Explainable AI on the AIFB RDF Dataset (Strategy 2)

Bobby Patel, Mohit Jain, and Tanmay Mhatre

University of Paderborn, Germany
{bpatel, jmohit, tmhatre}@mail.upb.de

Abstract. This project explores explainable machine learning on RDF graph-structured data using the AIFB dataset. The RDF data is transformed into a tabular format by extracting literal predicates as features. A Random Forest classifier is trained to predict research group affiliations, and model explanations are provided using SHAP (SHapley Additive exPlanations). We demonstrate both global and local interpretability techniques and evaluate the performance and explainability trade-offs of classical ML models applied to knowledge graph data.

1 Introduction

Knowledge graphs have emerged as a foundational structure for representing rich semantic data. The Resource Description Framework (RDF) enables encoding entities and relationships using subject-predicate-object triples, and has been widely adopted in domains like the Semantic Web, bioinformatics, and academic knowledge bases. Despite their expressive power, RDF graphs present challenges for conventional machine learning (ML) algorithms due to their non-tabular nature.

Explainable AI (XAI) addresses the need for transparency in AI systems, especially when models are deployed in sensitive applications such as healthcare or research evaluation. Understanding not just
emphwhat the model predicts but also
emphwhy, is essential for trust and accountability. This project bridges the gap between RDF-based data and classical ML interpretability tools.

In this mini-project, we adopt a strategy that first flattens the RDF data into a tabular structure by extracting literal features, which are then used to train a Random Forest classifier. We then use SHAP (SHapley Additive exPlanations) to gain insight into both global and local decision processes of the model. Our results show that even simple models like Random Forests, when paired with XAI techniques, can provide transparent and reasonably accurate predictions on RDF datasets.

2 Data Analysis

2.1 Dataset Overview

The AIFB RDF dataset captures metadata related to researchers, their publications, affiliations, and contact details. It is structured as RDF triples, making it a suitable candidate for graph-based ML or symbolic reasoning.

- **Triples:** 8,277 RDF triples
- **Entities:** 828 unique subjects (e.g., persons, organizations)
- **Predicates:** 45 unique predicates including name, email, phone, title, etc.
- **Literals:** Only literal predicates were used for modeling to simplify interpretation.

2.2 RDF to Tabular Conversion

Using the `rdflib` Python package, we extracted only those triples where the object is a literal (e.g., strings, numbers). Each person in the dataset was mapped to a row, with each literal predicate becoming a column.

Missing values were filled with `NaN`, and categorical features were one-hot encoded. This step enabled compatibility with traditional ML models such as Random Forests.

2.3 Train-Test Split

Training and test splits were provided as TSV files:

- `aifb-train.tsv`: 160 labeled training instances
- `aifb-test.tsv`: 36 labeled testing instances

Target labels refer to research group URIs, which were mapped to readable names during post-processing.

3 Model Training and Evaluation

3.1 Preprocessing

The feature matrix underwent several preprocessing steps to ensure compatibility with the learning algorithm:

- Categorical features were one-hot encoded using Scikit-learn's `OneHotEncoder`, applied separately to training data to avoid leakage.
- Missing values introduced by sparse predicates were filled with zeros.
- Feature scaling was not necessary for tree-based models like Random Forest.

3.2 Classifier Description

We selected the Random Forest Classifier from Scikit-learn (`sklearn.ensemble.RandomForestClassifier`) for its interpretability, robustness, and native support for categorical and sparse data. Default hyperparameters were used in this baseline setup. Its ensemble nature helps mitigate overfitting while retaining sensitivity to feature interactions—important in semantic datasets.

3.3 Evaluation Metrics and Observations

The classifier was evaluated using precision, recall, F1-score, and support per class on the test set.

The classifier performs well on majority classes (e.g., Business Info) but shows weak generalization on minority groups, a common issue in imbalanced datasets. This reinforces the importance of class-aware evaluation when applying ML to real-world, heterogeneous data.

3.4 Results

Table 1. Performance Metrics for Random Forest Classifier

Class	Precision	Recall	F1-score	Support
Business Info & Comm. Systems	1.00	0.87	0.93	15
Efficient Algorithms	1.00	0.17	0.29	6
Knowledge Management	0.61	0.92	0.73	12
Complexity Management	0.25	0.33	0.29	3
Overall accuracy			**0.72**	36
Weighted avg F1			**0.70**	36

4 Model Explanation

4.1 Global SHAP Summary Plots

The SHAP summary plots display feature importance globally. The following figures illustrate how various features influenced predictions across the dataset.

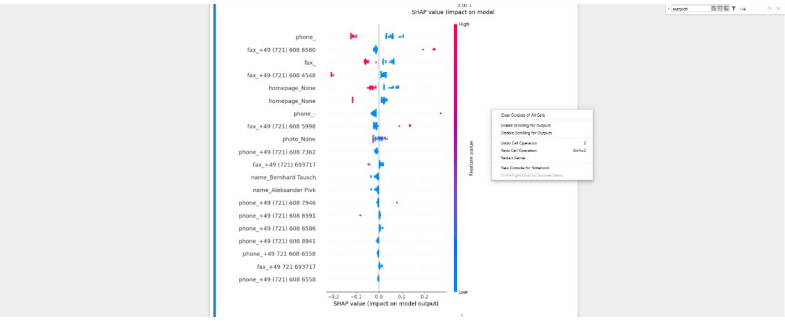

Fig. 1. SHAP summary plot (top features across samples)

