Proveniência em Feature Stores

João Gabriel Rodrigues Silva[1], **Daniel dos Santos Kaster**[1]

[1]Departamento de Computação – Universidade Estadual de Londrina (UEL)
Caixa Postal 10.011 – CEP 86057-970 – Londrina – PR – Brasil

joaogabriel.r1@uel.br, dskaster@uel.br

Abstract. *Machine learning is an area of artificial intelligence that seeks to develop learning systems capable of independently acquiring knowledge through past experiences. Data preparation is essential for machine learning. During this stage, it is very common to use several techniques to create features that best fit the problem, and it is important to record the techniques used in the processing of each feature. Feature Store is a recent technology that helps in several stages in the development of machine learning, having as its greatest utility the storage and reuse of features without the need to redo the techniques applied during its creations. To ensure an effective and safe reuse of features, it is essential to use the concept of provenance, a concept that consists of storing the origin and transformations of data over time. The objective of this work is to analyze the mechanisms provided by the Feature Stores currently available that are used to capture provenance, in order to identify the disparity between the available implementations and the solutions proposed in the provenance literature. In particular, this work focuses on open source Feature Stores, in order to indicate their potential and limitations in terms of provenance. The work also seeks to propose an instantiation and use of an open source Feature Store, intensively adopting provenance strategies.*

Resumo. *Aprendizado de máquina é uma área da inteligência artificial que busca desenvolver sistemas de aprendizado capazes de adquirir conhecimento de forma independente através de experiências passadas. O preparo de dados é essencial para o aprendizado de máquina. Durante essa etapa, é muito comum a utilização de diversas técnicas para a criação de características que melhor se adequem ao problema, sendo importante o registro das técnicas utilizadas no processamento de cada característica. Feature Store é uma tecnologia recente que auxilia em diversas etapas no desenvolvimento de aprendizado de máquina, tendo como sua maior utilidade o armazenamento e reúso de características sem a necessidade de refazer as técnicas aplicadas durante suas criações. Para garantir um reúso eficaz e seguro de características é fundamental utilizar do conceito de proveniência, um conceito que consiste em armazenar a origem e as transformações ao longo do tempo de um dado. O objetivo deste trabalho é analisar os mecanismos fornecidos pelas Feature Stores disponíveis atualmente que são utilizados para capturar proveniência, visando identificar a disparidade entre as implementações disponíveis e as soluções propostas na literatura de proveniência. Em particular, este trabalho concentra-se em Feature Stores de código aberto, com o intuito de indicar suas potencialidades e limitações no que tange proveniência. O trabalho busca também propor uma instanciação e utilização de Feature Store de código aberto adotando intensivamente estratégias de proveniência.*

1. Introdução

Aprendizado de máquina (*Machine Learning* – ML), é uma área da inteligência artificial que busca métodos computacionais capazes de aprender de acordo com experiências passadas[19]. Esse tópico se mostrou extremamente importante para diversas áreas e cresceu muito com o passar do tempo, crescendo também sua complexidade e necessidade de melhores ferramentas e tecnologias[18]

Conforme o uso do aprendizado de máquina se dissemina, surgem mais ferramentas focadas no desenvolvimento e manutenção de sistemas de aprendizado de máquina. O conceito de MLOps[18] engloba um conjunto de práticas e ferramentas para suportar o desenvolvimento, implantação e execução de sistemas baseados em ML. MLOps é inspirado na noção de DevOps, termo usado para englobar englobar o conjunto, as práticas e ferramentas para suportar o ciclo de vida de desenvolvimento, execução e manutenção de sistemas. MLOps visa simplificar o ciclo de vida de sistemas de ML, acelerar o desenvolvimento e mitigar diversos problemas que surgem devido a características específicas dessa área[13].

Um modelo de ML tipicamente recebe um vetor de características e devolve um ou mais valores indicando uma predição. Cada característica possui uma relevância específica no resultado final, sendo que modelos tendem a perder desempenho com grandes quantidades de características que pouco contribuem para o resultado[7]. Buscando uma melhor eficiência dos modelos de aprendizado de máquina, é comum a utilização de técnicas para a criação de novas características e seleção das características mais relevantes. Tais técnicas fazem parte de um processo mais amplo, popularmente conhecido como engenharia de características.

Devido à natureza do aprendizado de máquina, grandes quantidades de dados são necessárias, onde muitas vezes esses dados passam por transformações para melhor desempenho do modelo. A falta de cuidado com o registro das transformações pode causar problemas de confiabilidade e reprodutibilidade[15]. Para auxiliar nesse aspecto, é comum utilizar-se o conceito de proveniência.

Proveniência, ou linhagem, é um conceito que surgiu inicialmente na comunidade de banco de dados, que consiste basicamente em armazenar a origem de um dado e suas modificações ao longo do tempo[5]. No aprendizado de máquina, proveniência pode ser usada em diferentes etapas do ciclo de vida, sendo mais comum a utilização no preparo de dados e aprendizado de modelos. Devido à importância de proveniência em ML, tem-se buscado a incorporação deste conceito nas ferramentas, seja de maneira explicita ou implícita.

