Theoretical Framework for the Diagnosis and Prognosis of a disease through Explainable Artificial Intelligence

The role of explainable AI in the diagnosis and prognosis of a disease is

1. **Interpretable Insights**: Explainable AI provides clear and understandable insights into how a diagnosis or prognosis is reached. It helps medical professionals and patients understand the factors considered by the AI model in making its recommendations.

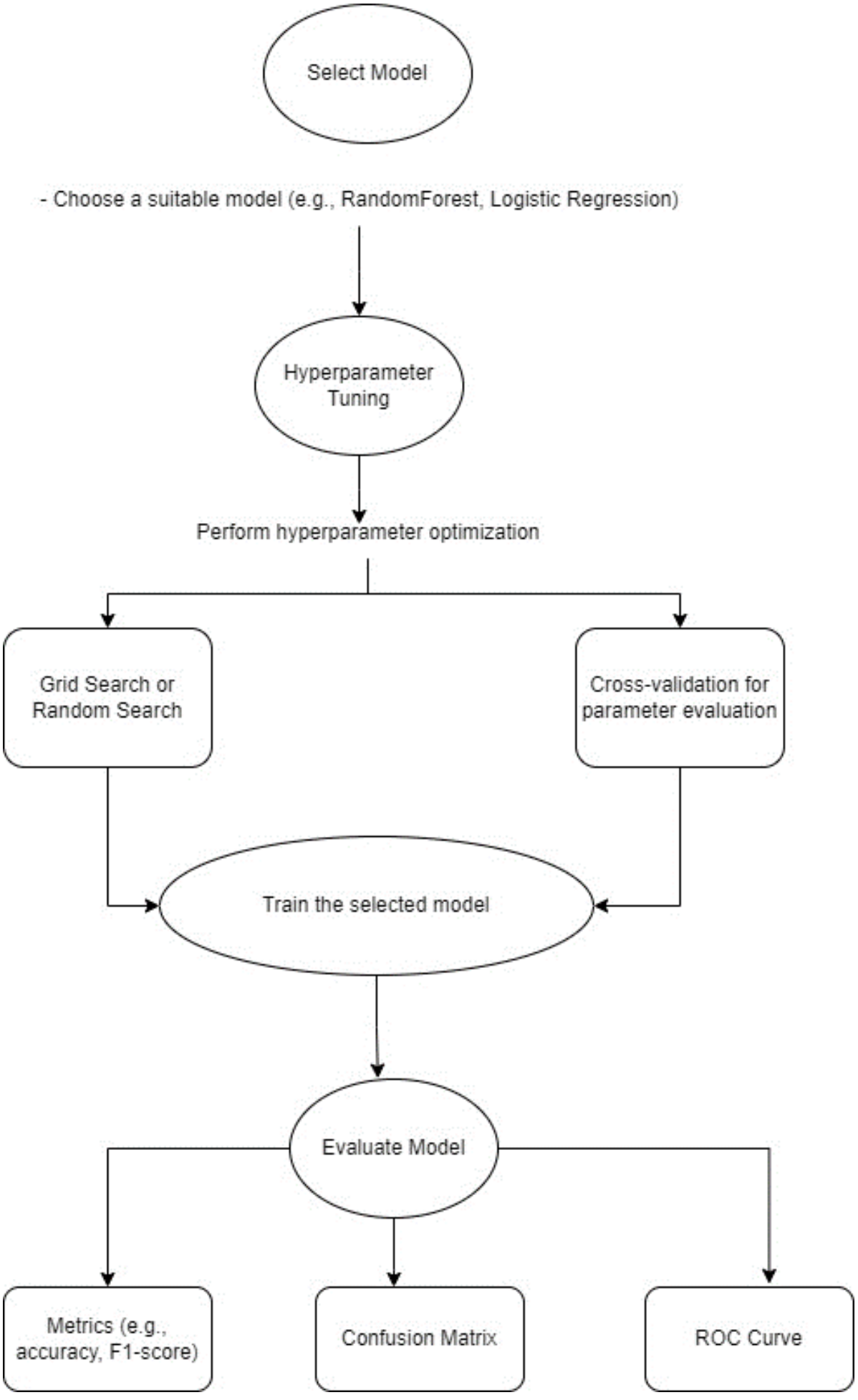
2. **Clinical Decision Support**: Explainable AI serves as a valuable tool for clinicians by offering additional information and context when making critical medical decisions. It assists doctors in confirming or refining their diagnoses and treatment plans.

