MPL PE sem5 ‘22

Vatsal Dhama

Introduction

Problem Statement:

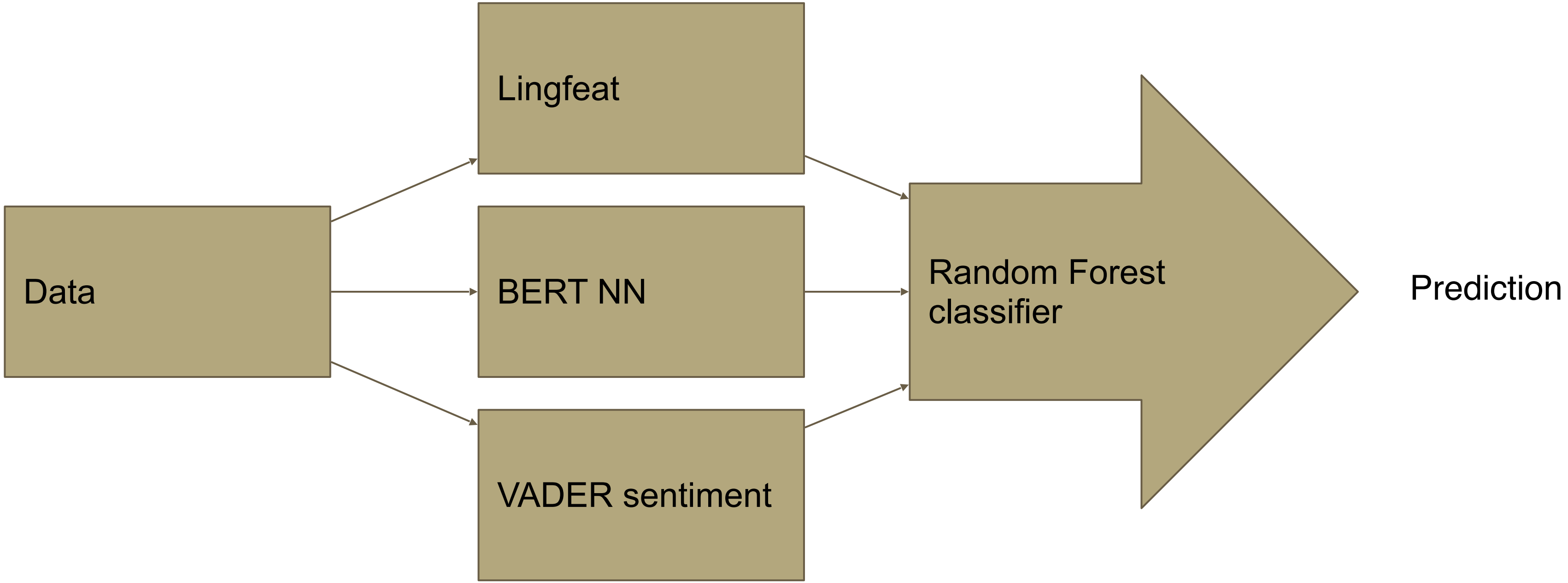
- Detecting self promotion in interviews
- Identifying self promoting statements in an answer
- Score an answer/interview based on self-promotion
- Evaluating the conciseness of the answers
- Explain the prediction or the score
- Provide feedback on how to improve by Generating supporting counter factuals

🟢 My areas of focus

Reading and Study Material

- Went through paper: Language Models as Zero-Shot Planners: Extracting Actionable Knowledge for Embodied Agents - Link
- Random Forest Feature Importance - Link
- DiCE: Link
- SHAP: Link Link
- LIME: Link
- Lingfeat: Link

