QUESTION BANK

EXPLAINABLE AI

By Prof- Yashwant Sudhakar Ingle.

16th March, 2025

1. Why is Explainable AI (XAI) important in modern Artificial Intelligence applications? Discuss

with examples.

1. **PDP (Partial Dependency Plot)** shows how one feature affects the prediction while keeping others fixed.
   ➜ Like checking how "age" changes results if everything else stays the same.

2. **SHAP for Decision Trees** gives each feature a score based on how much it helped in making the prediction.
   ➜ Like saying "credit score helped 30%, income helped 20%" in a loan prediction.

3. **LIME explains tricky models** by creating a simple version near the prediction to show which features mattered.
   ➜ Example: In fraud detection, LIME might show "big amount" and "foreign location" caused the fraud prediction.

4. **Ensemble models** combine many models to make better predictions.
   ➜ Like asking 5 friends instead of one – you get a smarter answer.

5. **Counterfactual Explanations (CFE)** tell you what small change could flip the model's decision.
   ➜ Example: "If your salary was ₹10,000 more, your loan would be approved."

6. **CEM (Contrastive Explanation Method)** shows what features made the model say "yes" and what missing things could have made it say "no."
   ➜ Like saying: "You got the job because of your degree, and you didn't lose it because you weren't late."

2. Differentiate between Explainability and Interpretability. Explain how they affect AI model understanding.

Aspect	Explainability	Interpretability
Meaning	How well the output can be explained in human terms	How well we understand what's going on inside the model
Focus	Focuses on the *results*	Focuses on the *inner working* of the model
Example	"Why did the model say loan = rejected?"	"How does the model process inputs to make decisions?"
Used For	Explaining predictions to users or stakeholders	Understanding the logic, math, and structure of the model
Techniques Needed	Often needs extra tools (like SHAP, LIME)	Simple models (like Decision Trees) are directly interpretable
Model Types	Even black-box models can be explainable with tools	Only simple models are naturally interpretable

Explainability is the degree to which a model's output can be explained to a human in understandable terms, while interpretability refers to how well a human can grasp the internal mechanics of the model itself. A decision tree is both explainable and interpretable, but deep neural networks are often only explainable through post-hoc techniques. These concepts affect transparency, trust, and the ability to debug or improve models effectively.

---

3. Explain the various types of XAI techniques with examples: Intrinsic vs Post-hoc, Local vs Global, and Model-specific vs Model-agnostic.

- **Intrinsic vs Post-hoc**: Intrinsic methods are interpretable by design, such as linear regression or decision trees, whereas post-hoc methods like SHAP or LIME explain decisions made by complex black-box models after training.
- **Local vs Global**: Local methods, such as LIME, provide explanations for individual predictions, while global methods, such as feature importance plots, provide insights into the overall behavior of the model.
- **Model-specific vs Model-agnostic**: Model-specific techniques work only with certain model types (e.g., SHAP with tree models), while model-agnostic techniques like LIME can be applied to any predictive model.

---

4. What is the role of VIF (Variance Inflation Factor) in Linear Regression? How can a high VIF value affect the final model?

✅ Role of VIF (Variance Inflation Factor):

- Identifies multicollinearity in linear regression
- Measures how much a variable is correlated with other variables
- Helps in spotting redundant or overlapping features
- Guides which features to remove or combine
- Ensures model stability and better interpretation

⚠️ Effects of High VIF (>10):

- Strong multicollinearity present
- Coefficients become unreliable or unstable

- Increases standard errors of estimates
- Reduces model accuracy and interpretability
- Can mislead decision-making based on model output

---

5. Explain how SHAP provides both local and global explanations in a machine learning model.

📌 SHAP: Local & Global Explanations

- Uses **Shapley values** to give fair feature importance
- Based on **game theory**

🔹 Local Explanation (Single Prediction):

- Explains why the model made a specific prediction
- Shows each feature's impact on that one output
- Helps in **case-wise debugging** and **trust building**

🔹 Global Explanation (Whole Model):

- Averages feature impacts across all data points
- Shows which features are most important overall
- Helps in **model understanding** and **feature selection**

✅ Benefits:

- Gives both **individual** and **overall** insights
- Useful for **transparency**, **trust**, and **debugging**