3. **Enhancing Trust**: The transparency of explainable AI builds trust among healthcare professionals. When the AI system can explain its reasoning, it fosters confidence in the technology's capabilities, leading to increased acceptance in medical practice.

4. **Error Detection and Correction**: Explainable AI can assist in identifying errors or biases in the data or model, enabling corrective actions. This is essential for ensuring the accuracy and reliability of disease predictions.

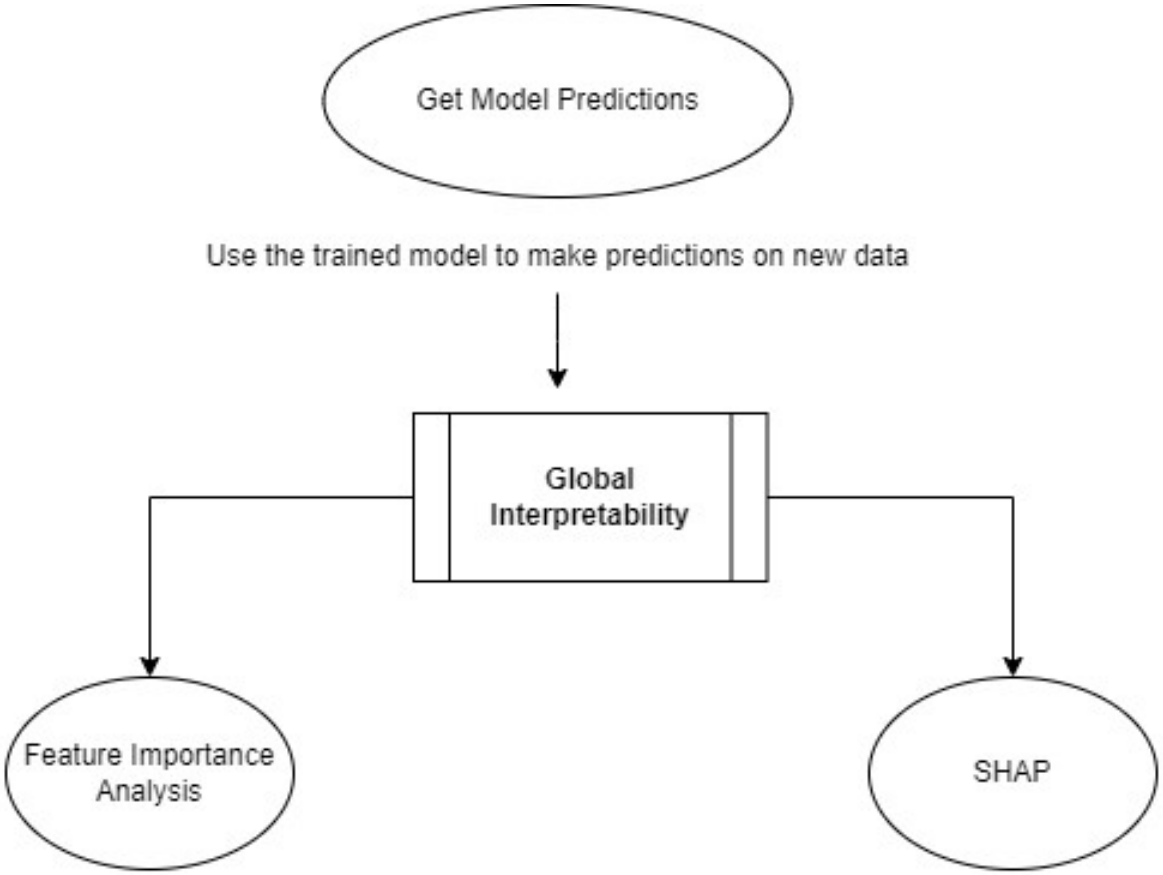
5**. Enhanced Medical Education**: Explainable AI can be used as an educational tool for medical students and healthcare practitioners. It aids in understanding complex medical cases, the rationale behind diagnoses, and treatment decisions.

