XAI technique: SHAP

1. Introduction

Due to the lack of transparency, some machine learning (ML) and deep learning (DL) models are black boxes. This is a challenge to improve model performance. To mitigate this problem, Shapley Additive Explanations(SHAP) is an Explainable Artifical Intelligence(XAI) technique that can be applied, as it can explain the importance of every feature to the model. This report will explore the SHAP work principle and advantages and disadvantages. Furthermore, this report will provide SHAP's practical applications for SVM, DNN, and Transformer. Based on research, this report will also propose recommendations of the SHAP technique for the Rail Break Prediction project.

2. Research

This section will explain the SHAP work principles, types, advantages, and disadvantages.

2.1 Working principle

Black box models can be explained through the SHAP technique. Through the SHAP technique, there is a display for feature contrinbution and importance and an explanation for the output.

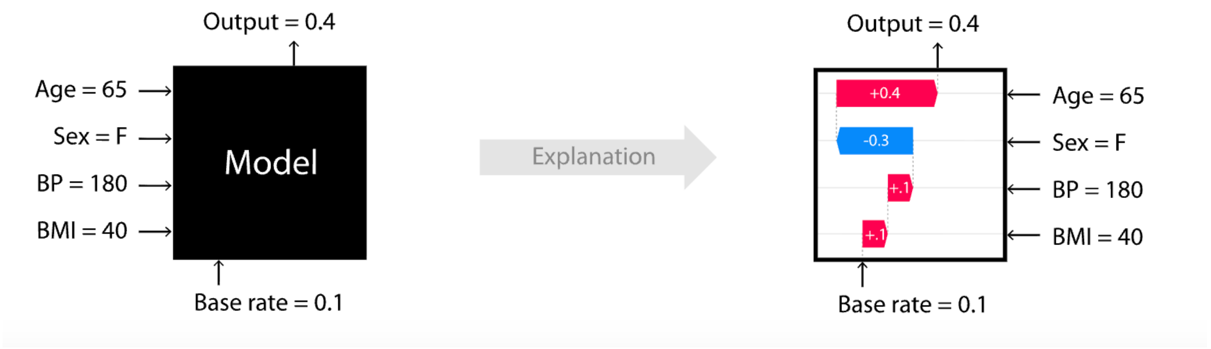


Figure 1: SHAP explanation

As Figure 1 shows [1], the left model is a black box which does not provide the explanation for the output. The right image shows that the base rate is 0.1and there are four input features, including 'Age=65', 'Sex=F','BP=180', and 'BMI=40'. Through SHAP, we found that the 'Age=65' increase the prediction by 0.4, and 'BP=180' and 'BMI = 40' increase the prediction by 0.1, which are positive contributors and assist to increase model prediction. While the

'Sex=F' is a negative contributor, which reduces the prediction by 0.3. The output is to add the base rate and all feature contributions, which is 0.4.

Also, the importance of features is displayed. Through obtaining the absolute number of every fearure, ‘Age = 65’ is the most important feature (0.4). While 'BP=180' and 'BMI = 40' are relatively unimportant features (0.1).

Therefore, the SHAP technique helps us to understand the importance and contribution of features and explain the model output.

2.2 Explainer Types

The SHAP technique applies explainers to explain outputs of models. There are different types of explainers for different types of models.

- Kernal explainer: This explainer can be applied to all models, but it is slower [2].
- Tree explainer: This explainer is for tree and ensemable models.
- Deep explainer: This is a specific explainer for DL models, usually for text [3].
- Gradient explainer: This is also a specific method for DL models, usually for image, audio, and EGC [3].

2.3 Advantages and Disadvantages

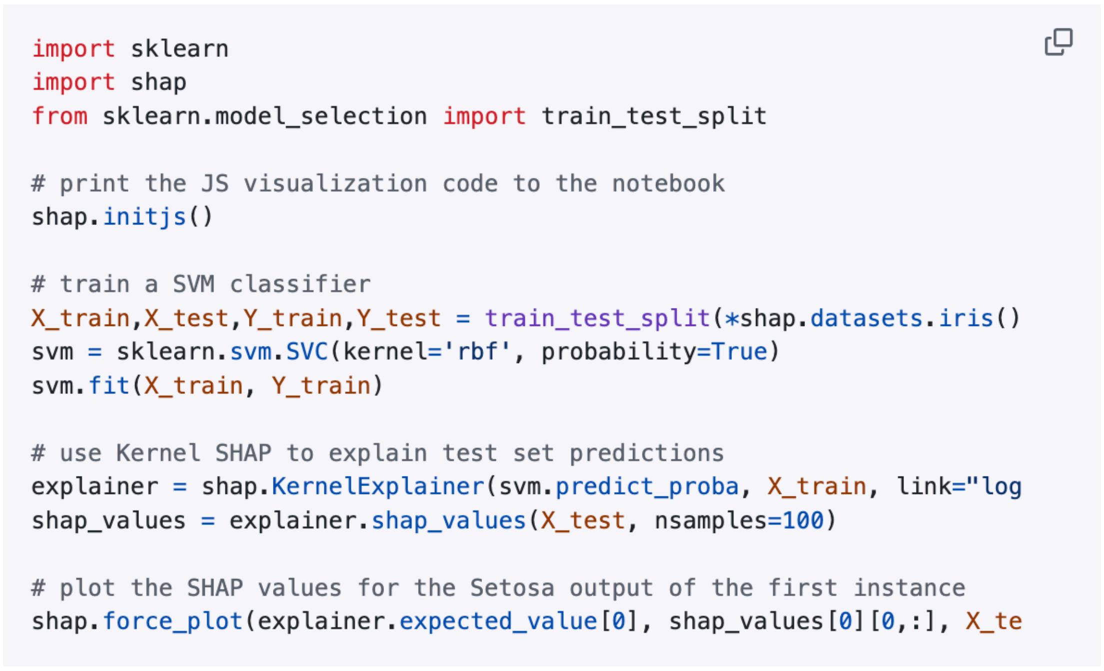
The SHAP technique can improve model transparency and explanation, as feature importance and output calculation can be displayed. Also, this technique can improve prediction accuracy and mitigate the overfitting risk [4]. While the selection of base rate can influence the value of feature contribution. This technique also runs slowly and increase computation complexity, such as the kernal explainer [5].