---

6. How does LIME work to explain individual predictions in a black-box model? Explain with an example.

📌 How LIME Works (Local Interpretable Model-agnostic Explanations):

- Focuses on explaining **individual predictions**
- Works with any **black-box model** (model-agnostic)
- **Perturbs** the input data (makes small changes)
- Observes how the **model's output changes**
- Trains a simple, interpretable model (like linear regression) on these changes
- The simple model shows which features mattered most for that specific prediction

🔍 Example: Sentiment Analysis

- Input: A movie review – “The acting was excellent, but the story was terrible.”
- LIME highlights:
    - **"Excellent"** pushed the prediction toward **positive**
    - **"Terrible"** pushed the prediction toward **negative**
- Shows which words influenced the model's decision the most

---

7. What is a Partial Dependency Plot (PDP), and what does it help visualize?

📘 Definition:

A **Partial Dependency Plot (PDP)** is a graphical tool used to show the **average effect of a feature** on a model's prediction while keeping all other features constant. It helps interpret how changes in one input feature influence the predicted outcome.

⚙️ How It Works:

- Select a feature (e.g., age)
- Vary its values across a range (e.g., 20 to 70)
- For each value, the model predicts outcomes **while keeping other features fixed**
- Plot the average predictions for each value
- The result shows how the feature **impacts** the model's output

🌟 Example:

In a **house price prediction** model:

- You want to see how **"number of rooms"** affects price
- PDP might show that as rooms increase from 2 to 5, predicted price increases steadily
- Beyond 5 rooms, the price might flatten — showing **diminishing return**

8. How does SHAP explain a Decision Tree model prediction?

📘 How SHAP Explains a Decision Tree Prediction:

- **SHAP** (Shapley Additive Explanations) assigns **fair contribution scores** to each feature.
- It traces the **role of each feature** by following the **decision paths** in the tree.
- For each prediction, SHAP calculates how much each feature **influenced** the final outcome.

⚙️ How It Works:

- **Traces the tree paths**: SHAP checks the decision rules followed at each split.
- **Calculates the impact**: It computes the **contribution** of each feature to the prediction by evaluating how the feature helps lead to a certain branch or outcome.
- **Adds contributions**: SHAP adds all the contributions of features to form the total prediction.

🌟 Example:

In a **loan approval** model:

- The model uses features like **income**, **credit score**, and **loan amount**.
- SHAP assigns scores based on how each feature influenced the prediction:
    - **Credit score** might contribute +20% to the "approved" decision.
    - **Income** could add +10%, while **loan amount** might have a minor effect of +5%.

---

9. Explain how LIME helps in understanding the predictions of non-linear models. Give one example.

📘 How LIME Helps in Understanding Non-Linear Models:

- **LIME** (Local Interpretable Model-agnostic Explanations) helps explain predictions made by **complex, non-linear models**.
- It works by:
    1. **Perturbing** (slightly changing) the input data around a specific prediction.
    2. Observing how the **model's output changes** based on these small changes.
    3. Fitting a **simple, interpretable model** (e.g., linear regression) on the perturbed data.
    4. **Identifying which features** influenced the model's decision for that particular instance.

⚙️ How It Works:

- LIME generates **local approximations** to the model's decision boundary.
- This helps break down complex predictions into **simple, understandable explanations**.

🌟 Example: Fraud Detection

- In a **fraud detection** model:
    - **Transaction amount** and **foreign IP address** may be key features.
    - LIME helps show that these features **heavily influenced** the model's prediction of fraud.
- This transparency increases **trust** in the model and helps review **decision boundaries**.

---

10. What are ensemble models and why are they used in machine learning?

📘 What are Ensemble Models?

- **Ensemble models** combine multiple base models (e.g., decision trees, neural networks) to create a stronger, more accurate model.
- The combined predictions from multiple models are used to make the final decision, improving overall performance.

✅ Why Are Ensemble Models Used in Machine Learning?

- **Improve accuracy** by combining strengths of different models.
- **Increase robustness**, making the model less sensitive to errors or noise in data.
- **Reduce overfitting** by averaging out predictions and focusing on the general trend rather than noise.
- **Increase predictive performance** by leveraging diversity in base models.