Model



Lingfeat
Data
BERT NN
VADER sentiment
Random Forest classifier
Prediction

Feature Importance Analysis

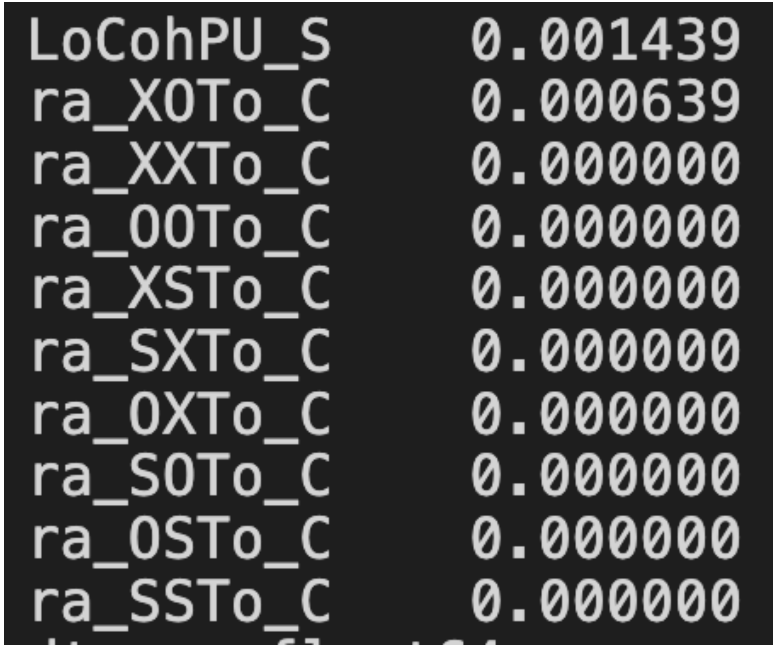
- Took the final random forest classifier model that consisted of 140 features. One of the features “prediction” is the output of BERT model which is used as an input for the random forest classifier. Rest are the lingfeat and sentiment features.
- Used SKlearn’s inbuilt feature importance functionality to get the relative importance of all the features while generating a prediction.
- Took average of 3 model training runs for a better analysis.
- Identified the top 10 most important features and top 10 least important features.

Outputs: most important features

- “prediction”: Prediction from the BERT model
- “to_NoTag_C”: total count of Noun POS tags
- “to_PrPhr_C”: total count of prepositional phrases
- “to_ContW_C”: total count of Content words
- “to_AjTag_C”: total count of Adjective POS tags
- “to_NoPhr_C”: total count of Noun phrases
- “to_SuTag_C”: total count of Subordnating Conjunction POS tags

These 7 features Consistently fell in the top 10 in every model training run

Outputs: 10 least important features



```
LoCohPU_S      0.001439
ra_XOTo_C      0.000639
ra_XXTo_C      0.000000
ra_OOTo_C      0.000000
ra_XSTo_C      0.000000
ra_SXTo_C      0.000000
ra_OXTo_C      0.000000
ra_SOTo_C      0.000000
ra_OSTo_C      0.000000
ra_SSTo_C      0.000000
```

Conclusions

- We can observe the BERT output which was fed as a feature constantly fells as one of the top 5 features in the random forest model. This validates our decision of using BERT output as a feature for improving the overall accuracy of the model.
- The least important features can be dropped off while training. This might improve the model training accuracy incase these features were contributing just in overfitting the random forest model.

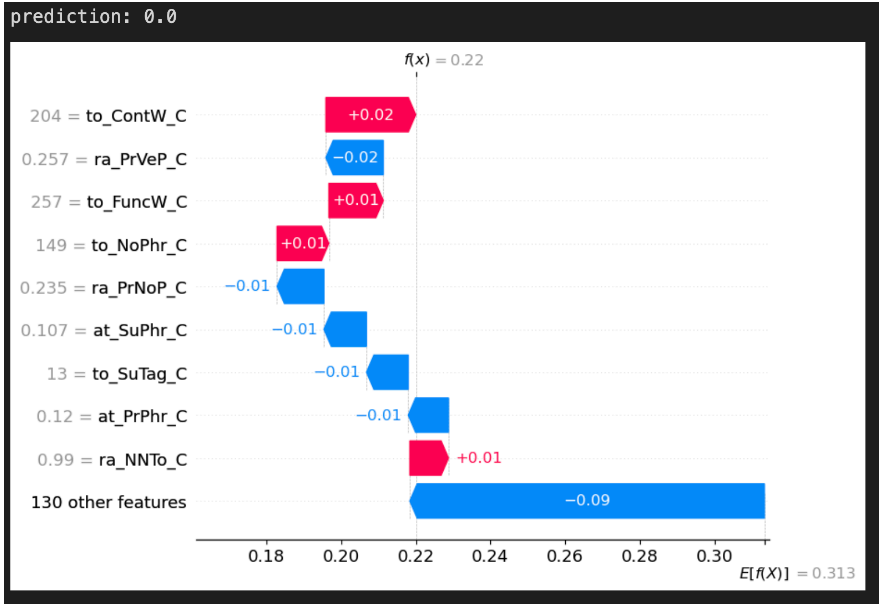
SHAP Explainer

- Set upped the SHAP Explainer to get a explainable visualisation for any prediction generated by our model for a sample.
- e.g. some sample with prediction 0

Features and their relative weights are arranged
In order of their effect on the prediction.

