Projeto Final de Programação

Relatório Técnico

PUC-Rio
Departamento de Informática
Pós-Graduação em Informática

Professora

Clarisse Sieckenius de Souza

Orientador
Simone Barbosa

Aluno
Bianca Moreira Cunha - 2220956

Turma
2023.2

Data de Entrega
08 de dezembro de 2023

Sumário

1. Introdução

Este relatório apresenta o trabalho desenvolvido para a disciplina de Projeto Final de Programação. Neste trabalho foi desenvolvida a biblioteca ExpViz, uma biblioteca python com o objetivo de reunir em um pacote implementações métodos de explicação de modelos de machine learning que vêm sendo muito utilizados na comunidade de XAI (Explainable AI), como SHAP e LIME. A ideia é facilitar o acesso a diferentes métodos para que esses possam ser comparados e analisados de forma mais ágil e também de iniciar uma tentativa de melhoria da interpretabilidade, uma vez que junto com as visualizações, a biblioteca também gera explicações verbais parametrizadas para a predição ou o modelo.

2. Descrição e objetivos gerais do Software

A área de Explainable AI vem crescendo significativamente nos últimos anos, por conta da necessidade de tornar possível interpretar e entender decisões tomadas por modelos chamados de caixa-preta, como redes neurais, principalmente quando eles são utilizados para questões sensíveis. Alguns métodos para melhorar a interpretabilidade de modelos foram propostos, mas há uma falta de avaliação para entender se as suas saídas realmente ajudam a melhorar a interpretabilidade ou não.

Tendo isso em mente, a motivação para a criação da biblioteca ExpViz foi inicialmente para facilitar o trabalho de pesquisadores que, como eu, estão estudando como os métodos de explicação existentes podem ou não ajudar nessa questão, oferecendo um acesso mais fácil para esses métodos juntando eles em um lugar e utilizando uma mesma sintaxe para gerar explicações. Além disso, há a tentativa de dar um passo na direção de melhorar a interpretabilidade dessas explicações, ao gerar explicações verbais para elas de forma parametrizada para que seja personalizada para os dados do usuário. Assim, além de facilitar o trabalho de pesquisadores, a biblioteca também pode ajudar usuários que queiram entender os

resultados de modelos treinados para domínios específicos e têm dificuldade de interpretar as visualizações existentes.

3. Especificação de requisitos do software

Considerando o contexto de Explainable AI em que a solução proposta está inserida e os seus objetivos, foram definidos os requisitos necessários para o seu desenvolvimento. Nesta seção então serão apresentados os requisitos funcionais e não funcionais do software.

3.1. Requisitos funcionais

[RF1] O sistema deve permitir que o usuário escolha um dos métodos conhecidos de explicabilidade de modelos de machine learning, fornecendo um modelo treinado por ele e suas features

[RF2] O sistema deve ser capaz de exibir visualizações das explicações geradas pelo método selecionado pelo usuário.

[RF3] O sistema deve ser capaz de fornecer uma explicação textual do que está sendo exibido na visualização para auxiliar o usuário a entendê-la.

[RF4] O sistema deve ter um atributo que explique ao usuário como cada tipo de visualização disponível funciona e quais são os cenários em que cada uma é mais adequada.

[RF5] O sistema deve possibilitar que o usuário gere visualizações e explicações de diferentes métodos de explicação utilizando uma mesma a simples sintaxe.

3.2. Requisitos não-funcionais

[RNF1] O sistema deve ser disponibilizado como uma biblioteca python para poder ser utilizada preferencialmente em notebooks.

[RNF2] O sistema deve ser escalável, de forma que novas classes de métodos de explicação ou novas classes de visualização possam ser adicionadas facilmente.

[RNF3] O sistema deve ser implementado de forma que possibilite fácil manutenção e atualização, seguindo as melhores práticas da Engenharia de Software.

4. Descrição da arquitetura

ExpViz é composta pelos pacotes *exp_explainers* e *visualizations*. Cada um contém uma classe abstrata base e classes que implementam os seus métodos. No pacote *exp_explainers* a classe abstrata base se chama *Explainer*, e é dela que as classes de cada método de explicação vão herdar para ter suas implementações específicas. Cada classe de método de explicação terá o seu conjunto de visualizações. As visualizações vêm do pacote *visualizations*, que contém a classe abstrata base *Plot*. Cada classe de visualização tem uma função que irá exibir tanto o

gráfico da explicação quanto a explicação textual parametrizada para ajudar na interpretação da visualização, considerando os dados fornecidos pelo usuário.

Essa estrutura faz com que a solução seja escalável, uma vez que é possível adicionar novos métodos de explicação e novos tipos de visualização de forma fácil, sem que os que já existem tenham que ser alterados.