	Feature	Mean |SHAP value|
306	phone_	0.041482
89	photo_None	0.032641
143	fax_+49 (721) 608 4548	0.031710
132	fax_	0.030187
85	homepage_None	0.027195
145	fax_+49 (721) 608 6580	0.025499
131	homepage_None	0.022042
149	fax_+49 (721) 693717	0.016000
370	phone_-	0.013978
144	fax_+49 (721) 608 5998	0.013965
358	phone_+49 (721) 608 7362	0.002498
355	phone_+49 (721) 608 6591	0.004840
220	name_Ketut Ngurah Sudharma	0.004699
227	name_Maheshwar Reddy Sathigari	0.004608
171	name_Bernhard Tausch	0.004320
158	name_Aleksander Pivk	0.004322
351	phone_+49 (721) 608 6586	0.003007
348	phone_+49 (721) 608 6557	0.002971
343	phone_+49 (721) 608 6036	0.002781
362	phone_+49 (721) 608 7946	0.002780

Fig. 2. SHAP summary plot (alternate view)

4.2 SHAP Feature Importance Bar Plot

The bar chart ranks features by mean absolute SHAP values. Higher bars denote features with larger average impact.

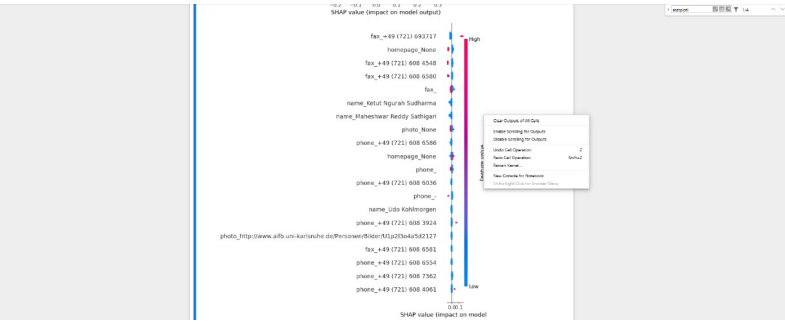

Fig. 3. SHAP mean absolute feature impact values

4.3 Local SHAP Force Plots

To understand the model's behavior for individual predictions, we use SHAP force plots. These visualizations illustrate how each feature contributes to a particular prediction by pushing the output value above or below the expected base value (mean prediction across the dataset).

In a correctly classified instance, positive SHAP values (shown in red) indicate features that pushed the prediction towards the correct class, while negative values (in blue) decreased the model's confidence in that class.

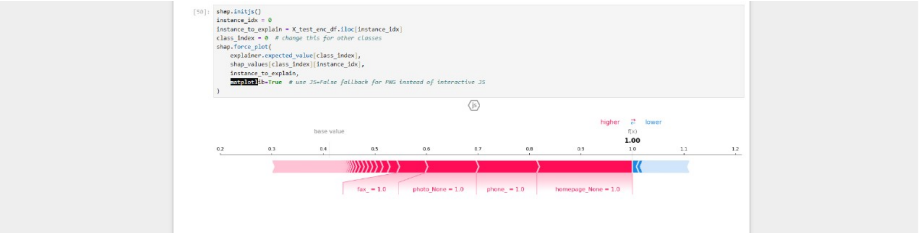

Fig. 4. SHAP force plot for correctly classified instance 1. Key features like `fax_None` and `homepage_None` strongly influenced the prediction.

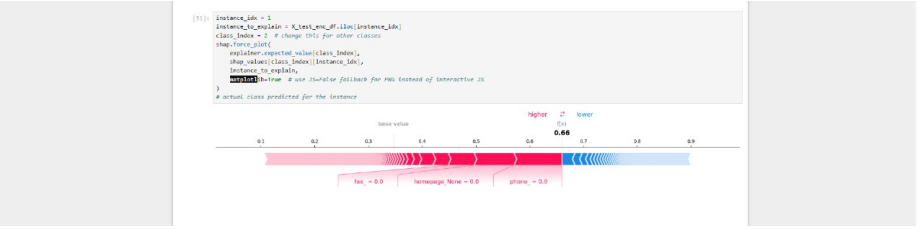

Fig. 5. SHAP force plot for correctly classified instance 2. Features such as `phone` and `title` were decisive for the classification.

These force plots enable instance-level explanations by showing how individual features influenced the final prediction. In both examples above, the classifier made accurate predictions supported by strong feature evidence. Such plots enhance model transparency and help identify which attributes contribute most to the model's decisions.

Incorrect Prediction Example:

Fig. 6. SHAP force plot — misclassified instance

In correct predictions, strong positive feature contributions dominate. For misclassified cases, weak or missing signals mislead the classifier.

5 Conclusion

We demonstrated that converting RDF data into tabular form can enable the use of classical ML models. When combined with SHAP, this pipeline provides both prediction and interpretability. Though Random Forest performs reasonably, performance suffers on minority classes, indicating scope for further enhancement with more balanced or relational features.

6 Contributions of Team Members

- **Bobby Patel:** Data loading, RDF parsing, SHAP visualizations, documentation.
- **Mohit Jain:** Feature engineering, model training, hyperparameter tuning, documentation.
- **Tanmay Mhatre:** Local explanation analysis, documentation, report integration.

Acknowledgements

We thank Dr. Stefan Heindorf for course guidance. We acknowledge OpenAI's ChatGPT for technical explanations during code development.

References

1. Lehmann, J. et al.: AIFB Dataset. Figshare. `https://figshare.com/articles/dataset/AIFB_DataSet/745364/1`
2. Heindorf, S.: Explainable AI course materials, University of Paderborn (2025)