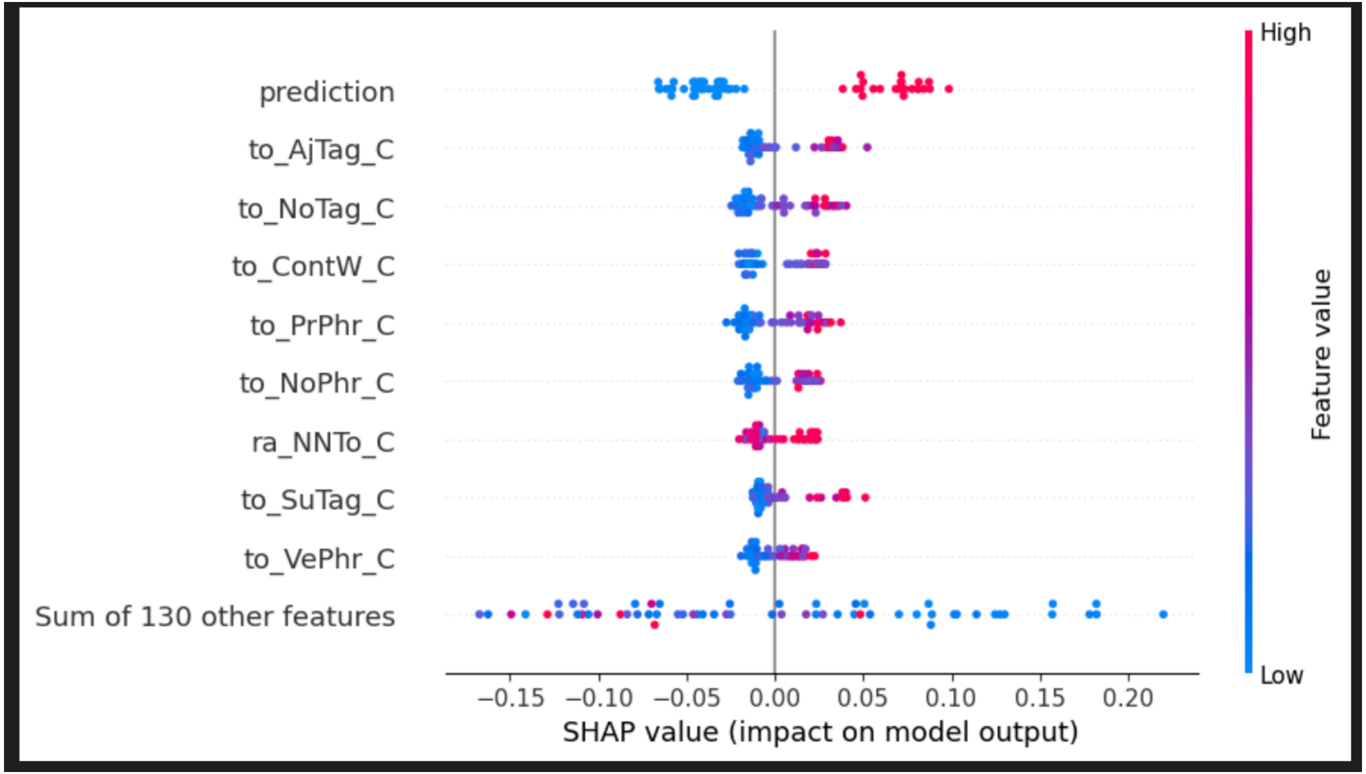
Red means that feature is pushing the prediction to 1.0

