Summary on the following Papers

1. **A historical perspective of explainable Artificial Intelligence**

   Summary: The term "explainable AI" (XAI) refers to the approaches and methodologies used to create AI programs that allow users to "understand why" they take certain actions. In other words, a system is deemed a XAI system if we can obtain explanations from it regarding its internal logic. We will discuss why explainability, a novel characteristic, started to acquire appeal in the AI field in recent years. Roberto and Ed give a historical overview of explainable artificial intelligence in this paper. They converse on how explainability has evolved over time, how it is perceived today, and how it might be perceived in the future. A rule interpreter or reasoner that utilizes the knowledge base, an interface via which the user can query the system for knowledge, and a knowledge base that stores the domain knowledge make up an expert system. Explanations are typically interpreted in one of two ways in the literature on expert systems: either as a line of reasoning or as a problem-solving activity. While some machine learning models, such as decision rules, decision tables, and decision trees, can be considered interpretable by design, the majority of machine learning models operate as "black boxes." A black-box, when given an input, gives the classification, prediction, suggestion, etc. result of a decision job, but it does not expose enough information about its internal behavior, leading to an opaque decision model. For this reason, the task of constructing an interpretable model that closely resembles the black-box model while often aiming for high fidelity is how explainability in machine learning is defined. There is a huge amount of literature on machine learning that is explicable or interpretable and they discuss over a survey. They review the literature on explainability in AI and give a historical overview of how the concept of explanation has been conceived from traditional to more recent perspectives, specifically in the context of expert systems, machine learning, recommender systems, and neural-symbolic learning and reasoning. They go over several concepts of explanations, examples of them, as well as their characteristics and metrics for judging them. As a result, the reader can "travel" through many explanation concepts and get a deeper understanding of the issue of explainable AI by using the article's extensive list of references.

2. **Captum: A unified and generic model interpretability library for PyTorch**

   Summary: In this paper, Narine and Ed introduce a brand-new PyTorch model interpretability library that is open-source and unified. A collection of feature, neuron, and layer significance algorithms, commonly known as gradient and perturbation-based attribution algorithms, are generically implemented in the library together with a set of metrics for assessing their performance. It can be applied to models for classification and non-classification, such as graph-structured models built on neural networks (NN). In this work, they provide a high-level overview of the attribution methods that are supported and demonstrate how to compute in a memory-efficient and scalable manner. The algorithm is broken down into three key categories: primary, neuron, and layer attributions. They provide an overview of the algorithm. Large-sized inputs are frequently used by state-of-the-art neural networks with many model parameters, which ultimately results in computationally expensive forward and backward passes. When doing attribution, they want to make sure they effectively utilize the memory and CPU/GPU resources that are available. Certain algorithms, especially those with internal input expansion, split inputs into smaller chunks, conduct the computations sequentially on each piece, and aggregate the resulting attributions to prevent out of memory situations. Many quantitative evaluation measures rely on priors of a dataset domain, and human annotations based on visual perception are frequently arbitrary. They implement two measures, maximal sensitivity and infidelity, in a general way that can be used to every PyTorch model and the majority of Captum methods. They can carry out several perturbations simultaneously, like other algorithms, which eventually enhances runtime performance. Finally, they go through many applications, including text classification and regression and, most significantly, multimodality, and introduce the Captum Insights model debugging tool. One of Captum library's primary goals is to support multi-modal neural networks. As a result, we are able to use Captum with machine-learning models that are constructed using features derived from many sources, such as audio, video, picture, text, category, or dense variables.