Classification and Explanation with XGBoost and LIME

Viraj Chandugade[1][0000−1111−2222−3333]
cviraj@mail.uni-paderborn.de
https://github.com/virajchandugade and Ruthwik Krishna Bandreddy[1][0000−0003−3909−6065]
rkb2610@mail.uni-paderborn.de
https://github.com/Ruthwik2610

Universität Paderborn, Warburger Str. 100, 33098 Paderborn, Germany

Abstract. This project presents a structured machine learning framework for predicting the research group affiliations of individuals within the AIFB RDF dataset. Rather than relying on graph-based models, we transformed the RDF data into a tabular format by extracting relevant information from entity relationships and attributes. To enhance model efficiency and reduce noise, we applied variance thresholding for feature selection. For classification, we employed the XGBoost algorithm, which is well-suited for structured data and supports multi-class classification. Despite the relatively small dataset, the model achieved a notable accuracy of 86%. To ensure interpretability of the model's predictions, we integrated LIME (Local Interpretable Model-Agnostic Explanations). LIME provides insights into how individual features influence specific classification outcomes. Our approach demonstrates that it is feasible to construct accurate and interpretable models from knowledge graphs without resorting to complex graph neural networks. The results highlight the potential of combining conventional machine learning techniques with explainability tools to support semantic reasoning in knowledge-driven applications.

Keywords: Explainable AI · LIME · AIFB · XGBoost

1 Introduction

To make the data suitable for classification, we first preprocess the RDF triples into a tabular format by extracting relevant features from entity-level predicates and relations. Dimensionality reduction is applied using variance thresholding to remove low-information features and reduce noise. This transformed dataset is then used to train an XGBoost classifier for multi-class prediction of research group affiliations. Although XGBoost is known for its strong performance in structured data, it offers limited interpretability, making it difficult to understand the rationale behind its predictions. In the context of semantic applications, where traceability and reasoning are often crucial, such a limitation can hinder trust and deployment. To address this, we incorporate LIME (Local Interpretable Model-Agnostic Explanations), which generates simple, human-readable approximations of the model's behavior around individual predictions. LIME identifies key features that most influence each decision, offering insight into which relations or attributes were decisive in assigning a research group to a person.

Our method balances transparency and accuracy. The model produces encouraging classification results despite the small dataset, and LIME successfully adds an interpretability layer that facilitates analysis, validation, and semantic comprehension. By combining classical machine learning with explainability tools, this project demonstrates a practical and accessible method for handling structured semantic data—particularly in scenarios where model decisions must be both accurate and justifiable.] This project explores the task of entity classification within the AIFB RDF dataset from the Karlsruhe Institute of Technology (KIT), where the objective is to determine an individual's research group affiliation based on structured semantic data.Rich information about entities and their relationships can be found in RDF graphs, but their high dimensionality, sparsity, and implicit relational structure make it difficult to use them directly in machine learning.

To make the data suitable for classification, we first preprocess the RDF triples into a tabular format by extracting relevant features from entity-level predicates and relations. Dimensionality reduction is applied using variance thresholding to remove low-information features and reduce noise. This transformed dataset is then used to train an XGBoost classifier for multi-class prediction of research group affiliations. Although XGBoost is known for its strong performance in structured data, it offers limited interpretability, making it difficult to understand the rationale behind its predictions. In the context of semantic applications, where traceability and reasoning are often crucial, such a limitation

can hinder trust and deployment. To address this, we incorporate LIME (Local Interpretable Model-Agnostic Explanations), which generates simple, human-readable approximations of the model's behavior around individual predictions. LIME identifies key features that most influence each decision, offering insight into which relations or attributes were decisive in assigning a research group to a person.

Our method balances transparency and accuracy. The model produces encouraging classification results despite the small dataset, and LIME successfully adds an interpretability layer that facilitates analysis, validation, and semantic comprehension. By combining classical machine learning with explainability tools, this project demonstrates a practical and accessible method for handling structured semantic data—particularly in scenarios where model decisions must be both accurate and justifiable.

2 Data Analysis

2.1 Dataset Overview

The dataset used in this study consists of three primary tab-separated files:

- aifbfixed$_{c}omplete.n3ContainsanRDFgraphthatcontains29226triples$.completeDataset.tsv$completed$
- trainingSet.tsv A subset of the full data used for choosing valid number of persons.
- testSet.tsv A hold-out subset used exclusively for evaluation.