Feature Store é uma tecnologia recente na comunidade de aprendizado de máquina, que consiste em um repositório centralizado para ML, provendo a capacidade de armazenamento, versionamento e compartilhamento de características[12]. Uma Feature Store auxilia em diversas etapas do ciclo de vida de ML, além de ser uma alternativa para alcançar proveniências importantes no aprendizado de máquina. Devido a isso, a utilização de Feature Stores vem sendo cada vez mais adotada.

Por ser um conceito recente que é cada vez mais adotado, as capacidades de uma Feature Store estão evoluindo rapidamente, porém, não existem muitos artigos que estudam o tema. Proveniência em Feature Stores, especificamente, é um tópico muito pouco

abordado na literatura, fazendo-se necessário o desenvolvimento de trabalhos que explorem como Feature Stores utilizam e se beneficiam de proveniência.

O objetivo deste trabalho é realizar uma análise sobre o estado da proveniência nas Feature Stores, a fim de determinar o quão estabelecido o conceito se encontra nas Feature Stores. Os resultados esperados deste trabalho incluem detalhar como diferentes Feature Stores utilizam proveniência, além de apresentar uma instanciação de uma Feature Store com proveniência, com exemplos de uso ilustrativos.

Este documento está organizado da seguinte maneira. A seção 2 apresenta os fundamentos necessários para o entendimento deste trabalho. A seção 3 mostra trabalhos na literatura relacionados com este trabalho. A seção 3 detalha os objetivos do trabalho. A seção 4 mostra o planejamento feito para alcançar os objetivos do trabalho.

2. Fundamentação Teórico-Metodológica e Estado da Arte

2.1. Aprendizado de máquina

Aprendizado de máquina é uma área da inteligência artificial que busca desenvolver sistemas de aprendizado capazes de adquirir conhecimento de forma independente através de experiências passadas[19].

Existem diferentes categorias de aprendizado de máquina, sendo a mais comum a de aprendizado supervisionado[19]. Durante o aprendizado supervisionado, um modelo recebe um vetor de características (também chamadas de features) como entrada e retorna uma ou mais predições como saída. Após a predição, a saída do modelo é comparada com a saída correta, também fornecida pela entrada, e os parâmetros do modelo são ajustados para próximas predições. Quando o conjunto de valores de saída é discreto, o problema é chamado de classificação, sendo comum buscar pela maior taxa de acerto possível. Quando o conjunto de valores de saída é contínuo, o problema é chamado de regressão, onde busca-se minimizar o erro obtido.

Outra categoria existente é a de aprendizado não supervisionado. Nesse caso, durante o aprendizado, apenas o vetor de características é fornecido, e o modelo tenta agrupar entradas semelhantes em diferentes conjuntos. Essa categoria é normalmente utilizada para descobrir novas observações sobre os dados ou para o agrupamento de dados sem a necessidade de observação humana[19].

Existe ainda a categoria de aprendizado semi-supervisionado. Combinando entradas rotuladas (com uma saída correta definida) e não rotuladas, essa categoria é um meio-termo entre aprendizado supervisionado e não supervisionado, sendo normalmente utilizada em situações de grande quantidade de dados de entrada, porém com poucos dados rotulados[16].

O processo de aprendizado de um modelo é um processo iterativo onde os dados utilizados para o aprendizado são fundamentais. Durante o aprendizado de um modelo, são realizados diversos ajustes com a finalidade de melhorar a precisão de um modelo, sendo comum a utilização de técnicas de engenharia de características e estratégias de gerenciamento das características obtidas.

2.2. Engenharia de Características

O preparo de dados é essencial para o aprendizado de máquina, sendo muitas vezes a etapa mais demorada no desenvolvimento de um sistema de aprendizado de máquina[12]. Um modelo tem sua precisão diretamente relacionada com as características utilizadas em sua entrada, quanto mais características relevantes para o modelo, mais eficiente o modelo fica[7]. Buscando obter características relevantes para um modelo específico, existem diversas tipos de técnicas que podem ser utilizadas para expandir ou reduzir o total de características, de forma a utilizar os dados disponíveis de maneira mais eficiente.

Construção de características se refere ao processo de obter novas características relacionando características já existentes. Construção de características aumenta o poder de expressão das características originais e é normalmente utilizada para expandir a dimensionalidade das características[8].

Extração, ou transformação, de características agrupa características existentes, criando novas características enquanto tenta manter os dados existentes nas características antigas. A extração de características é uma das maneiras de reduzir a quantidade total de características[6][17].