The tools and the workflow diagram for this explainable AI are provided in the following pages.


Select Model
- Choose a suitable model (e.g., RandomForest, Logistic Regression)
Hyperparameter Tuning
Perform hyperparameter optimization
Grid Search or Random Search
Cross-validation for parameter evaluation
Train the selected model
Evaluate Model
Metrics (e.g., accuracy, F1-score)
Confusion Matrix
ROC Curve

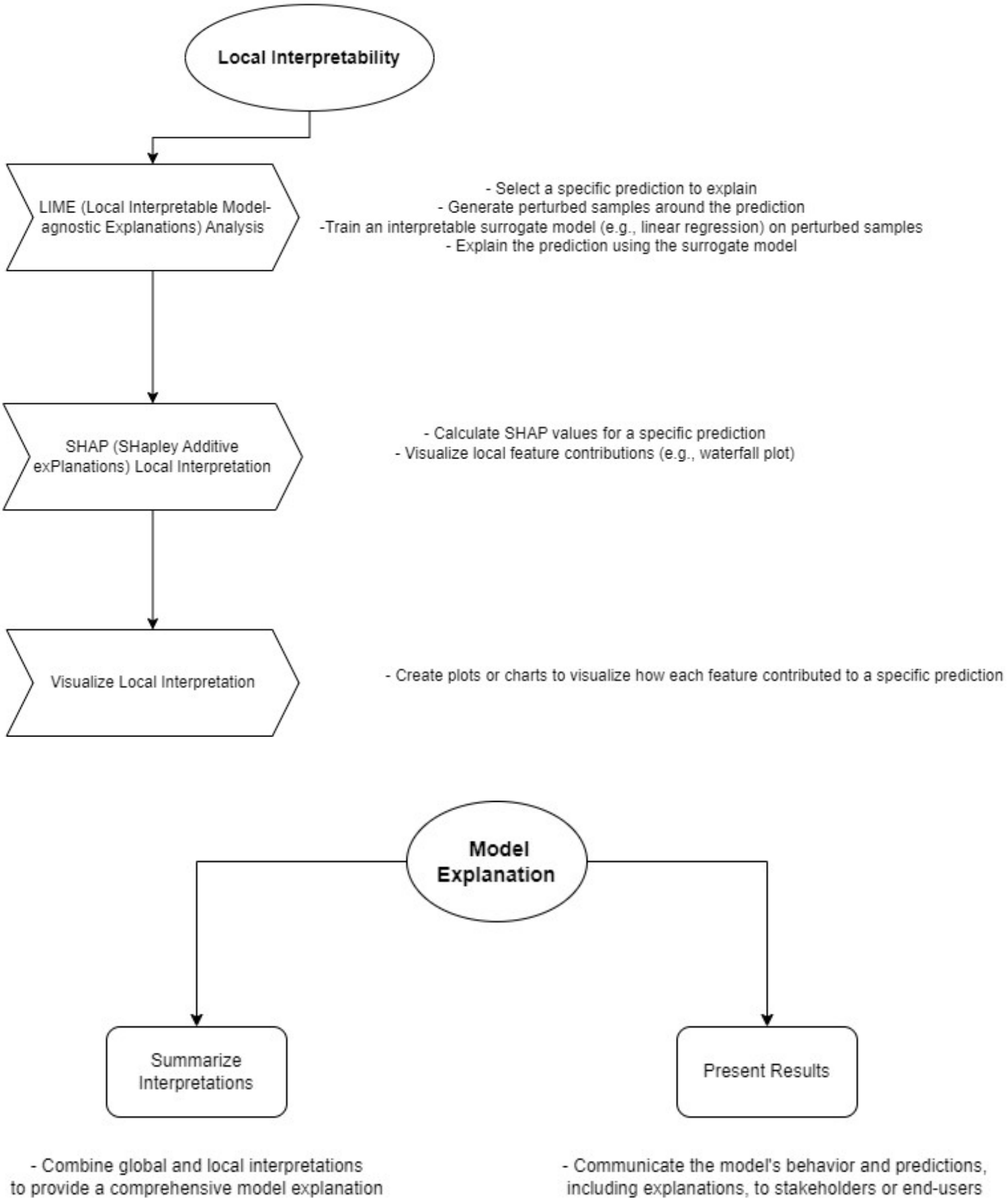
MODEL EXPLAINABILITY

Flow Chart



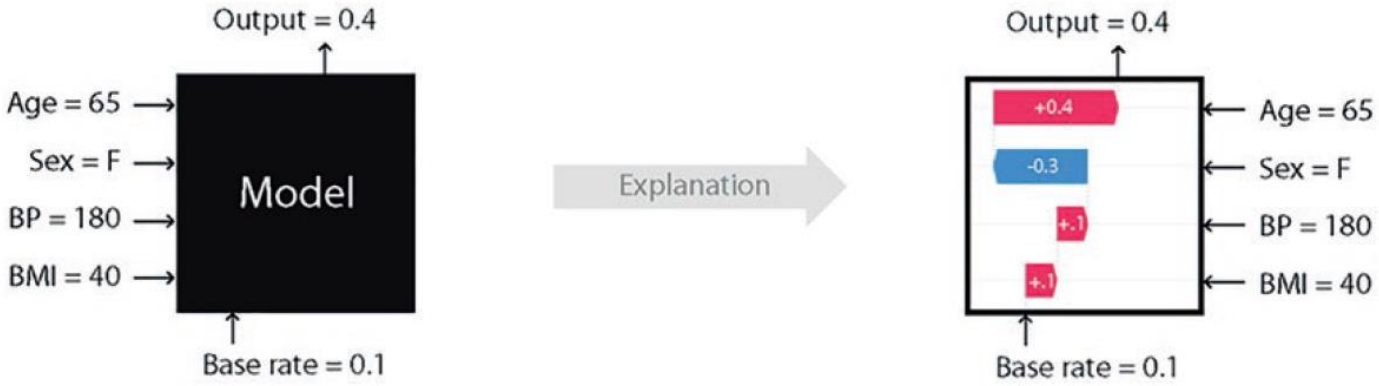
- Compute feature importance scores
- Visualize feature importance (e.g., bar chart)

- Calculate SHAP values for each feature
- Visualize global feature importance using summary plots


Local Interpretability
LIME (Local Interpretable Model-agnostic Explanations) Analysis
- Select a specific prediction to explain
- Generate perturbed samples around the prediction
-Train an interpretable surrogate model (e.g., linear regression) on perturbed samples
- Explain the prediction using the surrogate model
SHAP (SHapley Additive exPlanations) Local Interpretation
- Calculate SHAP values for a specific prediction
- Visualize local feature contributions (e.g., waterfall plot)
Visualize Local Interpretation
- Create plots or charts to visualize how each feature contributed to a specific prediction
Model Explanation
Summarize Interpretations
- Combine global and local interpretations to provide a comprehensive model explanation
Present Results
- Communicate the model's behavior and predictions, including explanations, to stakeholders or end-users