Each record in the training and test data files comprises the following fields:

- person: A unique URL identifier representing an individual.
- id: A numeric identifier assigned to each individual.
- label_affiliation: A URL identifier indicating the research group or affiliation.

2.2 Data Exploration and Quality Assessment

Structure Before converting into a pandas DataFrame, the raw N3 RDF data was parsed into a Python dictionary. This resulted in a structured dataset with 2829 rows and 47 columns. Missing values (NaN) were handled by replacing them with empty strings to maintain consistency. Since each RDF triple was represented as a link, the .split() method was used to extract readable labels for the column names.

2.3 Categorical and Distributional Analysis

Cardinality From the sampled data in completeDataset.tsv, the following cardinality was observed:

- Total unique persons: 176
- Total unique affiliations: 5

Each individual is associated with exactly one affiliation, confirming a one-to-one mapping suitable for multiclass classification.

Distribution of Affiliations Affiliation frequencies were calculated using value_counts(), revealing an imbalanced class distribution:

- id1instance (http://.../id1instance): 73 occurrences
- id2instance (http://.../id3instance): 28 occurrences
- id3instance (http://.../id2instance): 60 occurrences
- id4instance (http://.../id1instance): 16 occurrences
- id5instance (http://.../id1instance): 1 occurrence

Insight: The affiliation id1instance comprises two-thirds of all instances, indicating significant class imbalance. Additionally, instance 5 is absent from both train and test sets and has only one member, making prediction impossible.

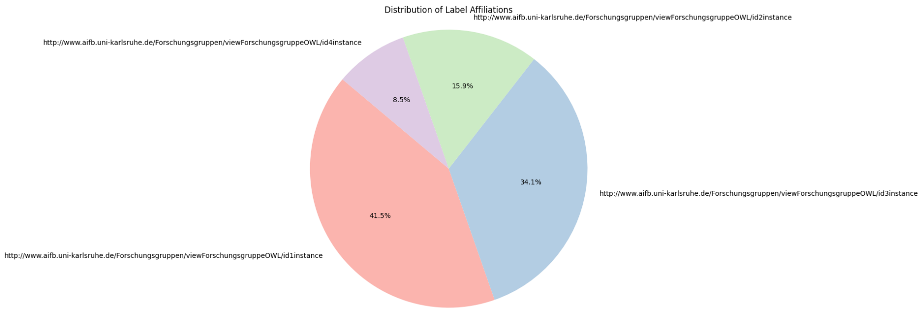


Fig. 1. Enter Caption

3 Model Training and Evaluation

3.1 feature selection and training

Based on the relevant columns holding person data, we filtered the data to keep only legitimate individuals from the train and test sets. Variance threshold was used to select features. There was roughly a 79.5% training and 20.5% test split between the initial training set shape (140, 6879) and the test set shape (36, 6879). Variance thresholding was used to reduce the training set to (140, 486) features. The final training set shape was (140, 482) after additional manual filtering eliminated features that might have led to data leakage. After filtering, a total of 482 features were selected

3.2 Model Selection and Setup

XGBoost was chosen for its ability to efficiently handle multi-class classification problems, such as predicting research group affiliations with multiple categories. Compared to Random Forest, XGBoost often provides better accuracy and faster training due to its gradient boosting approach, which iteratively improves model performance. In our case, this was important given the high dimensionality and sparsity of the RDF-derived features. Furthermore, XGBoost was an effective option due to how it worked with explainability tools like LIME, which let us pick a balance among interpretability and predictive power.

3.3 model evaluation

The XGBoost model achieved a training accuracy of 86.43% and a testing accuracy of 77.78%, indicating good generalization with no major overfitting. The classification report shows strong performance on the majority classes, particularly id1instance and id3instance, with F1-scores of 0.93 and 0.83, respectively.In contrast, performance on minority classes like id2instance was lower (F1-score: 0.25), likely due to class imbalance. The macro average F1-score was 0.64, while the weighted average was 0.75, showing that the model performs better on well-represented classes.

Table 1. Initial Classification Report (Test Set)

Class	**Precision**	**Recall**	**F1-Score**	**Support**
id1instance	1.00	0.87	0.93	15
id2instance	0.50	0.17	0.25	6
id3instance	0.71	1.00	0.83	12
id4instance	0.50	0.67	0.57	3
Accuracy	78.00% (36 instances)			
Macro Avg	0.68	0.67	0.64	
Weighted Avg	0.78	0.78	0.75	

3.4 Improved Results

The second model, trained using only the high-importance features identified by the initial model, achieved an improved accuracy of 86.00%. As shown in Table no. 1, the macro-averaged F1-score increased to 0.76, indicating better overall performance, particularly across underrepresented classes. Notably, minority classes such as class 2 and class 3 achieved F1-scores of 0.91 and 0.80, respectively, demonstrating the effectiveness of focused feature selection in enhancing class-wise prediction quality.

