Assignment 6: Explainability for Machine Learning Models

Krishna Kumar Bais (241110038)
Rohan (241110057)

1 Introduction

This project extends a machine learning system for predicting employee absenteeism hours by adding comprehensive explainability features. The system uses a Linear Regression model trained on workplace data.

2 Explainability Methods Implemented

2.1 SHAP (Shapley Additive exPlanations)

2.1.1 Description

SHAP is a game-theoretic approach that assigns each feature an importance value (Shapley value) for a particular prediction. It provides mathematically rigorous feature attribution based on cooperative game theory.

2.1.2 Why SHAP?

We chose SHAP for several reasons:

- **Consistency**: Provides consistent explanations—if a model changes to rely more on a feature, SHAP values reflect this
- **Local Accuracy**: The sum of SHAP values equals the prediction minus the expected value
- **Global Insights**: Can be aggregated across samples for global feature importance

2.2 LIME (Local Interpretable Model-agnostic Explanations)

2.2.1 Description

LIME explains individual predictions by fitting a simple, interpretable model (e.g., linear regression) locally around the prediction of interest. It perturbs the input and observes how predictions change.

2.2.2 Why LIME?

LIME complements SHAP by providing:

- **Model Agnosticism**: Works with any black-box model
- **Local Fidelity**: Focuses on approximating the model locally rather than globally
- **Simplicity**: Produces sparse, human-interpretable explanations
- **Intuitive Approach**: Easier to explain to non-technical stakeholders ("what if" scenarios)

2.3 Counterfactual Explanations

2.3.1 Description

Counterfactual explanations answer: "What would need to change for the prediction to be different?" They provide actionable insights by identifying minimal changes to input features that would achieve a desired outcome.

2.3.2 Why Counterfactuals?

Counterfactuals are unique in providing:

- **Actionability**: Direct suggestions for reducing absenteeism
- **Human-centric**: Aligns with how humans naturally explain ("if only...")
- **Causality**: Hints at causal relationships (though not definitive)
- **Practical Value**: Helps managers and HR make intervention decisions

3 Global and Local Explanations

3.1 Global Explanations

Global explanations describe overall model behavior across all predictions.

3.1.1 Insights Provided

- Which features are most influential overall
- Relative importance of different feature categories
- Model's general decision-making patterns

3.2 Local Explanations

Local explanations describe why a specific prediction was made for a particular instance.

3.2.1 Insights Provided

- Why this specific employee has high/low predicted absenteeism
- Which of their attributes contribute most to the prediction
- What they could change to reduce absenteeism

4 Addressing Common User Questions

Explainability should answer specific questions users have about model predictions. We map our implementations to common questions.

4.1 "Why did the model predict this?"

4.1.1 Answer Provided By

- **Primary**: SHAP local explanations
- **Secondary**: LIME local explanations

4.1.2 Implementation

- Show top 5-10 features ranked by absolute contribution
- Display direction of influence (increases/decreases prediction)
- Provide magnitude of impact in hours
- Generate natural language summary: "Service time increases prediction by 2.3 hours"

4.1.3 Example User Interaction

> *User*: "Why did John have 12 hours of predicted absenteeism?"
> *System*: "The prediction is 12.0 hours. Top factors: Service time (15 years) increases prediction by 3.2 hours, Age (55) increases by 2.1 hours, Social drinker status increases by 1.8 hours."

4.2 "What features does the model consider most important?"

4.2.1 Answer Provided By

- **Primary**: Global SHAP feature importance
- **Secondary**: Model coefficients (for linear models)

4.2.2 Implementation

- Rank features by mean absolute SHAP value
- Display as sortable table or bar chart in UI
- Updated weekly via caching mechanism

4.3 "How can I change the outcome?"

4.3.1 Answer Provided By

- **Primary**: Counterfactual explanations
- **Secondary**: SHAP contributions (show what to change)

4.3.2 Implementation

- Generate 5 actionable suggestions
- Focus on modifiable features (exclude immutable ones like age)
- Show expected impact: "Reducing workload by 50 units would decrease prediction to 9.2 hours"
- Rank by reduction potential and feasibility (distance)

4.4 "Is the model fair across different groups?"

4.4.1 Answer Provided By

- **Primary**: Fairness gap metrics
- **Secondary**: Group-wise SHAP value distributions

4.4.2 Implementation

- Compute MAE gaps across sensitive attributes:
  - Age groups: 0.00 hours (perfectly fair)
  - Education levels: 0.00 hours (perfectly fair)
- Display in model information endpoint
- Track over time for monitoring

4.5 "When is the model uncertain?"

4.5.1 Answer Provided By

- **Primary**: Confidence scores (if using ensemble/probabilistic models)
- **Secondary**: SHAP value variance, LIME explanation score

4.5.2 Implementation

Linear Regression provides point estimates without uncertainty. For future work:

- Could implement Bayesian Linear Regression for credible intervals
- Use LIME explanation score as proxy for local model complexity
- Bootstrap predictions for empirical confidence intervals

4.6 "What if the input was slightly different?"

4.6.1 Answer Provided By

- **Primary**: Counterfactual explanations
- **Secondary**: SHAP sensitivity analysis

4.6.2 Implementation

Counterfactuals directly answer this by showing alternative scenarios. SHAP provides gradient-like information about how predictions change with inputs.

4.7 "Can I trust this prediction?"

4.7.1 Answer Provided By

- **Primary**: Combination of all methods
- **Secondary**: Model performance metrics

4.7.2 Implementation

Trust is built through:

- **Transparency**: Show which features influenced the prediction
- **Consistency**: SHAP stability = 1.0 (perfect consistency)
- **Reasonableness**: Explanations align with domain knowledge
- **Performance**: Display model metrics (RMSE, MAE, R^2)
- **Fairness**: Show zero bias across demographic groups

5 Key Design Decisions

5.1 Caching Strategy

Global explanations are expensive to compute, so we:

- Cache results in `explain_global_cache.json`
- Set TTL to 7 days
- Invalidate on model updates

5.2 Fallback Mechanisms

To ensure robustness, we implement fallbacks:

- If SHAP fails, use model coefficients as importance
- If LIME fails, return coefficient-based weights
- If counterfactuals fail, return empty candidates with message
- Never return HTTP errors for explanation endpoints

5.3 Performance Optimization

- Use `LinearExplainer` for exact, fast SHAP computation
- Limit background samples to 100 for reasonable computation time
- Limit counterfactual search space to actionable features only
- Disable heavy explainers on resource-constrained deployments (Render free tier)

6 Deployment and Accessibility

6.1 Live Application

The application is deployed and accessible at:

https://absenteeism-app.onrender.com/

6.2 Source Code Repository

Full source code is available at:

https://github.com/krishna-kumar-bais/Assignment__6_

7 Conclusion

This project successfully implements comprehensive explainability for an absenteeism prediction system using three complementary methods: SHAP, LIME, and Counterfactual Explanations.

The implementation demonstrates that explainability is not a single technique but a suite of complementary methods. SHAP provides rigorous mathematical foundations, LIME offers intuitive local approximations, and Counterfactuals deliver actionable insights. Together, they create a transparent, trustworthy, and useful ML system for real-world deployment.