4.1. Diagrama de classes

O diagrama de classes da arquitetura descrita anteriormente está apresentado na Figura 1. Conforme mencionado anteriormente, existem duas classes abstratas base e classes concretas que herdam delas.

A primeira classe base é a *Explainer*, que lista as funções necessárias para gerar explicações, exibir suas visualizações e explicações textuais, além de listar as opções de visualizações disponíveis para uma classe concreta específica e os objetivos de uso de cada visualização (a função *get_visualization_objective* tem o intuito de facilitar a escolha da visualização mais adequada para o objetivo do usuário, ao explicar para ele como ela funciona e quais informações pode tirar dela). Nesta primeira versão de ExpViz, foram implementadas as classes *SHAPExplainer* e *LIMEExplainer*, que geram explicações a partir dos métodos SHAP e LIME, já conhecidos e vastamente utilizados no contexto de XAI. Cada uma implementa as funções listadas na classe abstrata de acordo com as funções das bibliotecas já existentes que implementam os métodos SHAP e LIME, assim unificando a geração de explicações e exibição das visualizações em funções únicas. As classes concretas têm também atributos de instância que recebem do usuário o modelo treinado, as observações recebidas pelo modelo para realizar predições, e, no caso de explicações locais (para uma predição específica), o valor numérico do índice onde a instância desejada se encontra no conjunto de observações recebido.

A segunda classe abstrata base é a classe *Plot*. Existem três atributos de instância nessa classe, que recebem o modelo treinado, o tipo de visualização desejada e, no caso de explicações locais (para uma predição específica), o valor numérico do índice onde a instância desejada se encontra no conjunto de observações recebido. Para os dois métodos de explicação disponíveis SHAP e LIME, temos quatro possíveis visualizações, três para o SHAP e uma para o LIME. Essas são visualizações que já estão disponíveis nas bibliotecas existentes de cada método, e foi decidido, para fins de simplificação, utilizá-las nessa primeira versão, possibilitando a implementação de novas visualizações não presentes nas bibliotecas no futuro. Cada classe concreta representa uma visualização e implementa funções para gerar a visualização da explicação assim como a explicação textual dela. Cada uma tem também um atributo de classe que contém um texto explicando como aquela visualização funciona e quais informações ela fornece, que pode ser acessado pela função *get_visualization_objective*, mencionada anteriormente.