🌟 Example:

- **Random Forest** is an ensemble model that combines multiple decision trees to make predictions. Each tree contributes to the final decision, reducing the risk of overfitting and improving accuracy.

---

11. Name any two types of ensemble learning techniques.

📘 Bagging (Bootstrap Aggregating):

- Combines multiple **independent models** (e.g., decision trees).
- Each model is trained on a **random subset** of the data (with replacement).
- Final prediction is made by **averaging** (regression) or **voting** (classification).

- **Example:** Random Forest.

📘 Boosting:

- Focuses on improving the **weak models** by sequentially training them.
- Each new model **corrects the errors** made by the previous one.
- The final prediction is made by **combining** the weighted predictions of all models.
- **Example:** Gradient Boosting Machines (GBM), XGBoost.

---

12. Explain how SHAP can be used to interpret the predictions of an ensemble classification model.

📘 SHAP for Ensemble Models:

- **SHAP** (Shapley Additive Explanations) helps interpret **ensemble classification models** like **random forests** and **gradient boosting machines**.
- It calculates **Shapley values**, which show each feature's **fair contribution** to the final prediction.
- These values provide **consistent and fair explanations** across all models in the ensemble.

⚙️ How It Works:

- For each prediction, SHAP **traces how each feature** affects the model's output.
- **Additive nature**: SHAP ensures that the contributions from all features add up to the final prediction, making it easy to understand how each feature influences the outcome.

🌟 Example: Healthcare Model for Heart Disease Risk

- In a healthcare model (e.g., predicting heart disease risk using random forests):
    - SHAP shows how features like **age**, **cholesterol level**, and **blood pressure** contribute to the prediction.

- For instance:
    - **Age**: +10% risk
    - **Cholesterol level**: +15% risk
    - **Blood pressure**: +5% risk
- This **local interpretation** helps explain why a specific individual was predicted to have a high risk.

---

13. What is the What-If Tool (WIT) by Google?

📘 What is the What-If Tool (WIT)?

- **WIT** is a **visual tool** designed to help users **inspect** and **experiment** with machine learning models without writing any code.
- It provides an **interactive interface** for analyzing model predictions and testing different scenarios.

⚙️ Key Features of WIT:

- **Model Inspection**: Allows you to **view** and **analyze** how your model makes predictions.
- **What-If Scenarios**: Lets you test **"what-if"** scenarios by changing input features to see how the predictions change.
- **Exploration of Fairness**: Can help assess **bias** or fairness in predictions.
- **Works without code**: Users can experiment directly through a graphical interface.

🌟 Example:

- In a **loan approval** model:

- You can use WIT to test how **changing income** or **loan amount** impacts the approval prediction without needing to modify the code.

---

14. What are Counterfactual Explanations (CFEs) in XAI?

📘 What are Counterfactual Explanations (CFEs)?

- **CFEs** are explanations that suggest **minimal changes** to an input that would change the model's prediction.
- They show what needs to be **different** in the input data to **flip** the prediction to a desired class or outcome.

⚙️ How CFEs Work:

- The goal is to find the **smallest change** in features that would result in a different prediction.
- These changes help the user understand **decision boundaries** and provide actionable insights.

🌟 Example:

- In a **loan approval** model:
  - If a loan is rejected, a **CFE** might suggest:
    - Increase income by **₹10,000** and reduce existing debt by **₹5,000** to get approval.
- This helps the user understand how to **alter inputs** to change the prediction.

---

15. What are Counterfactual Explanations (CFEs), and how do they improve explainability?

📘 What are Counterfactual Explanations (CFEs)?

- **CFEs** show **hypothetical "what-if" scenarios**, asking questions like: "What should be different in this input to change the output?"
- They identify the **minimal changes** in input features that would flip a model's prediction to a desired class or outcome.

⚙️ How CFEs Improve Explainability:

- **Clarifies decision boundaries**: By showing how small changes affect predictions, CFEs help users understand what drives a model's decision.
- **Actionable insights**: They provide users with specific, understandable steps to modify inputs and potentially change a prediction (e.g., for loan approvals or healthcare decisions).
- **Empowers users**: CFEs give users the ability to see how they can **influence** decisions and improve outcomes.
- **Enhances fairness and transparency**: By highlighting minimal required changes, CFEs can reveal any biases in the model and offer fairer decision-making insights.

