Models That Explain Themselves: Project Report

Radhika Yadav
yadav1@uni-potsdam.de

Abstract

Explainable Artificial Intelligence refers to the methods used to explain models, their performances, and their results. In this paper, we describe four projects surrounding tasks and models that employ this concept. We perform the task for each of our four chosen datasets and then measure the model performance and the explainability of the results. The first is a tree-based algorithm for classification. The second is a language task with feature attribution analysis. The third is a multi-modal task involving both text and images as inputs. The fourth is also a multi-modal task, however, it involves the use of large language models and understanding their explanatory capabilities. We find that the explainability aspect, especially for language tasks still has a long way to go.

1 Introduction

What is an explanation? When is it considered satisfying? Is an explanation only being understood by the people who make the model enough? Is the trade-off between model performance and model explainability worth it?

Explainable AI is a set of all the methods used to open and explain Machine Learning models, their results, and their logic. Discussing explainable AI also brings up the need to discuss a few general concepts first. Although there is a lack of standard and widely accepted definitions, Clinciu and Hastie (2019) summarizes them quite well. Throughout this paper, we will try to understand and find answers to these questions.

There is a section on the Presentation paper that summarizes the main points of the paper. The next 4 sections describe all the four projects in detail. They have an Introduction subsection, which includes information about the task and dataset. It is followed by the Method, Result, Examples, and Discussion sections, which talk about how the project was carried out, the model performances, some examples, and the explainability aspects. Finally, there is a General Discussion section that encompasses a general view of explainability and the limitations surrounding this work.

2 Presentation paper

A black box model is a model that produces results without giving information about its inner workings or how it reached the given result. Guidotti et al. (2018) is a survey of various methods for explaining such models. Whether being used by scientific communities or by different industries, it is important that black box models and their decisions utilizing user data are completely understood.

Also, these models have been shown to acquire certain biases during training. These can be seen during loan crediting, resume filtering, facial recognition, or even in a question-answering system. Hence, it becomes imperative to generate explanations for the decisions the model makes to bring more transparency to the process.

Guidotti et al. (2018) define a taxonomy by classifying the different methods of explaining black boxes into four categories.

1. The aim of the methods under the **Model Explanation** problem is to explain the working and logic of the entire black box. All of the methods here provide globally interpretable models.

2. The aim of the methods under the **Outcome Explanation** problem is to understand the relation between the data of a record and the predicted output. The methods here provide locally interpretable models. These models are able to explain the prediction only for a specific instance or record.

3. The aim of the methods under the **Model Inspection** problem is to provide a textual or visual representation of how the model works.

4. The aim of the methods under the **Transparent Box Design** problem is to provide a model that is interpretable by itself.

The paper puts a lot of emphasis on the need to explain these models for practical, legal, and ethical purposes. However, it additionally stresses the point that only explaining the models is not enough. The explanations should be interpretable, i.e., they should be comprehensible and easy to understand.

3 Project 1: Tree Algorithms

3.1 Introduction

For this task, we implemented a classification task with a decision tree. We chose the Stellar Classification Dataset 2017 (Accetta et al., 2022), which is a public domain dataset released by Sloan Digital Sky Survey [1].

The aim of the dataset is to classify stars into galaxies (0), quasars (1), and stars (2). The task is based on spectral characteristics such as declination angle, ascension angle, ultraviolet filter, redshift value, etc. Some samples from the dataset can be seen in Table 1.

The dataset has 100,000 rows (samples) and 18 columns (features), though we removed two columns that were not important for the analysis. The first is 'obj_id' which is a unique identification number for each sample. The second is 'rerun_id' which is a number to specify how each image was processed and is the same value for each sample.

3.2 Method

We split the dataset in a 70:30 ratio for training and testing respectively. Then we trained a decision tree model on our data. We tested the model and calculated its accuracy and overall confusion matrix along with the feature importance scores. This is our non-optimized model or the model before hyperparameter tuning.

We repeated the same process for our optimized model after tuning four hyperparameters, namely, 'criterion', 'max_depth', 'max_leaf_nodes', and 'splitter'. The best chosen hyperparameters were:

1. Criterion: 'entropy'
2. max_depth: 4
3. max_leaf_nodes: 9
4. splitter: 'best'

The hyperparameter tuning aims at maximising explainability, as opposed to improving performance which it is most commonly used for. We will see its effects in the Result section.

Additionally, we repeated the training and testing processes for both models after we removed the top three features from the data.

3.3 Result and Feature Importance

All the results are summarised in Table 2, and further explained in this section.

3.3.1 Non-optimized decision tree

The model accuracy before hyperparameter tuning is 96.51%, indicating that the model performed quite well. The confusion matrix can be seen in Figure 8 in Appendix A. When visualized, the tree is not comprehensible. This can be seen in Figure 9 in Appendix A.

Since the main aim of the project was explainability, feature analysis was the natural next step. Especially in the context of decision trees, feature importance values can give insights into the results.

Figure 1 demonstrates the features and their importance scores for the non-optimized model. The five most important features in this case were:

1. 'redshift': a value based on the wavelength increase
2. 'g': green filter value in the photometric system
3. 'u': ultraviolet filter value in the photometric system
4. 'r': red filter value in the photometric system
5. 'i': infrared filter value in the photometric system

To further understand the importance of the features and their effects, we removed the top three features and calculated the accuracy for the non-optimized model again. The accuracy was 79.50%, which is quite a bit less than the initial accuracy.