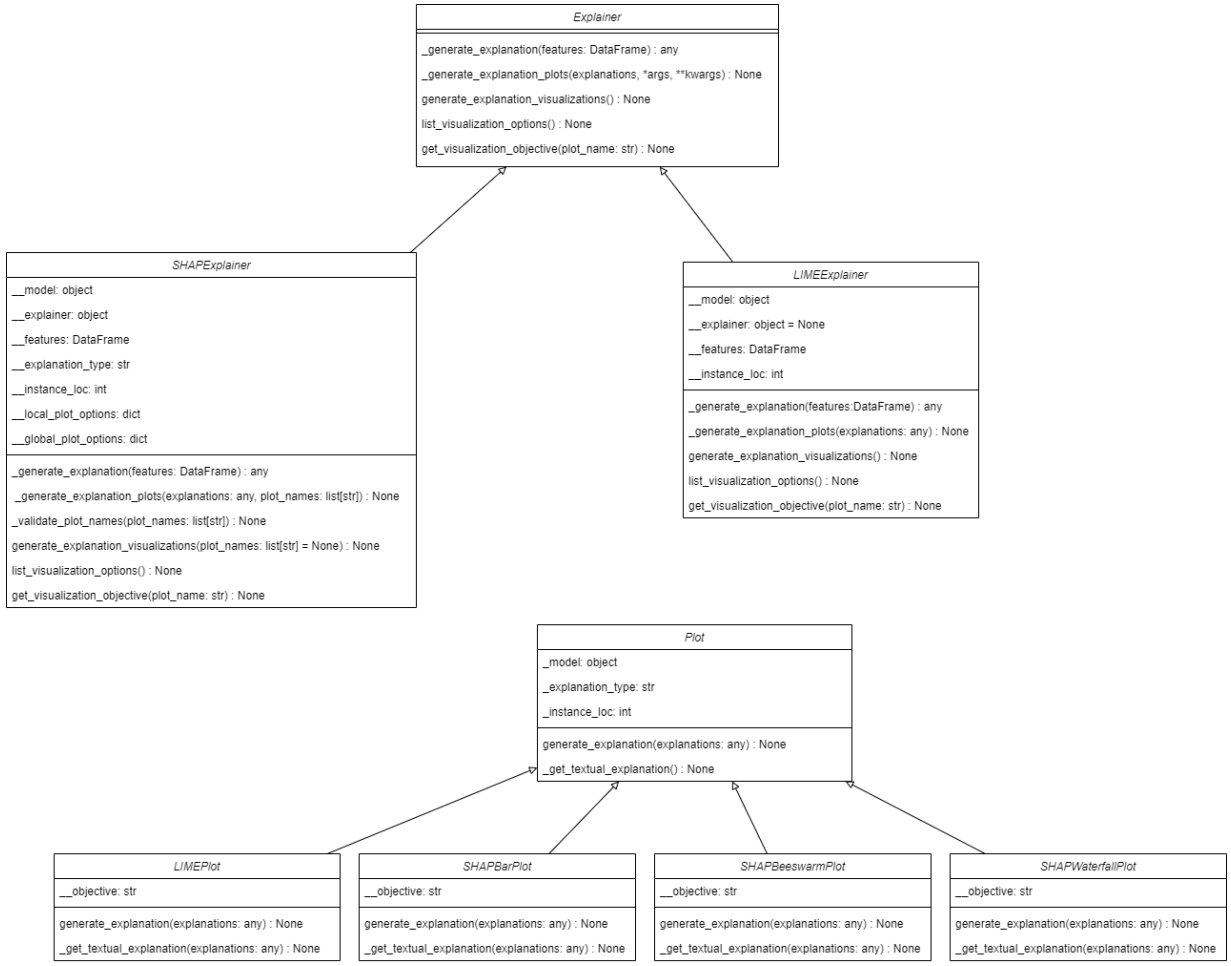

Figura 1. Diagrama de classes

5. Manual de utilização

Como ExpViz é uma solução voltada para desenvolvedores, o manual de utilização está apresentado no arquivo README.md no repositório onde a aplicação se encontra e que será acessado para a instalação da biblioteca.

6. Cenários de uso do software

Nesta seção serão apresentados alguns cenários de uso do software, tanto casos em que o usuário tem sucesso, descritos na seção 5.1, quanto cenários em que ele encontra problemas, descritos na seção 5.2.

6.1. Cenários de uso com sucesso

6.1.1. Cenário 1

Um cientista de dados pode utilizar a biblioteca ExpViz para gerar explicações para um modelo de machine learning que ele esteja utilizando ou desenvolvendo com o intuito de entender como o seu modelo se comporta ou como cada atributo do

modelo impacta predições específicas. Para isso ele irá instalar a biblioteca no seu ambiente de trabalho e utilizar um IPython notebook para suas análises. Dentro do notebook ele irá importar a biblioteca ExpViz e outras necessárias, irá obter os dados e construir um modelo de aprendizado de máquina. Ele irá então instanciar um explainer (SHAPExplainer ou LIMEExplainer) de sua preferência, fornecendo para ele o modelo treinado e o conjunto de dados sobre o qual ele deseja ter explicações. A partir do explainer instanciado, o usuário poderá verificar quais visualizações estão disponíveis para ele e o objetivo de cada uma utilizando as funções *list_visualization_options* e *get_visualization_objective*. Ao obter a descrição do funcionamento de cada visualização ele poderá escolher a visualização mais adequada para o seu objetivo e irá gerar a visualização a partir da função *generate_explanation_visualizations*. A visualização será exibida juntamente com uma explicação textual das informações que ela está fornecendo, e assim o usuário poderá entender o comportamento do seu modelo de forma global ou o impacto de cada atributo em uma predição específica escolhida por ele.

6.1.2. Cenário 2

Um pesquisador que esteja estudando a interpretabilidade de explicações geradas por métodos de explicação de modelos de machine learning, como SHAP e LIME, pode utilizar a biblioteca ExpViz para gerar explicações e visualizações de diferentes métodos existentes na literatura com facilidade, já que estão concentrados em uma biblioteca. Para isso ele irá instalar a biblioteca no seu ambiente de trabalho e utilizar um ipython notebook para suas análises. Dentro do notebook ele irá importar a biblioteca ExpViz e outras necessárias, irá obter os dados e construir um modelo de aprendizado de máquina. Ele irá então instanciar os explainers dos métodos que ele deseja avaliar se geram uma melhoria para a interpretabilidade dos modelos ou não, fornecendo para eles o modelo treinado e o conjunto de dados utilizado. A partir dos explainers instanciados, o usuário poderá verificar quais visualizações estão disponíveis utilizando a função *list_visualization_options* e escolher quais visualizações quer utilizar em seu estudo. O pesquisador irá gerar as visualizações desejadas a partir da função *generate_explanation_visualizations* e elas será exibida juntamente com uma explicação textual das informações que ela está fornecendo. Assim o usuário poderá utilizar as visualizações, assim como as explicações textuais, em seu estudo para avaliar se elas auxiliam ou não no entendimento do comportamento do modelo.

