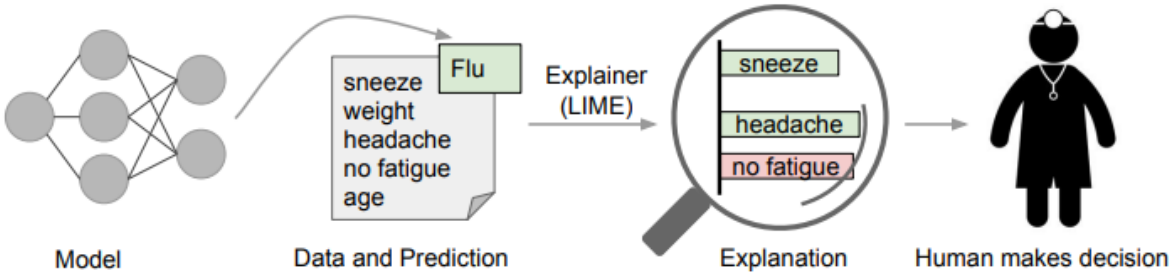
LIME Explainability: Shedding Light on Machine Learning Models

In the ever-evolving world of artificial intelligence and machine learning, the black box nature of complex models has long been a challenge. While deep learning algorithms have achieved remarkable success in various domains, understanding how and why these models make certain predictions remains elusive. Enter Lime, an interpretability technique that seeks to unravel the inner workings of machine learning models. In this blog post, we will explore Lime explainability, its significance, and how it brings transparency to the decision-making process of AI systems.

LIME: Lime, short for "Local Interpretable Model-Agnostic Explanations," is a method developed by Marco Ribeiro, Sameer Singh, and Carlos Guestrin in 2016 and it was proposed as part of paper "Why Should I Trust You?" to address the explainability problem. Lime is a model-agnostic approach, meaning it can be applied to any machine learning model, regardless of its complexity or underlying architecture.

The primary goal of Lime is to provide explanations for individual predictions by approximating the model's behavior in the vicinity of a specific data point. It works by perturbing the features of the input data and observing how the model's predictions change. By analyzing these changes, Lime creates an interpretable model, such as a linear regression, that locally approximates the behavior of the complex model.



Picture: Why should I trust you?

How Lime Works?

Here's a step-by-step breakdown of how Lime works.

- **Selection of Instances:** Lime begins by selecting instances from the dataset that are similar to the instance for which an explanation is desired. These instances are chosen to represent the local neighborhood of the specific data point.
- **Feature Perturbation:** Lime perturbs the features of the selected instances while keeping the label constant. It generates a new dataset by sampling perturbed instances. The perturbation process involves modifying the feature values within a defined range, such as adding or subtracting small amounts from the original values.

- **Prediction Analysis:** The perturbed dataset is passed through the black box model, for which the explanations are sought, and the resulting predictions are recorded. Lime collects the predictions made by the model on the perturbed instances, capturing the model's behavior in the local neighborhood**.**
- **Local Model Fitting:** Lime fits a local interpretable model, such as linear regression or decision tree, to explain the predictions based on the perturbed data. The interpretable model is trained using the perturbed instances and their corresponding predictions obtained from the black box model. The interpretable model approximates the behavior of the black box model within the local neighborhood.
- **Explanation and Generation:** Once the local interpretable model is fitted, Lime generates explanations based on its analysis. It highlights the features that contribute most to the prediction made by the complex model, providing insights into the model's reasoning.
  Lime assigns importance weights to each feature, indicating their influence on the prediction. Higher weights signify greater importance, while lower weights suggest less significance.

Why should we use Lime?

Lime offers several advantages over other explainability techniques.

1. Model Agnosticism: Lime can be applied to any machine learning model, regardless of its architecture or complexity.
2. Local Explanations: Lime provides insights into the decision-making process of the model for specific instances, allowing users to understand the reasoning behind individual predictions.
3. Flexible Feature Importance: Lime provides fine-grained feature importance weights, indicating the contribution of each feature to the predictions.
4. Intuitive and Interpretable: Lime aims to generate human-interpretable explanations using simple and familiar models, making them accessible to users without deep technical knowledge.
5. Bias Detection and Fairness: Lime can help identify biases or discriminatory behavior in models by analyzing the importance of different features.

6. Regulatory Compliance: Lime enables organizations to provide justifiable and interpretable AI systems, which is essential in regulated industries.