3.3.2 Optimized decision tree

The model accuracy after hyperparameter tuning is 96.22%. The confusion matrix can be seen in Figure 10 in Appendix A There is a slight decrease in performance but the decision tree became much

[1] https://www.kaggle.com/datasets/fedesoriano/stellar-classification-dataset-sdss17

	obj_ID	alpha	delta	u	g	r	i	z	run_ID	rerun_ID	cam_col	field_ID	spec_obj_ID	class	redshift	plate	MJD	fiber_ID
Sample 1	1237660961327743232	135.68	32.49	23.87	22.27	20.39	19.16	18.79	3606	301	2	79	6543777369295181824	GALAXY	0.63	5812	56354	171
Sample 2	1237670961088168192	39.14	28.10	21.74	20.03	19.17	18.81	18.65	5934	301	4	122	2751763212482406400	STAR	-7.895373e-06	2444	54082	232
Sample 3	1237680272039609088	340.99	20.58	23.48	23.33	21.32	20.25	19.54	8102	301	3	110	5658976714552006656	QSO	1.424659	5026	55855	741

Table 1: Samples from the Stellar Classification dataset

	With all features	Without top 3 features
Non-optimized (before tuning)	96.51%	79.50%
Optimized (after tuning)	96.22%	73.87%

Table 2: Accuracy for both decision tree models with all features and without top 3 features

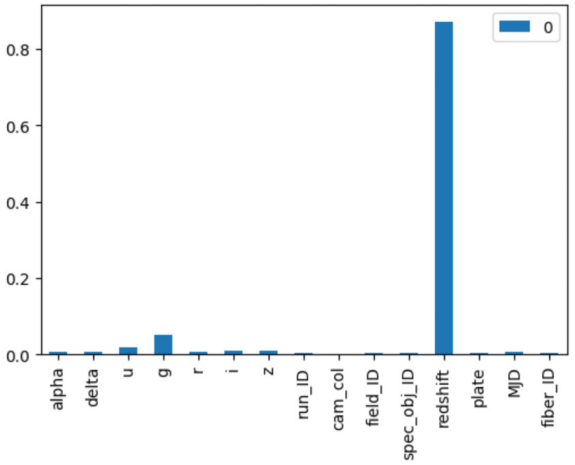


Figure 1: Feature Importance scores for the non-optimized decision tree

more understandable and optimized. Figure 11 in Appendix A proves the same.

Figure 2 demonstrates the features and their importance scores for the optimized model. Only three features were considered important for the tree:

1. 'redshift': a value based on the wavelength increase
2. 'g': green filter value in the photometric system
3. 'spec_obj_id': a unique value that helps to identify similar types of spectroscopic objects

Just as earlier, we removed these top three features and calculated the accuracy for the optimized model again and found that it dropped to 73.87%.

Hence, we can conclude that the 'redshift' and 'g' values are extremely important features for this classification. This is further verified through the decision tree visualizations where 'redshift' is the feature in the root node.

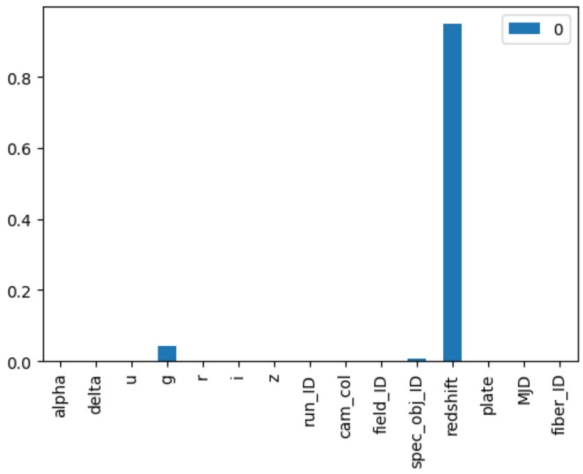


Figure 2: Feature Importance scores for the non-optimized decision tree

3.4 Examples

Figure 3 shows the classification of one sample from the dataset using the decision tree. It is correctly classified as a star (label: 2). The image highlights the path followed while classifying the input.

Some more samples are included in Figures 12 and 13 in Appendix A.

3.5 Discussion

Decision trees are inherently explainable as they provide the logic for their results. The explanation for any input can be found by following a path based on feature values starting from the root node.

Although the trade-off between accuracy and explainability was almost insignificant for our model and data, it is an important consideration.

4 Project 2: Language Task and Feature Attribution Analysis

4.1 Introduction

For the language task, we chose Question Answering. In this task, a model gets a question along with some context as input and has to output an answer. We chose an extractive task, so the answer to the question is a span of tokens that lies within the given context.

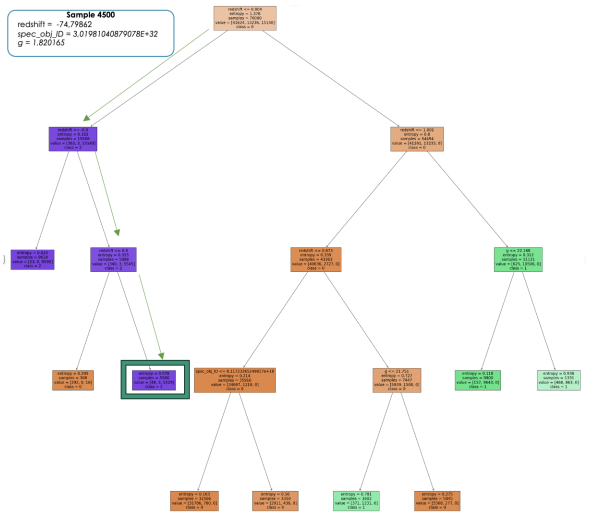


Figure 3: Correctly classified example for project 1

The dataset chosen was the MLQA dataset [2] (Lewis et al., 2019). It is sourced from Wikipedia, contains multiple languages, and also accommodates cross-lingual question answering, but we selected around 12k instances only in English. It contains 4 columns, viz., ID, question, context, and answer. Some samples from the dataset can be seen in Table 3. The answers are in a specific format, it is a dictionary of the starting position of the answer and the actual answer.