6.2. Cenários de uso com problemas

6.2.1. Cenário 1

Um cientista de dados que queira entender o comportamento de um modelo de machine learning que está utilizando, mas que não esteja familiarizado com o seu dataset e os atributos utilizados para treinar o modelo, pode ter dificuldade para entender as saídas das funções da biblioteca ou pelo menos para fazer uso adequado dessa informação que ele recebe. Inicialmente ele irá instalar a biblioteca

no seu ambiente de trabalho e utilizar um ipython notebook para suas análises. Dentro do notebook ele irá importar a biblioteca ExpViz e outras necessárias, irá obter os dados e construir um modelo de aprendizado de máquina. Ele irá então instanciar um explainer (SHAPExplainer ou LIMEExplainer) de sua preferência, fornecendo para ele o modelo treinado e o conjunto de dados sobre o qual ele deseja ter explicações. A partir do explainer instanciado, o usuário poderá verificar quais visualizações estão disponíveis para ele e o objetivo de cada uma utilizando as funções list_visualization_options e get_visualization_objective. Como ele não tem conhecimento sobre o seu dataset, ele terá dificuldade em escolher qual visualização é a mais adequada, já que ao não conhecer os atributos, ele não vai saber como eles podem se relacionar entre si e com a saída, e não terá uma visão clara de que informações quer tirar da explicação. O ideal é que o usuário estude o seu dataset e entenda os atributos e as relações entre eles antes de gerar o modelo e explicações para ele, para que ele tenha em mente o seu objetivo ao gerar explicações e assim saiba escolher a visualização mais adequada para que a informação entregue seja utilizada de forma otimizada.

6.2.2. Cenário 2

Um usuário que tente usar a biblioteca ExpViz e queira utilizar um método de explicação ainda não disponibilizado pela biblioteca pode acabar se frustrando por não encontrar o método desejado. Inicialmente ele irá instalar a biblioteca no seu ambiente de trabalho e utilizar um ipython notebook para suas análises. Dentro do notebook ele irá importar a biblioteca ExpViz e outras necessárias, irá obter os dados e construir um modelo de aprendizado de máquina. Ele então precisará instanciar um explainer de sua preferência, porém vai perceber que o método que gostaria de utilizar não está disponível. Ele precisará utilizar um outro método, que já esteja incluído na biblioteca, ou não conseguirá gerar as suas explicações. Outra opção seria entrar em contato com o suporte da biblioteca e solicitar que o método que deseja seja incluído na solução.

7. Considerações finais

Este relatório apresentou a biblioteca ExpViz, criada como parte de um estudo que tem o propósito de avaliar se explicações geradas por métodos conhecidos como SHAP e LIME realmente geram uma melhora na interpretabilidade de modelos de machine learning, conforme afirmado por seus criadores. O objetivo da criação da biblioteca foi facilitar o acesso a diferentes métodos de explicação de modelos de machine learning existentes na literatura, já que alguns deles já possuem implementação em python, mas têm formas de uso bastante distintas.

Inicialmente, a finalidade da biblioteca foi facilitar esse acesso para que explicações pudessem ser geradas de forma mais ágil para serem utilizadas em um estudo que avalia se elas auxiliam no entendimento do comportamento de modelos. Ela pode ser útil para diversos estudos que precisem gerar uma variedade de

explicações ou que queiram comparar diferentes métodos de explicação. Porém, ela pode também ser utilizada por cientistas de dados que desejam ter fácil acesso a diferentes métodos de explicação de modelos de machine learning para selecionar o mais adequado para fornecer explicações para seus modelos.

Esta é uma versão inicial da biblioteca, que cobre apenas dois métodos de explicação, SHAP e LIME, e que contém somente as visualizações que as implementações já existentes deles já forneciam. Porém, a arquitetura da aplicação foi desenvolvida de forma que novos métodos e novas visualizações possam ser incluídos e implementados de forma ágil, sem que os anteriores precisem ser alterados.

8. Referências

Lundberg, S. M., & Lee, S. I. (2017). **A unified approach to interpreting model predictions**. Advances in neural information processing systems, 30.

RADEČIĆ, D. LIME: **How to Interpret Machine Learning Models With Python**. Disponível em: <https://towardsdatascience.com/lime-how-to-interpret-machine-learning-models-with-python-94b0e7e4432e>. Acesso em: 08 dez. 2023

Ribeiro, M. T., Singh, S., & Guestrin, C. (2016, August). **" Why should i trust you?" Explaining the predictions of any classifier**. In Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining (pp. 1135-1144).

TREVISAN, V. **Using SHAP Values to Explain How Your Machine Learning Model Works**. Disponível em <https://towardsdatascience.com/using-shap-values-to-explain-how-your-machine-learning-model-works-732b3f40e137>. Acesso em: 08 dez. 2023