Blue means that feature is pushing the prediction to 0.0



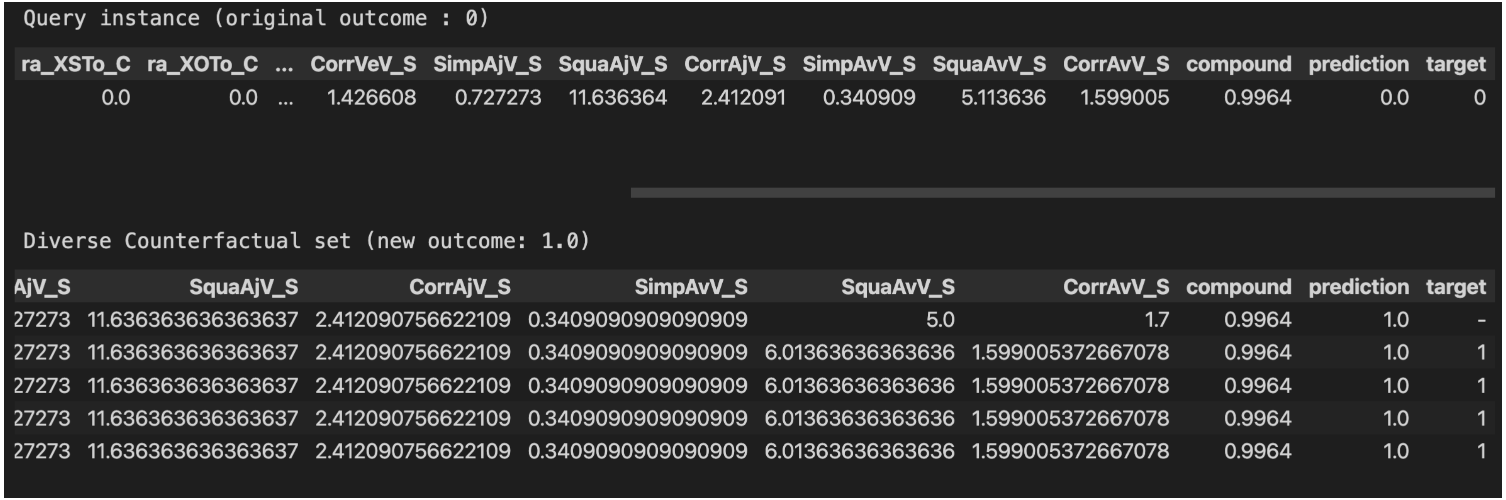
```
prediction: 0.0
```

SHAP most impactful features



DiCE: Diverse Counterfactual Explanations

- Set upped DiCE with our model. This will let us choose number of diverse counterfactual explanations to generate which will help us in getting our targetted prediction. E.g. if my prediction is 0, DicE tells us what features can we improve on with exact values for changing the prediction to 1.
- e.g.



```
Query instance (original outcome : 0)

ra_XSTo_C  ra_XOTo_C  ...  CorrVeV_S  SimpAjV_S  SquaAjV_S  CorrAjV_S  SimpAvV_S  SquaAvV_S  CorrAvV_S  compound  prediction  target
      0.0        0.0  ...   1.426608   0.727273  11.636364   2.412091   0.340909   5.113636   1.599005    0.9964         0.0       0
```

```
Diverse Counterfactual set (new outcome: 1.0)

AjV_S            SquaAjV_S          CorrAjV_S            SimpAvV_S          SquaAvV_S           CorrAvV_S  compound  prediction  target
27273  11.636363636363637  2.412090756622109  0.3409090909090909                5.0                 1.7    0.9964         1.0       -
27273  11.636363636363637  2.412090756622109  0.3409090909090909  6.01363636363636  1.599005372667078    0.9964         1.0       1
27273  11.636363636363637  2.412090756622109  0.3409090909090909  6.01363636363636  1.599005372667078    0.9964         1.0       1
27273  11.636363636363637  2.412090756622109  0.3409090909090909  6.01363636363636  1.599005372667078    0.9964         1.0       1
27273  11.636363636363637  2.412090756622109  0.3409090909090909  6.01363636363636  1.599005372667078    0.9964         1.0       1
```

Conclusions

- We can now use the power of Explainable methods like DiCE and SHAP to create a feedback system for any sample interview to reduce the self promotion.
- Although it is a quantitative analysis, but along with some manual interpretation of features, we can give a detailed feedback for improvement.
- We can also use explainable methods to further remove some non important or non impactful features from the model and improve it.
- Finally We are now able to interpret our models in a better way!

Additional work

- Tried to implement the evaluation of answer conciseness, using the 777 compression algorithm. Later dropped it as a lot of contextual info was required in the text. Also occurrences of compound sentences made the evaluation difficult.
- Assisted Shreyansh and Sarthak with implementing some of the models for self promotion identification.