4.2 Method

4.2.1 Training and Testing

We chose a BERT base cased model which was fine-tuned on the Stanford Question Answering Dataset (SQuAD). SQuAD also consists of questions and answers from a huge range of Wikipedia articles. Since our chosen model was already finetuned on this dataset, it could have been simply used for our task without further fine-tuning for our dataset. However, we decided to train the model with our data. We split our dataset into train, validation, and test splits in a 50:20:30 ratio respectively.

Then we trained the model using our training and validation data using certain fixed hyperparameters:

1. learning_rate= 2e-5
2. per_device_train_batch_size=16,
3. per_device_eval_batch_size=16,
4. num_train_epochs=3,
5. weight_decay=0.01,

[2]https://github.com/facebookresearch/MLQA

The total evaluation loss is 1.1014. Then we tested the model and calculated an exact match (EM) and an F1 score. These are the most common metrics for evaluating question-answering systems.

As the BERT models only accept 512 tokens, we had to reduce the length of some of the contexts in the data. We limited their length to 1600 characters, which was chosen arbitrarily after experimenting with different limits and checking their effect on the evaluation metrics.

4.2.2 SHAP explanations

We also generated explanations for why a certain word is the start of an answer using SHAP plots. This is an approach based on game theory. First, for the selected sample, we had to calculate the Shapley start values for all tokens in it. These values can be positive or negative. The average of all of these values is then taken as a base value.

Now for each token, the Shapley value of all tokens either increases (if they are positive) or decreases (if they are negative) the base value, thus indicating the chances of this particular token being the start token. The token with the highest value is selected.

This works similarly for generating plots for the end word of an answer.

A SHAP plot can be seen later in the Examples subsection.

4.3 Result

All the scores can be seen in Table 4.

Our model has an exact match score of 58.843%, which is a decent score given that this metric only considers a predicted answer the truth if it exactly matches the actual answer. It does not consider answers that are similar or almost correct.

The model also has an F1 score of 71.81%. An F1 score of 70% or up is generally considered good, so the model performs well.

4.4 Examples

Let us consider the following inputs and outputs and then see the corresponding SHAP plot:

Question: When did development begin in Paris?
Context: Development on Wordfast version 1 (then called simply Wordfast) was begun in 1999 in Paris, France, by Yves ...
Predicted answer: 1999
Actual answer: 1999

	ID	Question	Context	Answer
Sample 1	"a4968ca8a18de16aa3859be760e43dbd3af3fce9"	"Who analyzed the biopsies?"	"In 1994, five unnamed civilian contractors and the widows ..."	{"answer_start": [457], "text": ["Rutgers University biochemists"]}
Sample 2	"7743df4f1208c3755d4ea02b25410bc86a010925"	"How many representatives from FIFA are on the board?"	"The laws of the game are determined by the International Football ..."	{"answer_start": [612], "text": ["four"]}

Table 3: Samples from the MLQA dataset

	Exact Match	F1
Fine-tuned BERT model	58.84%	71.81%

Table 4: Exact Match and F1 scores for our BERT model

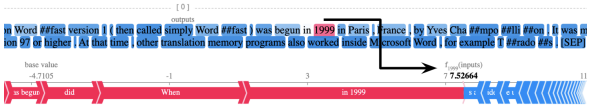


Figure 4: Good prediction example for project 2

Figure 4 shows the SHAP plots for the starting word of the model's answer which is correct here. This particular answer has no ending word.

Some more samples are included in Figures 14 and 15 in Appendix B.

4.5 Discussion

As mentioned earlier, we had to shorten some of the contexts which resulted in a loss of data. In some cases it is possible that the actual answer got removed, hence making it impossible for the model to answer the question.

In terms of explainability, SHAP generates scores for starting and ending words but why a certain word has a higher score to be considered as the starting word is unclear. One can make inferences but the plots themselves do not provide enough details. The overall documentation also does not seem adequate, as we had to look into multiple external sources to understand the meaning and logic of the plots.

5 Project 3: Multimodal Task and Feature Attribution

5.1 Introduction

For the multimodal task, we chose a classification task where the aim was to classify a piece of clothing based on whether it was targeted towards men (0) or women (1).

We used the Fashion Product Images Dataset [3], which contains around 45k samples. Each sample has an image of the product and ten features such as 'id', 'masterCategory', 'subCategory', 'season', 'usage', 'productDisplayName' etc., describing the product. Some samples from the dataset are shown in Table 5.

5.2 Method

We selected around 2k samples by choosing only the 'apparel' from the 'masterCategory' and not accessories, shoes, etc.

We extracted features and their importance scores from the text and image data separately using the models mentioned below. Then we combined these features and scores and used them for classification with an XGBoost model. We also plotted a decision tree and performed a feature importance analysis.

5.2.1 Text data

For the text input, we combined the values of the 'productDisplayName', 'season', and 'usage' columns to have a more comprehensive description of the product. Our text model was 'clothing_subcategory_classifier', which is a fine-tuned version of DeBERTa.

The model gave us eight features and their scores for each sample. Some of the features are: Tops, Dresses, Outerwear etc.

5.2.2 Image data

For our image input, we used the provided images directly. We used the ids from the 2k selected samples to get the corresponding images. Our image model was 'vit-base-clothing-leafs-example-full-simple_highres'. It is a fine-tuned version of the Vision Transformer (ViT) model.