3. SHAP examples (Code)

Our team has trainned three models in the project, including SVM, DNN, and Transformer. This section provides some code examples from GitHub to show the combination of SHAP and our project models [1].

3.2.1 SHAP and SVM

For the SVM model, Figure 2 is an example of applying shape.KernelExpaliner to explain its output.

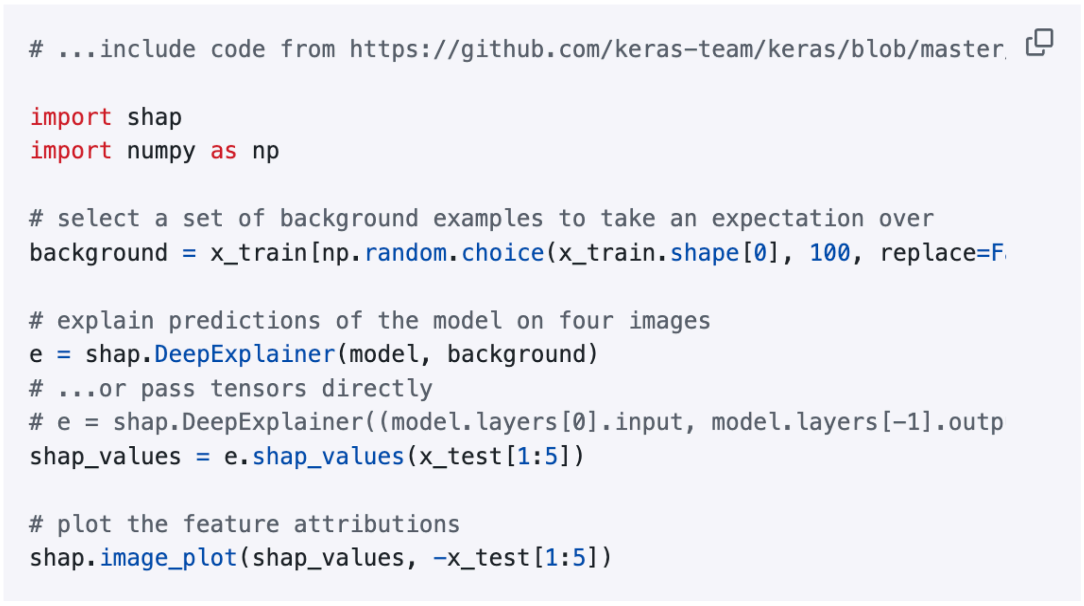


```
import sklearn
import shap
from sklearn.model_selection import train_test_split

# print the JS visualization code to the notebook
shap.initjs()

# train a SVM classifier
X_train,X_test,Y_train,Y_test = train_test_split(*shap.datasets.iris()
svm = sklearn.svm.SVC(kernel='rbf', probability=True)
svm.fit(X_train, Y_train)

# use Kernel SHAP to explain test set predictions
explainer = shap.KernelExplainer(svm.predict_proba, X_train, link="log
shap_values = explainer.shap_values(X_test, nsamples=100)

# plot the SHAP values for the Setosa output of the first instance
shap.force_plot(explainer.expected_value[0], shap_values[0][0,:], X_te
```

Figure 2: SHAP and SVM

3.2.2 SHAP and DNN

Figure 3 is a case to explain the output of the DNN model through the shape.DeepExplainer.