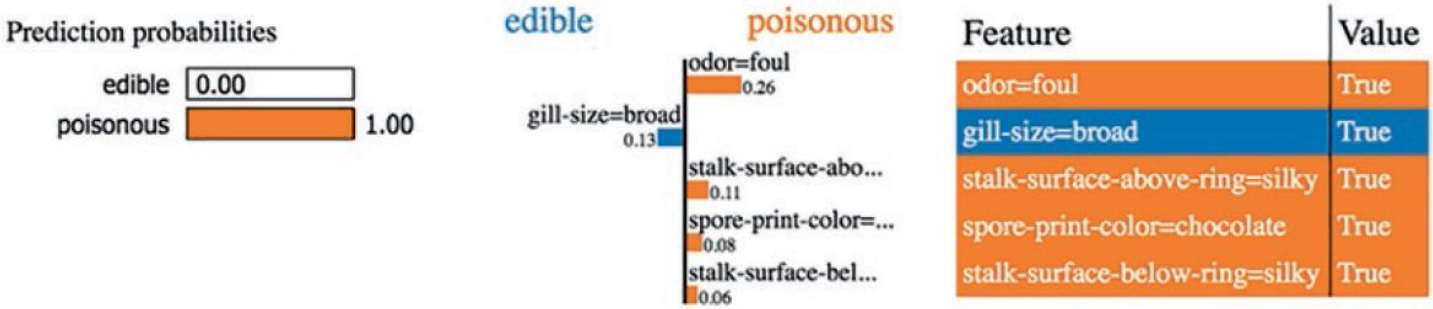
Tools for Model Explainability

The **SHAP (SHapley Additive explanations)** library is a Python-based unified approach to explain the output of any machine learning model. The SHAP Python library is based on game theory with local explanations. The game theory approach is a way to get predictions if one factor is present vs. when it is absent. If there is a significant change in the expected outcome, then the factor is very important to the target variable. This method unites several previous methods to explain the output generated by the machine learning models.



LIME

LIME is stands for Local Interpretable Model-Agnostic Explanations. Local refers to the explanation around the locality of the class that was predicted by the model. The behavior of the classifier around the locality gives a good understanding about the predictions. Interpretable means if the prediction cannot be interpreted by a human being, there is no point. Hence the class predictions need to be interpretable. Model agnostic implies instead of understanding a particular model type, the system and method should be able to generate the interpretations.



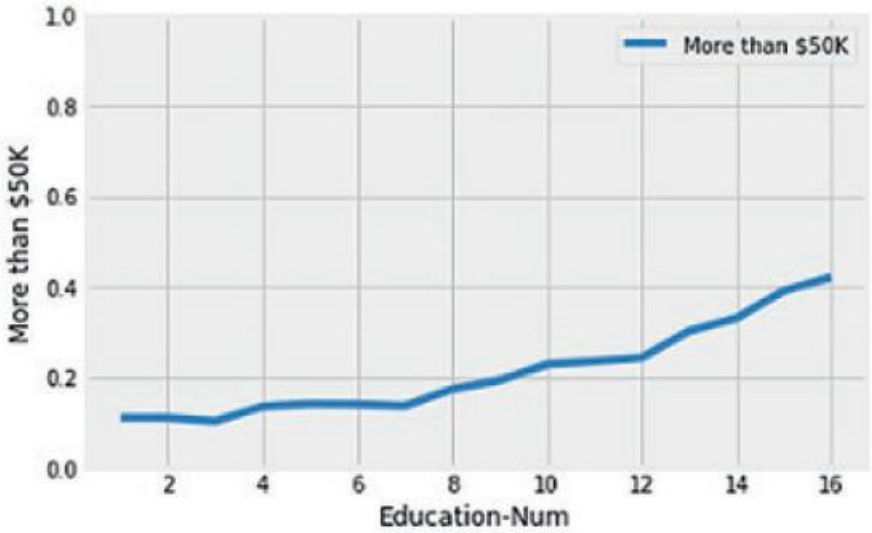
ELI5

ELI5 is a Python-based library intended to be used for an explainable AI pipeline, which allows us to visualize and debug various machine learning models using a unified API. It has built-in support for several ML frameworks and provides a way to explain black box models.

Weight	Feature
0.3717	relationship
0.1298	marital_status
0.1247	education_num
0.1108	capital_gain
0.0611	capital_loss
0.0362	age
0.0307	occupation
0.0298	sex
0.0289	hours_per_week
0.0188	workclass
0.0161	native_country
0.0160	race
0.0132	fnlwgt
0.0123	education

Skater

Skater is an open source unified framework to enable model interpretation for all forms of models to help us build an interpretable machine learning system, which is often needed for real world use-cases. Skater supports algorithms to demystify the learned structures of a black box model both globally (inference on the basis of a complete data set) and locally (inference about an individual prediction).



PDP with Skater