The model gave us thirty-one features and their scores for each sample. Some of the features are: Tee, Jersey, Tank, Sweatpants etc.

5.3 Result and Feature Importance

Table 6 contains all the results for this project.

We chose accuracy as a metric for evaluation. The accuracy for just text features is 79.46%, whereas the accuracy for just image features is 88.81%. So the image features alone perform quite well and give quite a higher accuracy than just the text features. When both features are combined,

[3] https://www.kaggle.com/datasets/paramaggarwal/fashion-product-images-dataset

	id	gender	masterCategory	subCategory	articleType	baseColour	season	year	usage	productDisplayName	image
Sample 1	15,790	Men	Apparel	Topwear	Shirts	Navy Blue	Fall	2011	Casual	"Turtle Check Men Navy Blue Shirt"	
Sample 2	26,960	Women	Apparel	Topwear	Shirts	Purple	Summer	2012	Casual	"Jealous 21 Women Purple Shirt"	

Table 5: Some samples from the Fashion Product Images Dataset

	Accuracy
Only text features	79.46%
Only image features	88.81%
Both text and image features combined	89.81%

Table 6: Accuracy scores for XGBoost model with only text features, only image features, and both

their accuracy is 89.81%, which is the highest of all.

The confusion matrix can be seen in Figure 16 in Appendix C.

The top five most important features just for the text data are:

1. dresses
2. sleepwear
3. bottoms
4. activewear
5. swimwear

The top five most important features just for the image data are:

1. Dress
2. Top
3. Trunks
4. Leggings
5. Chinos

The combined feature importance scores for text and image inputs are represented in Figure 5. The top five most important features in this case are:

1. Dress
2. Top
3. dresses
4. Shorts
5. Trunks

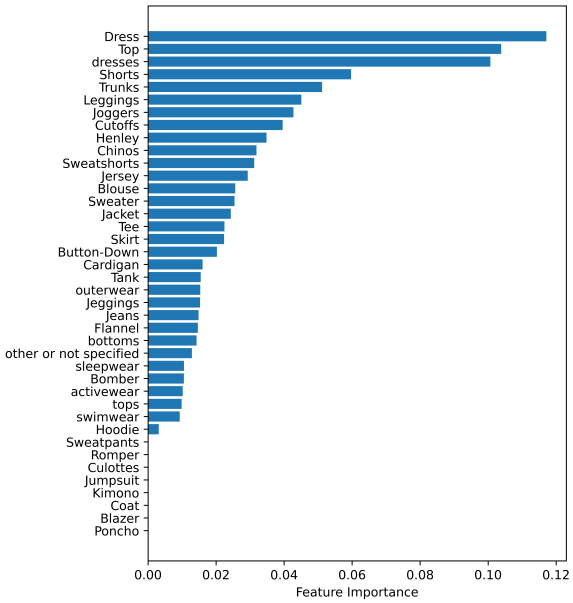


Figure 5: Feature Importance scores for XGBoost classifier with both modalities

Feature names beginning with a capital letter come from images, and the ones beginning with a small letter come from the text. We can observe here that of the top combined features, most of them come from the images. This further adds to the accuracy result, as just the image features had quite a high score. We can also see the top features at the top of the decision tree in Figure 17 in Appendix C.

5.4 Examples

Figure 6 shows a sample correctly classified as "Men". It has the image input on top, followed by the text input, and finally the top 3 features at the end. Features are extracted from these image and text inputs.

Some more examples can be seen in Figures 18 and 19 in Appendix C.

5.5 Discussion

The top features being "Dress", "Top", and "dresses" also make sense because if a clothing item is labeled any of these, then there is a high chance it is targeted towards women. Such features

'Reebok Navy Blue Polo T-shirt Fall Casual'

Feature	Probability
tops	0.994
Tee	0.821
Henley	0.081

Figure 6: Correctly classified example for project 3

can help the model make a clear distinction and hence are extremely helpful.

It can be observed that the image features perform almost just as well as both features combined. One reason for that can be that more features are extracted from the image data and thus the model has more data to learn from.

Extracting useful features is the most difficult part of this task. It is important both for correct classification and later explaining that classification. However, most models work with embeddings which makes feature extraction impossible. That further causes the final result to be in-explainable as we can not trace back the vector representations to see what path the model followed to arrive at a particular answer. So we had to find models that extract features, specifically for our chosen domain.

6 Project 4: Multimodal Task with Prompting Large Language Models

6.1 Introduction

We chose a Visual Question Answering task for this project. It is similar to our previous question-answering task, except here the model gets a question and an image as input and has to output an answer.

We used the A-OKVQA dataset (Schwenk et al., 2022) which is a knowledge-based dataset with over 25k questions. The answers here are not directly found in the input and have to be deduced through commonsense, visual knowledge, physical knowledge etc. Although it is more difficult than simple extractive QA, it is quite a diverse and interesting task.

We used 100 samples from the validation set and considered only the direct answers, not the multiple-choice answers.

Some samples from the dataset can be seen in Table 7. The required images were downloaded from the Common Objects in Context (COCO) dataset (Lin et al., 2014) by using their corresponding 'image_id'.

6.2 Method

Our chosen model was 'google/flan-t5-large' which is a large language model trained on a huge amount of data to perform a variety of natural language tasks in multiple languages.

Since the model cannot work with an image as an input directly, it first has to be represented textually. The dataset already included captions for the images. We gave a question and caption as input. After we got the answer, we gave the model the same question, the same caption, and the generated answer as input and asked the model to generate an explanation for giving this answer.