Table 2. Improved Classification Report (Test Set)

Class	**Precision**	**Recall**	**F1-Score**	**Support**
Instance 0	0.84	1.00	0.91	16
Instance 1	1.00	0.25	0.40	4
Instance 2	0.83	1.00	0.91	5
Instance 3	1.00	0.67	0.80	3
Accuracy	0.86 (28 instances)			
Macro Avg	0.92	0.73	0.76	
Weighted Avg	0.88	0.86	0.83	

subsectionSummary

An initial XGBoost model trained on the full feature set achieved a test accuracy of 78%. Based on its feature importance scores, low-importance features were removed, and a second model was trained using only the most relevant features. This refined model achieved a significantly higher test accuracy of 86% and a training accuracy of 97%, indicating improved generalization and reduced noise. Although the 11% gap between train and test accuracy suggests mild overfitting, the overall performance gain confirms the effectiveness of model-driven feature selection.

This strategy was inspired by a practical guide on XGBoost-based feature selection by Dhanya [?]. The results validate the usefulness of this method, particularly in small-scale, structured knowledge graph settings where focusing on informative features significantly enhances classification performance.

4 Model Explanation

In this section, we present an in-depth mathematical and algorithmic description of the LIME (Local Interpretable Model-agnostic Explanations) procedure as applied to our XGBoost classifier for person-affiliation prediction. We begin with the formal LIME objective, proceed through each computational step—perturbation, weighting, surrogate model fitting—and conclude with a detailed case study for test instance #0, including derivations and parameter settings.

4.1 Overview of LIME

LIME seeks to approximate a complex, black-box model $f : R^d \to [0,1]^K$ by a simpler, interpretable surrogate g in the vicinity of a target instance x. For classification, $f(x) = \big(f_1(x), \ldots, f_K(x)\big)$ returns the estimated class probabilities. We focus on the predicted class $k^* = \arg\max_k f_k(x)$.

LIME formulates an optimization problem:

$$g \in \mathcal{G} \arg\min \underbrace{\mathcal{L}\big(f, g, \pi_x\big)}_{local fidelity} + \underbrace{\Omega(g)}_{model complexity} .$$

Here:

- $\mathcal{G}$ is the family of interpretable models (we use sparse linear functions).
- $\mathcal{L}(f, g, \pi_x) = \sum_{z \in Z} \pi_x(z) \big(f_{k^*}(z) - g(z)\big)^2$ measures the weighted squared error between f and g over perturbed samples z.
- $\Omega(g)$ penalizes complexity; for sparse linear models $g(z) = w_0 + w^\top z$, $\Omega(g) = \lambda \|w\|_1$ encourages few nonzero weights.
- $\pi_x(z)$ is a proximity kernel defining locality around x.

4.2 Instance Perturbation

Discretization of Continuous Features Continuous features in x (e.g., numerical encodings of URLs) are first binned into quantiles. Let $\{b_0, b_1, \ldots, b_m\}$ be the bin edges computed on the training data for each feature. The instance x_j falls into bin q if $b_{q-1} < x_j \le b_q$. This yields a simplified representation $\tilde{x}_j \in \{1, \ldots, m\}$.

Generating Perturbations From $\tilde{x}$, LIME generates N samples $\{z^{(i)}\}_{i=1}^{N}$ by independently sampling each feature $z_j^{(i)}$:

$$z_j^{(i)} = \{\tilde{}x_j, withprobabilityp_{orig}, U\{1, \dots, m\}, withprobability1 - p_{orig},$$

where $p_{orig} = \frac{1}{m}$ by default (uniform probability to retain original bin). This yields binary indicator vectors in one-hot form, but for simplicity we treat $z_j^{(i)}$ as bin indices.

4.3 Locality Kernel

Define a distance function $D(x, z)$. For mixed binary and categorical features, we use the cosine distance:

$$D_{\cos}(x, z) = 1 - \frac{x^{\top} z}{\|x\|_2 \, \|z\|_2}.$$

The proximity kernel is then

$$\pi_x(z) = \exp\Big(-\frac{D(x, z)^2}{\sigma^2}\Big)$$

with kernel width σ set to $\sqrt{d}$ by default. This ensures that samples more similar to x receive exponentially higher weight in the surrogate fitting.