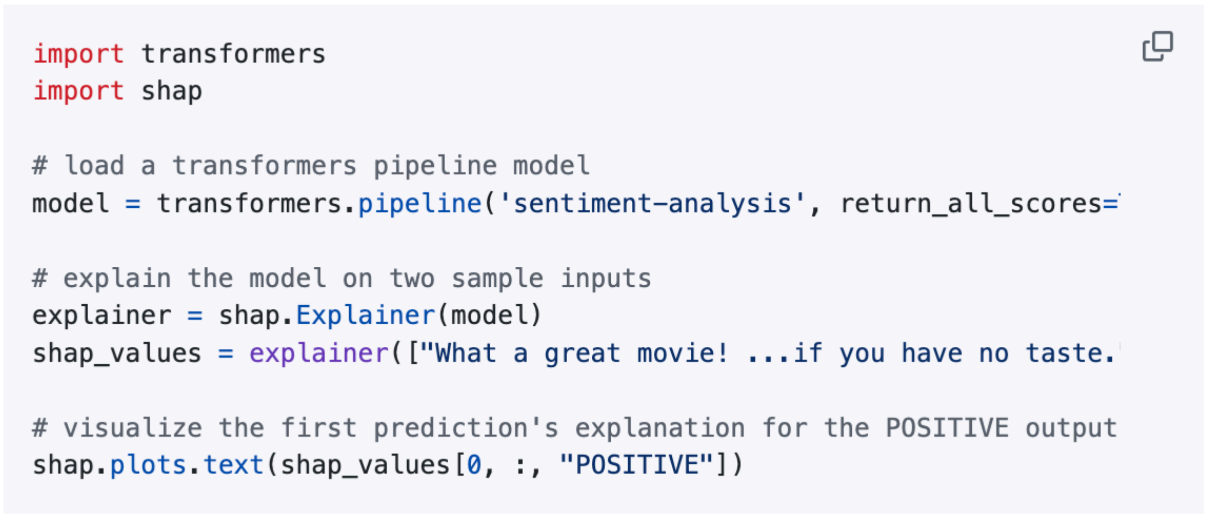


```
# ...include code from https://github.com/keras-team/keras/blob/master

import shap
import numpy as np

# select a set of background examples to take an expectation over
background = x_train[np.random.choice(x_train.shape[0], 100, replace=F

# explain predictions of the model on four images
e = shap.DeepExplainer(model, background)
# ...or pass tensors directly
# e = shap.DeepExplainer((model.layers[0].input, model.layers[-1].outp
shap_values = e.shap_values(x_test[1:5])

# plot the feature attributions
shap.image_plot(shap_values, -x_test[1:5])
```

Figure 3: SHAP and DNN

3.2.3 SHAP and Transformer

shap.Explainer can automatically check the explainer type for the related model. Figure 4 is an example to show the output of the Transformer model through shap.Explainer.



```python
import transformers
import shap

# load a transformers pipeline model
model = transformers.pipeline('sentiment-analysis', return_all_scores=

# explain the model on two sample inputs
explainer = shap.Explainer(model)
shap_values = explainer(["What a great movie! ...if you have no taste.

# visualize the first prediction's explanation for the POSITIVE output
shap.plots.text(shap_values[0, :, "POSITIVE"])
```

Figure 4: SHAP and Transformer

4. Recommendations

Based on the above research and current project situation, it is recommended that applying the SHAP technique to our models. This is because our three models have overfitting problems, which can be mitigated through the SHAP technique. Furthermore, this technique has a wide range of applications, which can be used for three models in the project. Also, this technique facilitates us to understand the importance and contribution of every feature to model performance to improve feature selection. In code, we can directly use shape.Explainer for three models, which can automatically use the suitable explainer for every model.

5. Conclusion

In conclusion, the SHAP is an XAI technique to display feature importance and contribution and explan the model output. This report introduces the principles, explainer types, and strengths and weaknesses of SHAP. Moreover, this report provides examples of a combination of SHAP with SVM, DNN, and Transformer. Based on research and real examples, this report provides the feasibility of SHAP for this project.

References

[1] S. Lundberg *et al.*, “shap/shap,” *GitHub*, Aug. 17, 2023. https://github.com/shap/shap

[2] E. Mosca, F. Szigeti, S. Tragianni, D. Gallagher, and G. Groh, “SHAP-Based Explanation Methods: A Review for NLP Interpretability,” *ACLWeb*, Oct. 01, 2022. https://aclanthology.org/2022.coling-1.406

[3] Jeya Vikranth Jeyakumar, J. Noor, Y.-H. Cheng, L. Garcia, and M. Srivastava, “How Can I Explain This to You? An Empirical Study of Deep Neural Network Explanation Methods,” *Advances in Neural Information Processing Systems*, Dec. 2020, Available: https://www.researchgate.net/publication/351903436_How_Can_I_Explain_This_to_You_An_Empirical_Study_of_Deep_Neural_Network_Explanation_Methods/citations

[4] Corne van Zyl, X. Ye, and R. Naidoo, “Harnessing eXplainable artificial intelligence for feature selection in time series energy forecasting: A comparative analysis of Grad-CAM and SHAP,” *Applied energy*, vol. 353, pp. 122079–122079, Jan. 2024, doi: https://doi.org/10.1016/j.apenergy.2023.122079.

[5] K. Roshan and A. Zafar, “Utilizing XAI Technique to Improve Autoencoder based Model for Computer Network Anomaly Detection with Shapley Additive Explanation(SHAP),” *International journal of Computer Networks & Communications*, vol. 13, no. 6, pp. 109–128, Sep. 2021, doi: https://doi.org/10.5121/ijcnc.2021.13607.