6.2.1 Prompting strategy

We used multi-shot prompting for answer generation, which means we gave the model some examples to learn from and then our question and context. We used Hakimov and Schlangen (2023) as a reference for the prompt. The prompt can be seen in the box below.

Please answer the question given an image context. Use these examples for help.

Example 1
Image context: A bunch of bananas sitting on top of a wooden table.
Question: What treat can be made with this fruit and ice cream?
Answer: banana split

Example 2
Image context: A pan filled with onions sitting next to a pan of stew.
Question: The reddish-brown food in the further bowl is what type of food?

	split	image_id	question_id	question	choices	correct_choice_idx	direct_answers	difficult_direct_answer	rationales
Sample 1	val	13546	7Q2MqmLkDhRu93vhjMPJXJ	What type of game he played?	[tennis, volleyball, basket ball, skating]	3	['skate board', 'jumping', 'skating', 'skateboarding', 'skating', 'road skiing', 'scatting', 'skating', 'skating', 'skateboarding']	False	["The game is skating.", "This is obvious given he's in a skate park", "He's on a skateboard doing tricks"]
Sample 2	val	579070	9mBjXaxYD4jzJptHzwbeYW	What material are the white cups?	[wood, glass, paper, plastic]	2	['paper', 'paper', 'paper', 'paper', 'plastic', 'paper', 'plastic', 'plastic', 'paper', 'paper']	False	["This is commonly the case. they're often coated these days with b as well.", 'The small white cups are made of a disposable paper material.', 'They are single-use and typically kept in bathrooms.']

Table 7: Some samples from the A-OKVQA dataset

Answer: meat

Image context: A blue and white train is moving on the rails.
Question: What is the primary purpose for this train?
Answer:

For explanation generation, we did zero-shot prompting, which means we did not provide the model with any examples. The prompt can be seen in the box below.

Please generate explanations for the given answer.

Image context: A blue and white train is moving on the rails.
Question: What is the primary purpose for this train?
Answer: to transport people
Explanation:

We tried some different combinations of multi-shot and zero-shot prompting to generate the answers and explanations, but this strategy gave us the best results.

An interesting point to note is that the model would always perform better when the prompt to generate an answer or prediction started with "Please". One can only speculate as to why this might be. It is possible that in the training data, there was an overwhelming number of samples where people asked questions nicely and received help on forums and blogs, etc. and the model learned this behavior.

6.2.2 Annotation scheme

After getting explanations from the model, we also evaluated them manually. For this purpose, we decided on two main metrics: Goodness and Satisfaction. While we were deciding on the metrics and the parameters to annotate them, we had to discard some parameters from the annotation schemes we used. We decided to only use the ones that helped us judge the explanations and not the model. However, even after the initial reduction, we still found some difficulties while actually annotating and had to reconsider some parameters.

Goodness is a simple binary metric to judge whether or not the explanation is 'good' without looking at any other information or context. An explanation has a score of 1 if it is good, and 0 if it is not. For Goodness, we referred to the Explanation Goodness Checklist by Hoffman et al. (2018). It has the following seven parameters:

1. The explanation helps me understand how the [software, algorithm, tool] works.
2. The explanation of how the [software, algorithm, tool] works is satisfying. (G1)
3. The explanation of the [software, algorithm, tool] sufficiently detailed. (G2)
4. The explanation of how the [software, algorithm, tool] works is sufficiently complete. (G3)
5. The explanation is actionable, that is, it helps me know how to use the [software, algorithm, tool]
6. The explanation lets me know how accurate or reliable the [software, algorithm] is.
7. The explanation lets me know how trustworthy the [software, algorithm, tool] is. (G4)

We initially selected parameters 2, 3, 4, 6, and 7 based on how helpful we thought they were with our aim. But while annotating we realized that parameters 6 and 7 are quite similar so we decided to remove the former one. In a way, they both required us to compare the model-generated explanations to the ground truth from the dataset. So we finally had the measures G1, G2, G3, and G4 as indicated above. Our individual annotation scores

	G1	G2	G3	G4
Annotator 1 scores (Radhika)	0.44	0.43	0.43	0.44
Annotator 2 scores (Valentina)	0.44	0.53	0.82	0.52
Average of both annotator scores	0.44	0.48	0.62	0.48

Table 8: Goodness annotation scores for model-generated explanations

and the average for Goodness can be seen in Table 8.

Satisfaction is a metric to understand how useful explanations are and how much people understand an explanation. It is calculated on a 5-point Likert scale. We referred to the System Causability Scale (1-10 below) by Holzinger et al. (2020) and the Explanation Satisfaction scheme (11-18 below) by Hoffman et al. (2018) for this.

1. I found that the data included all relevant known causal factors with sufficient precision and granularity.
2. I understood the explanations within the context of my work.
3. I could change the level of detail on demand.
4. I did not need support to understand the explanations. (S1)
5. I found the explanations helped me to understand causality.
6. I was able to use the explanations with my knowledge base.
7. I did not find inconsistencies between explanations.
8. I think that most people would learn to understand the explanations very quickly. (S2)
9. I did not need more references in the explanations: e.g., medical guidelines, regulations.
10. I received the explanations in a timely and efficient manner.
11. From the explanation, I understand how the [software, algorithm, tool] works.
12. This explanation of how the [software, algorithm, tool] works is satisfying. (S3)
13. This explanation of how the [software, algorithm, tool] works has sufficient detail. (S4)
14. This explanation of how the [software, algorithm, tool] works seems complete. (S5)
15. This explanation of how the [software, algorithm, tool] works tells me how to use it.
16. This explanation of how the [software, algorithm, tool] works is useful to my goals.
17. This explanation of the [software, algorithm, tool] shows me how accurate the [software, algorithm, tool] is.
18. This explanation lets me judge when I should trust and not trust the [software, algorithm, tool]