Let's take an example of how Lime explainability works.

Before we get started, we need to install lime.

```
!pip install lime
```

Import all required libraries.

```
# Importing the necessary libraries
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
```

Load dataset

```
# Loading the dataset using sklearn
from sklearn.datasets import fetch_california_housing
data = fetch_california_housing()
# Displaying relevant information about the data
print(data['DESCR'][200:1420])
```

```
attributes and the target

   :Attribute Information:
       - MedInc        median income in block group
       - HouseAge      median house age in block group
       - AveRooms      average number of rooms per household
       - AveBedrms     average number of bedrooms per household
       - Population    block group population
       - AveOccup      average number of household members
       - Latitude      block group latitude
       - Longitude     block group longitude

   :Missing Attribute Values: None
```

Separating data in feature variable and target variable

```
# Separating data into feature variable X and target variable y respectively
from sklearn.model_selection import train_test_split
X = data['data']
y = data['target']
# Extracting the names of the features from data
features = data['feature_names']
# Splitting X & y into training and testing set
X_train, X_test, y_train, y_test = train_test_split(
  X, y, train_size=0.90, random_state=50)
# Creating a dataframe of the data, for a visual check
df = pd.concat([pd.DataFrame(X), pd.DataFrame(y)], axis=1)
df.columns = np.concatenate((features, np.array(['label'])))
print("Shape of data =", df.shape)
# Printing the top 5 rows of the dataframe
df.head()
```

```
Shape of data = (20640, 9)
```

	MedInc	HouseAge	AveRooms	AveBedrms	Population	AveOccup	Latitude	Longitude	label
0	8.3252	41.0	6.984127	1.023810	322.0	2.555556	37.88	-122.23	4.526
1	8.3014	21.0	6.238137	0.971880	2401.0	2.109842	37.86	-122.22	3.585
2	7.2574	52.0	8.288136	1.073446	496.0	2.802260	37.85	-122.24	3.521
3	5.6431	52.0	5.817352	1.073059	558.0	2.547945	37.85	-122.25	3.413
4	3.8462	52.0	6.281853	1.081081	565.0	2.181467	37.85	-122.25	3.422

Fitting model and predicting

```python
# Instantiating the prediction model - an extra-trees regressor
from sklearn.ensemble import ExtraTreesRegressor
reg = ExtraTreesRegressor(random_state=50)
# Fitting the predictino model onto the training set
reg.fit(X_train, y_train)
# Checking the model's performance on the test set
print('R2 score for the model on test set =', reg.score(X_test, y_test))
```

```
R2 score for the model on test set = 0.798174451576825
```

Importing model for lime tabular explainer

```python
# Importing the module for LimeTabularExplainer
from lime import lime_tabular
# Instantiating the explainer object by passing in the training set,
# and the extracted features
explainer_lime = lime_tabular.LimeTabularExplainer(X_train,
                                                   feature_names=features,
                                                   verbose=True,
                                                   mode='regression')
```

Visualization the explainer

```python
# Index corresponding to the test vector
i = 10
# Number denoting the top features
k = 5
# Calling the explain_instance method by passing in the:
# 1) ith test vector
# 2) prediction function used by our prediction model('reg' in this case)
# 3) the top features which we want to see, denoted by k
exp_lime = explainer_lime.explain_instance(
  X_test[i], reg.predict, num_features=k)
# Finally visualizing the explanations
exp_lime.show_in_notebook()
```

Conclusion: Lime explainability represents a significant advancement in the quest for understanding complex machine learning models. By providing local, interpretable explanations for individual predictions, Lime helps bridge the gap between black box models and human comprehension. Its applications extend beyond mere explanation, enabling bias detection, model debugging, and regulatory compliance. As AI continues to play an increasingly significant role in our lives, the need for explainability becomes paramount. Lime is a powerful tool in the pursuit of transparency and accountability in AI systems, empowering users to make informed decisions and build fair and reliable machine learning models.

If you want to read more article like this, you can refer to aiensured-blogs.

References:

https://arxiv.org/pdf/1602.04938.pdf

https://medium.com/analytics-vidhya/model-interpretability-lime-part-2-53c0f5e76b6a

https://www.kaggle.com/code/prashant111/explain-your-model-predictions-with-lime

https://www.steadforce.com/blog/explainable-ai-with-lime

https://www.analyticsvidhya.com/blog/2022/07/everything-you-need-to-know-about-lime/