4.4 Surrogate Model Fitting

We fit a locally weighted linear model

$$g(z) = w_0 + \sum_{j=1}^{d} w_j \, I\{z_j = \tilde{x}_j\}$$

where $I\{\cdot\}$ indicates whether feature j remains in its original bin. In matrix form, let $Z \in \{0, 1\}^{N \times d}$ be the binary design matrix for the N perturbations. Let $\Phi = \operatorname{diag}\big(\pi_x(z^{(1)}), \dots, \pi_x(z^{(N)})\big)$ be the diagonal weight matrix. We solve:

$$\min_{w_0, w} \, \big(f_{k^*}(Z) - w_0\mathbf{1} - Zw\big)^{\top} \Phi \big(f_{k^*}(Z) - w_0\mathbf{1} - Zw\big) \; + \; \lambda\|w\|_1.$$

Here $f_{k^*}(Z) \in R^N$ is the vector of model predictions on the perturbed samples. This is a weighted Lasso regression, efficiently solvable via coordinate descent.

4.5 Hyperparameters and Computational Details

- Number of perturbations: $N = 5000$.
- Kernel width: $\sigma = \sqrt{d}$, with $d = |features| = 150$.

- Complexity penalty: λ chosen by cross-validation on a small subset to yield approximately 10 nonzero weights.
- Distance metric: cosine distance for high-dimensional binary data.
- Discretization bins: $m = 10$ quantile bins per feature.

The overall time complexity of LIME per explanation is $O(N \times T_f + Nd)$, where T_f is the cost of one model prediction. In our setting, $T_f \approx 10^{-4}$s and $d = 150$, yielding explanations in under 1s.

4.6 Case Study (initial model): Test Instance: Daniel Ried

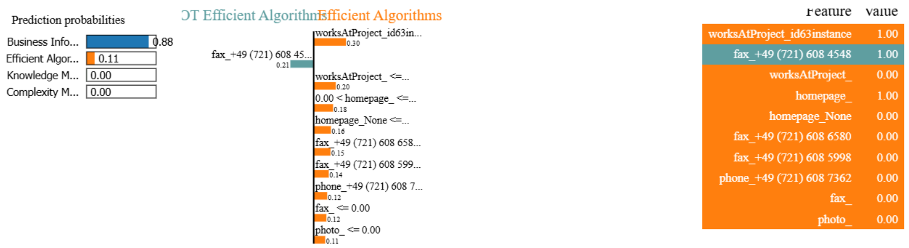
For the first test sample $x = x^{(0)}$, the XGBoost model predicted class `Business Information and Communicat` with a probability

$$f_{\mathtt{id1}}(x) = 0.88.$$

Applying LIME produced the top 10 feature weights w_j shown in Table 4, along with their conditions. Negative weights decrease the local prediction when the feature matches the original instance; positive weights increase it.

Table 3. Detailed LIME Weights for Test Instance: Daniel Ried

Feature Condition	Weight
`worksAtProject_id63instance > 0.00`	+0.2952
`fax_+49 (721) 608 4548 > 0.00`	−0.2137
`worksAtProject_ <= 0.00`	+0.2021
`0.00 < homepage_ <= 1.00`	+0.1773
`homepage_None <= 0.00`	+0.1577
`fax_+49 (721) 608 6580 <= 0.00`	+0.1538
`fax_+49 (721) 608 5998 <= 0.00`	+0.1380
`phone_+49 (721) 608 7362 <= 0.00`	+0.1232
`fax_ <= 0.00`	+0.1161
`photo_ <= 0.00`	+0.1051



Feature	Value
worksAtProject_id63instance	1.00
fax_+49 (721) 608 4548	1.00
worksAtProject_	0.00
homepage_	1.00
homepage_None	0.00
fax_+49 (721) 608 6580	0.00
fax_+49 (721) 608 5998	0.00
phone_+49 (721) 608 7362	0.00
fax_	0.00
photo_	0.00

Fig. 2. LIME Explainer(initial model)

4.7 Interpretation and Implications

Each weight w_j represents the contribution of a specific feature toward increasing or decreasing the model's confidence in predicting `id3instance` for the given entity. For example, the most negative feature, `fax_+49 (721) 608 4548`, reduces the predicted probability by 0.2137, indicating that this attribute is more strongly associated with other research group affiliations. In contrast, features like `worksAtProject_id63instance` and non-missing `homepage` entries contribute positively, increasing the likelihood of classification into `id1instance`.