	S1	S2	S3	S4	S5
Annotator 1 scores (Radhika)	3.54	3.52	2.92	2.91	2.84
Annotator 2 scores (Valentina)	3.34	3.73	2.82	2.95	3.72
Average of both annotator scores	3.44	3.62	2.87	2.93	3.28

Table 9: Satisfaction annotation scores for model-generated explanations

	G1	S3
Inter-Annotater Agreement (IAA) score	0.63	0.27

Table 10: Cohen Kappa-based IAA score for a similar Goodness and Satisfaction parameter

At first, we chose 4, 8, 9, 12, 13, 14, and 17 as our annotation parameters for Satisfaction. During annotating, we found that parameters 9 and 17 were redundant so we discarded them. Finally, we had measures S1, S2, S3, S4, S5 as indicated above. Our individual annotation scores and the average for Satisfaction can be seen in Table 9.

Since G1 and S3 both talk about whether or not "the explanation is satisfying", we also calculated our inner-annotator agreement for those two similar parameters using the Cohen Kappa score. It was 0.63 for the Goodness metric, indicating decent agreement between the two of us. Whereas for the Satisfaction metric, it was 0.27, indicating a low level of agreement. These are also summarised in Table 10.

6.3 Result

Table 11 has the evaluation scores for the model.

One of the metrics used in the A-OKVQA paper is the METEOR score, which is why we chose this metric. The score for the generated answers is 31.42% and the score for the generated explanations is 34.42%.

	METEOR
Answers generated by the model	31.42%
Explanations generated by the model	34.42%

Table 11: METEOR scores for the generated answers and explanations

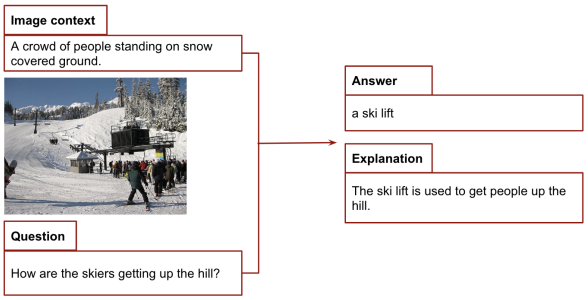


Figure 7: Good prediction example for project 4

6.4 Examples

Figure 7 shows a good answer and explanation prediction by the model.

More examples are included in Figures 20 and 21 in the Appendix D

6.5 Discussion

The captions for the images did not include enough information. Most of them described the images vaguely and did not contain details about the background, the colors, etc. So the model did not have enough context to work with and answer the question. Some questions in the dataset also required common sense and outside knowledge irrespective of the image captions.

Although we used a fixed annotation scheme, it was still difficult to fully agree with the definition for each parameter among ourselves.

7 General discussion

We started with a decision tree and a classification task. It was the easiest model to explain as it lets you trace the path of its logic. It is a simple model and it is difficult to extend it for more elaborate tasks.

So as we moved on to more complex language tasks, we also chose more complex models. However, the ease of explaining the models went down.

This is a trade-off that forms the basis of Explainable AI. If we are not willing to lose any performance, then we might not be able to make our models more understandable, accountable, and transparent. However, an understandable model is useless if it performs poorly and gives good explanations for its wrong results.

Our BERT base model in the second project suffered from being given a shortened version of the contexts as inputs because it can only accept 512 tokens. Adding to the difficulties, the SHAP plots generated scores for the start and end tokens of the answers but these were highly unperceptive. One can never fully understand why a word was chosen over another just by looking at numerical scores.

The third project also employed a decision tree and hence the explainability aspect became exponentially easier. However, the challenge was to extract good features for the decision tree model. We found good models to extract features from our input data but this is not always the case for all domains. Therefore our tools have to extend to work with more models and data.

In the final project, we had different results for the Large Language model's answers based on different prompting strategies. We considered different examples while multi-shot prompting, we tried changing the number of examples, and we tried different ways of giving the task instruction. There are many ways of giving a prompt, such as chain-of-thought prompting, open-ended prompting, etc., and the different combinations are always worth exploring as there is no fixed strategy.

There is a strong lack of a uniform and standard annotation scheme for the explanations generated by black box models. But that cannot change unless there is more consistency in defining an explanation in the first place. We started with a few questions and answered some throughout this paper but we are still no closer to agreeing on a simple, universal definition of a 'good explanation'.

References

Katherine Accetta, Conny Aerts, Victor Silva Aguirre, Romina Ahumada, Nikhil Ajgaonkar, N Filiz Ak, Shadab Alam, Carlos Allende Prieto, Andres Almeida, Friedrich Anders, et al. 2022. The seventeenth data release of the sloan digital sky surveys: Complete release of manga, mastar, and apogee-2 data. *The Astrophysical Journal Supplement Series*, 259(2):35.

Miruna A Clinciu and Helen F Hastie. 2019. A survey of explainable ai terminology. In *1st Workshop on Interactive Natural Language Technology for Explainable Artificial Intelligence 2019*, pages 8–13. Association for Computational Linguistics.

Riccardo Guidotti, Anna Monreale, Salvatore Ruggieri,

Franco Turini, Fosca Giannotti, and Dino Pedreschi. 2018. A survey of methods for explaining black box models. *ACM computing surveys (CSUR)*, 51(5):1–42.

Sherzod Hakimov and David Schlangen. 2023. Images in language space: Exploring the suitability of large language models for vision & language tasks. *arXiv preprint arXiv:2305.13782*.