🌟 Example: Loan Approval

- If a **loan application is rejected**, a **CFE** might suggest:
    - Increase **income by ₹10,000** and reduce **debt by ₹5,000** to get approval.
- This provides **clear guidance** on what actions the applicant can take to change the result.

16. What is the Contrastive Explanation Method (CEM) in Machine Learning?

📘 What is the Contrastive Explanation Method (CEM)?

- **CEM** is a method used in **explainable AI (XAI)** that focuses on providing both **positive** and **negative** explanations for a model's prediction.
- It highlights:
    - **Positive features**: Features that **justify** the model's decision (i.e., they are important for the prediction).
    - **Negative features**: Features whose **absence** would **change** the model's prediction.

⚙️ How CEM Works:

- **Pertinent positives**: Features that **must be present** for the prediction to happen.
- **Pertinent negatives**: Features that, if present, would have caused a **different prediction**.
- CEM uses a **loss function** to optimize and find the minimal changes in input features that influence the model's outcome.

🌟 Example: Disease Diagnosis

- For a **disease prediction** model:
    - **Pertinent positives**: Features like **smoking history** and **high blood pressure** may increase the risk of disease.
    - **Pertinent negatives**: **Lack of exercise** or **absence of a family history of disease** might be important for predicting a healthy outcome.
- This contrastive explanation provides both **justification** and the **absence of features** that affect the decision.

---

17. How does CEM differ from other explanation methods like SHAP or LIME?

📘 Difference Between CEM, SHAP, and LIME:

- **CEM (Contrastive Explanation Method)**:
    - **Contrastive**: Focuses on the **presence and absence** of features.
    - **Key Focus**: Identifies **pertinent positives** (features that justify the prediction) and **pertinent negatives** (features whose absence would change the prediction).
    - Provides **contrastive insights**, helping users understand **what features need to be present** or **absent** for a model's decision.
- **SHAP (Shapley Additive Explanations)**:
    - Focuses on **additive importance scores** for each feature.
    - Calculates the **Shapley value** for each feature, showing how much each feature **contributes** to the final prediction.
    - Provides **global and local** explanations, offering a **fair contribution score** for each feature.
- **LIME (Local Interpretable Model-agnostic Explanations)**:
    - **Local model approximation**: Generates simple, interpretable models (e.g., linear regression) to approximate the black-box model's behavior for a specific instance.
    - Focuses on the **local behavior** of the model and explains **how input features affect individual predictions**.

⚙️ Key Difference:

- **CEM**: Emphasizes **contrastive** reasoning by identifying both **presence** and **absence** of features.
- **SHAP & LIME**: Focus on **additive** importance scores, showing how each feature contributes or influences the model’s prediction.

---

18. Explain how CEM works using the Alibi library for tabular data.

📘 How CEM Works with Alibi for Tabular Data:

1. **Key Concepts**:
    - **Pertinent Positives**: Features that are **essential** for the model to make a particular prediction (e.g., presence of exercise for predicting a healthy outcome).
    - **Pertinent Negatives**: Features whose **absence** would **change** the model's prediction (e.g., absence of smoking history for predicting a healthy outcome).
2. **Optimization Process**:
    - CEM uses a **loss function** to **optimize the input data**.
    - It aims to find the **minimal changes** in input features that will **flip** or influence the prediction.
    - The **loss function** balances:
        - **Prediction accuracy**: Ensures the prediction is still valid after changes.
        - **Sparsity**: Keeps the changes to a minimum.
        - **Contrast**: Focuses on what makes the prediction different when features are **present** or **absent**.

🌟 Example: Disease Diagnosis:

- In a **disease diagnosis model**, the features might include:
    - **Age**
    - **Smoking history**
    - **Exercise frequency**
    - **Cholesterol levels**

- **CEM in Action**:
    - **Pertinent positives**: Features like **exercise frequency** and **low cholesterol levels** may contribute to a **healthy** prediction.
    - **Pertinent negatives**: The **absence of smoking** and **absence of family history** could also influence the **healthy** outcome.
    - CEM identifies the **minimal input changes** that would flip the model's prediction from **unhealthy** to **healthy**.
