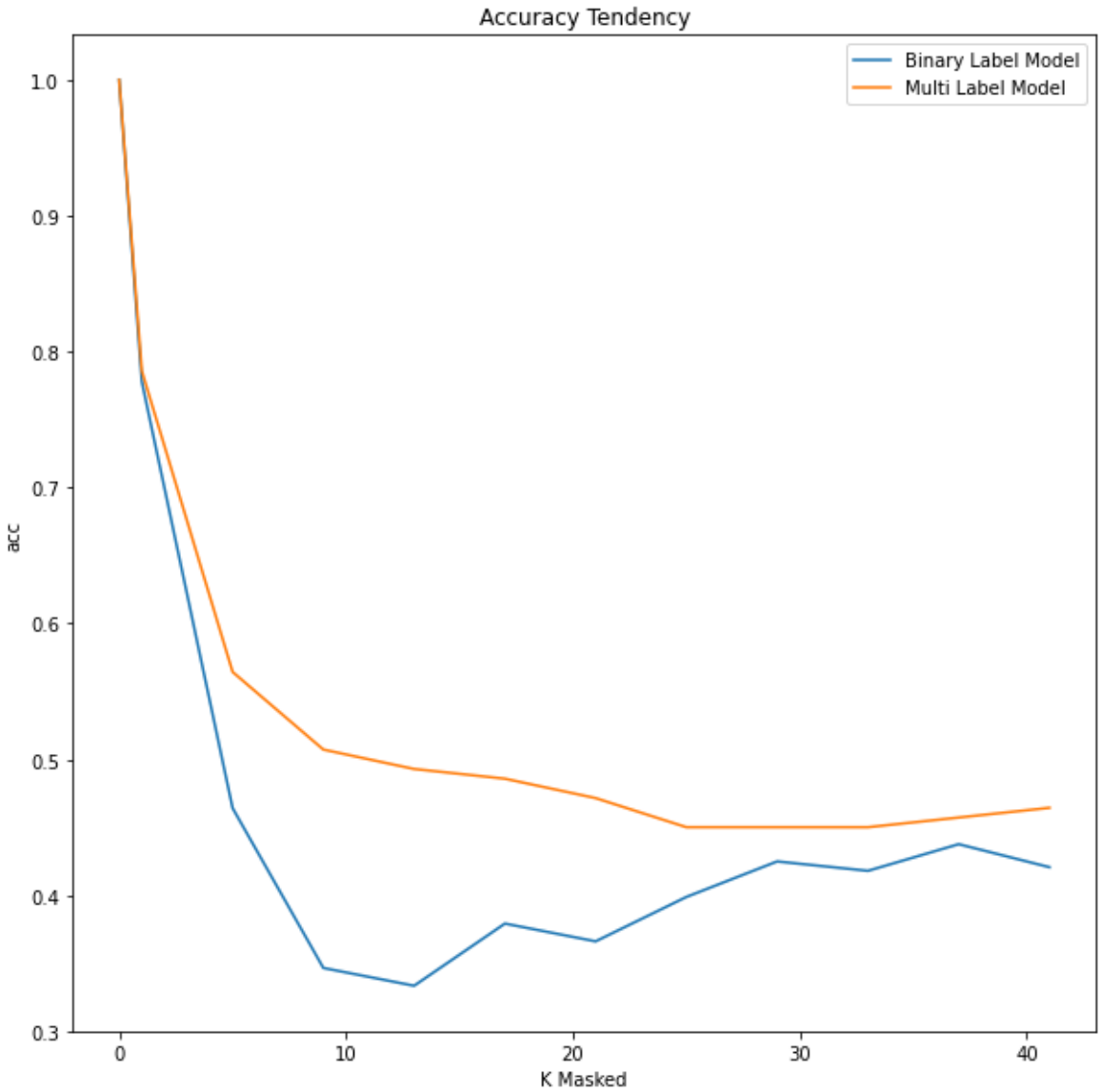
Result

Accuracy Tendency



Result

Accuracy Tendency



Code

Github Link:

https : /github.com/RmmLeo/STAT6289_Homework/tree/main/Midterm%20Project

```python
def mask_top_k(k, pred_label_no_mask, values, returned_tokens):
    """
    masked the k tokens that have the max shap values
    :param k: specify the largest k value
    :param values: shap values
    :param returned_tokens: a list of tokens
    :return: review, which is a str constructed from a list words
    """
    shap_values_0, shap_values_1, shap_values_2, shap_values_3, shap_values_4 = zip(*values)
    if pred_label_no_mask == 0:
        values = shap_values_0
    elif  pred_label_no_mask == 1:
        values = shap_values_1
    elif  pred_label_no_mask == 2:
        values = shap_values_2
    elif  pred_label_no_mask == 3:
        values = shap_values_3
    elif  pred_label_no_mask == 4:
        values = shap_values_4
    # print(values)
    values = np.array(values)
    # ids_top_k = np.argpartition(values, -k)[-k:]
    ids_top_k = (-values).argsort()[:k]
    for idx in ids_top_k:
        # print(idx)
        returned_tokens[idx] = "[UNK] "
    masked_review = "".join(returned_tokens)
    # print(masked_review)
    return masked_review

def predict_label(pipe, masked_review):
    """
    predict the label for the masked_review
    :param pipe: pipeline
    :param masked_review: string
    :return: 0 or 1, indicating the label
    """
    prediction = pipe([masked_review])
    labelstr=prediction[0]['label']
    if labelstr == 'LABEL_0':
        pred_label=0
    elif labelstr == 'LABEL_1':
        pred_label=1
    elif labelstr == 'LABEL_2':
        pred_label=2
    elif labelstr== 'LABEL_3':
        pred_label=3
    elif labelstr == 'LABEL_4':
        pred_label=4
    return pred_label
```

Figure: Code1

Code

```
shap_values_list = []
token_data_list = []
top_k = [1, 5, 9, 13,17,21,25,29,33,37,31]
all_labels =[]
# use GPU
gpu_explainer = shap.Explainer(pipe, tokenizer)
i = 0
for review, label in zip(cor_reviews, cor_labels):
    label=label-1
    print(f"process {i}-th review")
    i += 1
    label4review =[]
    label4review.append(label)
    # to-do: truncate review if len(review)>80
    tokens = tokenizer.tokenize(review)
    if len(tokens) > 80:
        tokens_truncated = tokens[:80]
        review = " ".join(token for token in tokens_truncated)
    pred_label_no_mask = predict_label(pipe, review) # predicted label for review without mask
    label4review.append(pred_label_no_mask)
    shap_values = gpu_explainer([review])
    values = shap_values.values[0] # 2-dim ndarray
    returned_tokens = shap_values.data[0]
    for k in top_k:
        masked_review = mask_top_k(k, pred_label_no_mask, values, returned_tokens) # mask review by the shap values
        predicted_label= predict_label(pipe, masked_review)
        label4review.append(predicted_label)
    # label4review = [True_label, pred_label_without_mask, masked_label_1, masked_label_2, masked_review_3, masked_review_4]
    all_labels.append(label4review)
```

Figure: Code2