Applying LIME produced the top 10 feature weights w_j shown in Table 4, along with their conditions. Negative weights decrease the local prediction when the feature matches the original instance; positive weights increase it.

4.8 Case Study (improved model): Test Instance: Amir Safari

For the first test sample $x = x^{(0)}$, the XGBoost model predicted class `Business Information and Communicat` with a probability

$$f_{\texttt{id4}}(x) = 0.97.$$

Applying LIME produced the top 10 feature weights w_j shown in Table 4, along with their conditions. Negative weights decrease the local prediction when the feature matches the original instance; positive weights increase it.

Table 4. Detailed LIME Weights for Test Instance: Amir Safari

Feature Condition	Weight
`20.00 < fax <= 23.00`	$+0.0745$
`48.50 < phone <= 67.25`	-0.0656
`worksAtProject <= 17.75`	-0.0345
`publication > 96.00`	$+0.0322$
`homepage <= 12.00`	-0.0185
`3.00 < type <= 4.00`	-0.0047

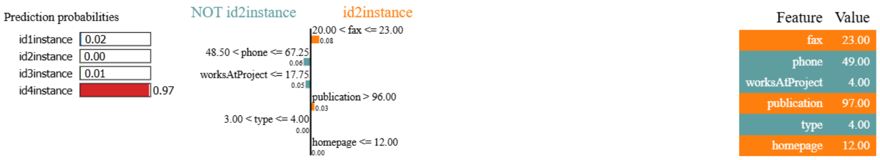


Fig. 3. LIME Explainer(improved model)

Each weight w_j represents the contribution of a specific feature toward increasing or decreasing the model's confidence in predicting `id4instance` for the given entity. For example, the most influential positive feature, `20.00 < fax <= 23.00`, increases the predicted probability by 0.0745, suggesting that a higher number of fax-related entries is associated with this affiliation. Conversely, features such as `48.50 < phone <= 67.25` and `worksAtProject <= 17.75` contribute negatively, reducing the model's confidence in assigning the label `id4instance`.

These weights indicate how specific patterns in structured data influence the classifier's decision for an individual prediction. By interpreting these contributions, LIME provides a clear explanation of the local behavior of the model, aligning with the goals of explainable AI to promote transparency and trust in predictive outcomes.

5 Conclusion

This mini-project demonstrates a tabular approach for classifying entities in RDF-based knowledge graphs, using the AIFB dataset as a case study. Initially, we trained an XGBoost model using features selected via variance thresholding. Based on the feature importance scores from this model, we refined the feature set and trained a second model, which achieved an improved test accuracy of 86%. The most informative features contributing to research group prediction included `phone`, `fax`, `worksAtProject`, `publication`, and `homepage`. These attributes consistently influenced the classification outcome. By integrating LIME for local interpretability, we gained valuable insights into the model's decision process, identifying key feature contributions at the instance level. Overall, this work shows that even without deep graph-based architectures, a structured tabular pipeline paired with explainability tools can effectively address classification tasks on RDF data while maintaining interpretability and performance.

Acknowledgements

This project was collaboratively completed by Ruthwik Krishna Bandreddy and Viraj Chandugade. Viraj Chandugade focused primarily on coding, data processing, model development, and ensuring alignment with the goals of explainable AI. Ruthwik Krishna Bandreddy was responsible for structuring the report, documenting findings, and contributing to result visualization. The authors also acknowledge the support of ChatGPT in enhancing the grammatical clarity of the report and refining the presentation of LIME-based explanations. All core analysis, reasoning, and model development were independently carried out by the authors.

6 Bibliography

@miscaifb$_{d}ataset, title = AIFBDataset, howpublished = https : //datahub.io/dataset/aifb, note = Accessed : 2025 - 07 - 02$

@miscdatacamp$_{x}gboost, author = DataCamp, title = XGBoostinPython, year = n.d., howpublished = https : //www.datacamp.com/tutorial/xgboost - in - python, note = Accessed : 2025 - 07 - 02$

@misclime$_{t}utorial, author = Ribeiro, MarcoTulio, title = LIME - BasicUsage, TwoClassCase, year = n.d., howpublished = https : //marcotcr.github.io/lime/tutorials/Limenote = Accessed : 2025 - 07 - 02$

@miscdhanya2021xgboost, author = Dhanya, N. M., title = Feature Selection Using XGBoost, year = 2021, howpublished = https://medium.com/@dhanyahari07/feature-selection-using-xgboost-f0622fb70c4d, note = Accessed: 2025-07-02