Robert R Hoffman, Shane T Mueller, Gary Klein, and Jordan Litman. 2018. Metrics for explainable ai: Challenges and prospects. *arXiv preprint arXiv:1812.04608*.

Andreas Holzinger, André Carrington, and Heimo Müller. 2020. Measuring the quality of explanations: the system causability scale (scs) comparing human and machine explanations. *KI-Künstliche Intelligenz*, 34(2):193–198.

Patrick Lewis, Barlas Oğuz, Ruty Rinott, Sebastian Riedel, and Holger Schwenk. 2019. Mlqa: Evaluating cross-lingual extractive question answering. *arXiv preprint arXiv:1910.07475*.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. 2014. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pages 740–755. Springer.

Dustin Schwenk, Apoorv Khandelwal, Christopher Clark, Kenneth Marino, and Roozbeh Mottaghi. 2022. A-okvqa: A benchmark for visual question answering using world knowledge. In *European Conference on Computer Vision*, pages 146–162. Springer.

A Project 1 Appendix

This appendix contains all extra information, graphs, and figures for Project 1: Tree Algorithms covered in Section 3.

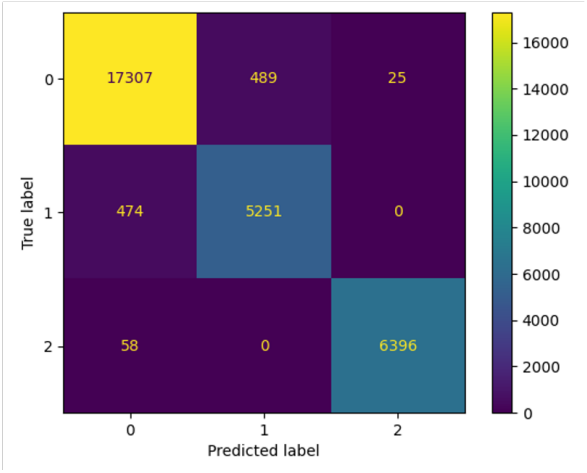


Figure 8: Confusion Matrix for the non-optimized decision tree

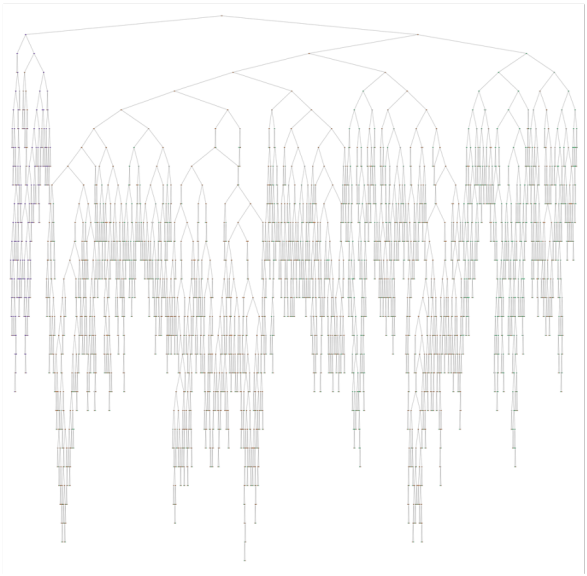


Figure 9: Decision Tree for the non-optimized model

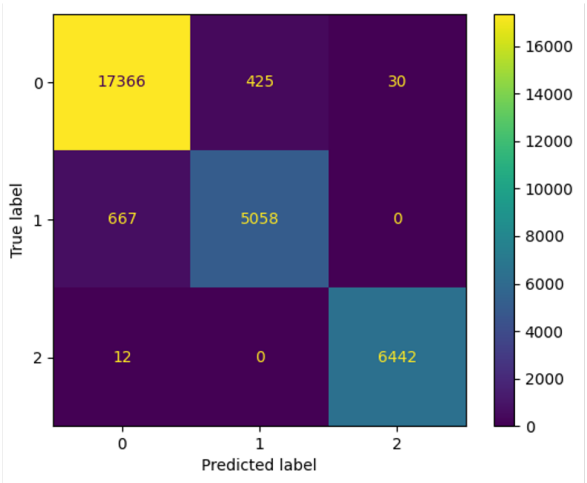


Figure 10: Confusion Matrix for the optimized decision tree

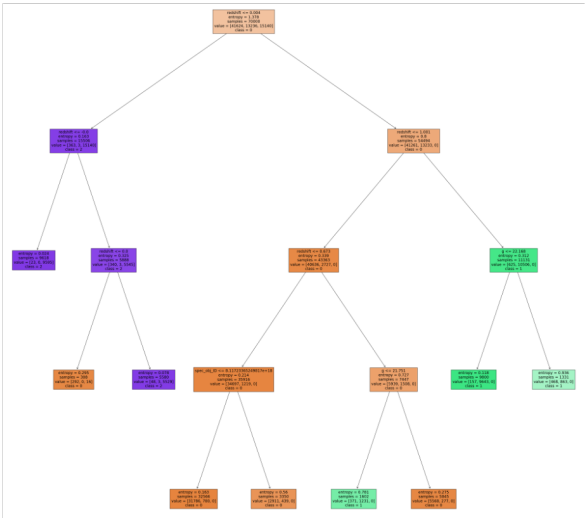


Figure 11: Decision Tree for the optimized model

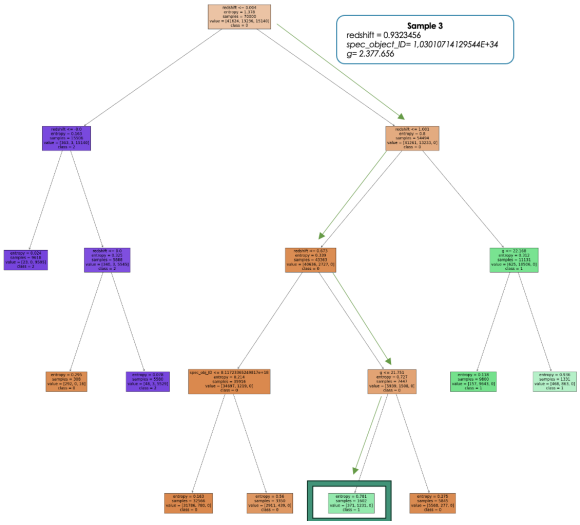


Figure 12: Correctly classified example for project 1

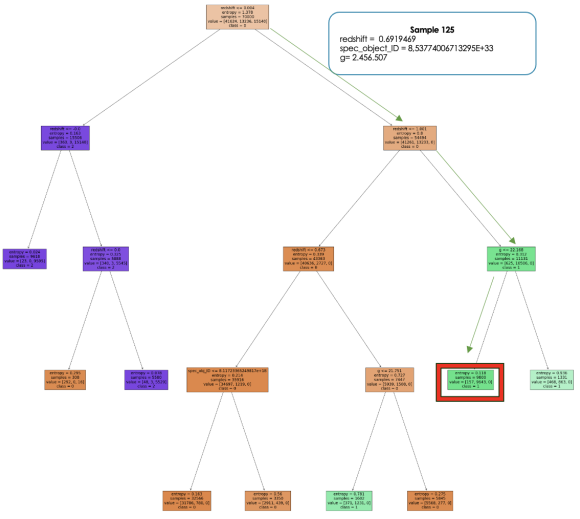


Figure 13: Inorrectly classified example for project 1

B Project 2 Appendix

This appendix contains all extra information, graphs, and figures for Project 2: Language Task and Feature Attribution Analysis covered in Section 4.

Question: Arabia has influenced the language since what century?

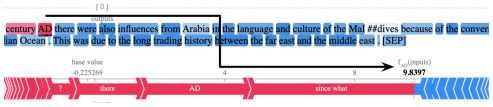
Context: Since the 12th century AD there were also influences from Arabia in the language and culture of the Maldives because of the ...

Predicted answer: 12th century AD

Actual answer: 12th century AD

(a) Inputs and outputs for the example

(b) Start SHAP plot



(c) End SHAP plot

Figure 14: Good prediction example for project 2

Question: What funeral did Nixon attend?

Context: In 1978, Nixon published his memoirs, RN: The Memoirs of Richard Nixon, the first of ten books he was to author in his retirement ...

Predicted answer: its former ally's

Actual answer: Shah of Iran

(a) Inputs and outputs for the example

(b) Start SHAP plot

(c) End SHAP plot

Figure 15: Bad prediction example for project 2

C Project 3 Appendix

This appendix contains all extra information, graphs, and figures for Project 3: Multimodal Task and Feature Attribution covered in Section 5.

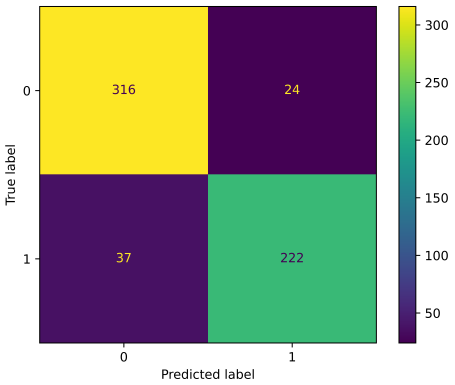


Figure 16: Confusion matrix for XGBoost classifier with both modalities

Figure 17: Decision tree for XGBoost classifier with both modalities

'Lotto White Collared Jacket Fall Sports'

Feature	Probability
tops	0.961
Tee	0.398
Jacket	0.130

Figure 18: Correctly classified example for project 3

'W Printed Purple Kurtas Fall Ethnic'

Feature	Probability
Dress	0.979
tops	0.847
dresses	0.042

Figure 19: Incorrectly classified example for project 3

D Project 4 Appendix

This appendix contains all extra information, graphs, and figures for Project 4: Multimodal Task with Prompting Large Language Models covered in Section 6.

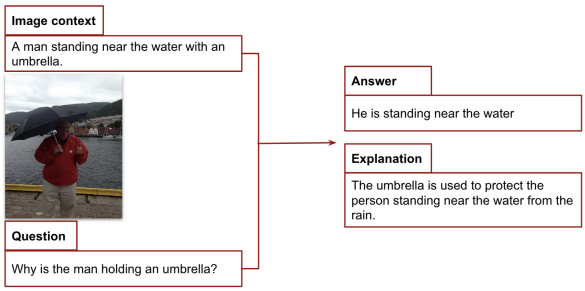


Figure 20: Good prediction example for project 4

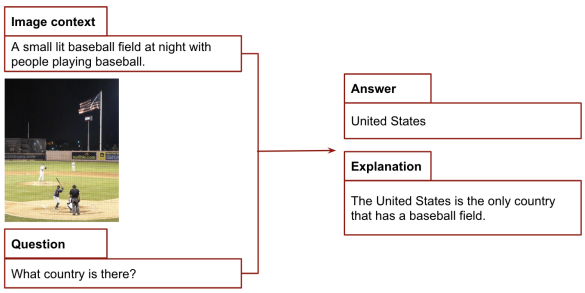


Figure 21: Bad prediction example for project 4