Figura 1. Processo de extração de características. Fonte: [17]

Seleção de características é outra técnica utilizada para reduzir o total de características. Diferente da extração, a seleção não cria nenhuma nova característica, buscando apenas selecionar as características mais relevantes de acordo com um determinado critério. Exitem 3 tipos de seleções de características: supervisionada, não supervisionada e semi-supervisionada[1].

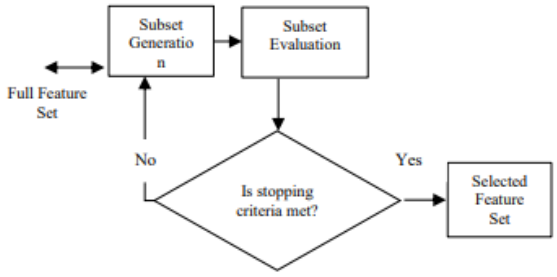

Figura 2. Processo de seleção de características. Fonte: [17]

No preparo de dados, é comum a combinação de várias técnicas para a criação de características adequadas. Durante essa etapa, é importante o armazenamento das características obtidas e a documentação das técnicas utilizadas, de forma a garantir um reúso seguro dessas características sem a necessidade de refazer todo o processo de preparação de dados e engenharia de características, o que frequentemente acontece quando não há a adoção de ferramentas adequadas para o gerenciamento de bases de características.

2.3. Feature Stores

Feature Store (FS) é um conceito recente na área de aprendizado de máquina, ganhando destaque após sua utilização na plataforma de aprendizado de máquina da Uber[12]. Desde então, a utilização de Feature Stores vêm crescendo rapidamente, com grandes empresas como Amazon, Facebook, Google e diversas outras incorporando suas próprias Feature Stores em suas plataformas de aprendizado de máquina, além do desenvolvimento de Feature Stores de código aberto como Feast[1] e Hopsworks[2].

Uma Feature Store é um repositório centralizado de dados curados para aprendizado de máquina, possuindo três principais componentes[12].

- Armazenamento: pode armazenar características de diversas maneiras, como banco de dados SQL, NoSQL, armazenamento em nuvem, sistemas distribuídos, entre outros.
- Interface: possui uma interface utilizada para acesso das características, definindo como deve ser feita a leitura e escrita de novas características na FS.
- Registro de metadados: armazena todas as informações importantes relacionadas à características, como versões e origem, possibilitando a pesquisa e compartilhamento de características por parte de usuários.

As figuras 3 e 4 mostram uma das vantagens da utilização de Feature Stores. Na figura 3, características de diferentes fontes são utilizadas em diferentes modelos, sendo necessário o processamento dessas características para cada modelo que as utiliza. Na figura 4, as características são processadas apenas uma vez, em seguida são armazenadas na Feature Store, ficando disponíveis para serem utilizadas por qualquer modelo, sem a necessidade de processá-las novamente.

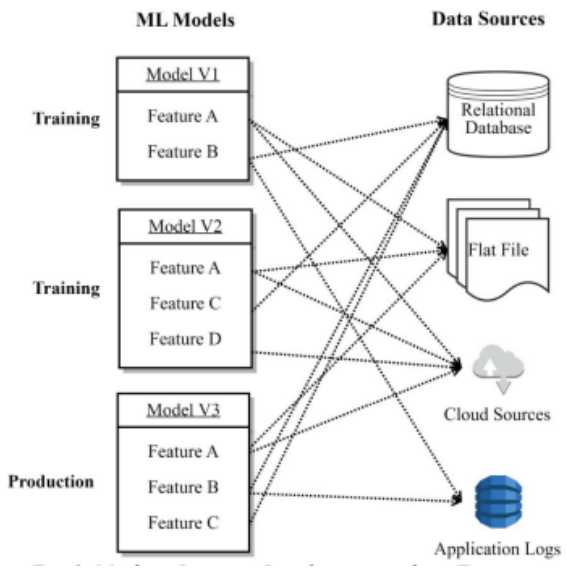

Figura 3. Aprendizado de máquina tradicional. Fonte: [11]

Feature Stores também auxiliam em diversas etapas do desenvolvimento de sistemas de aprendizado de máquina, disponibilizando características para o aprendizado de

[1]https://feast.dev/
[2]https://www.hopsworks.ai/

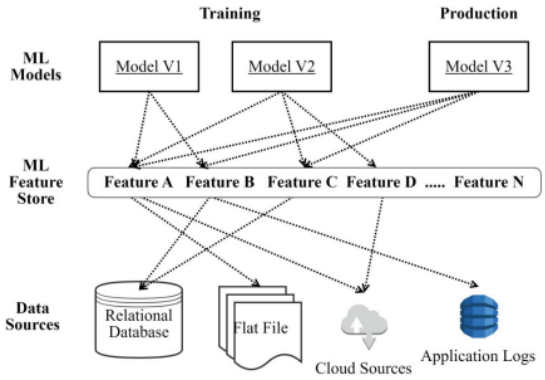

Figura 4. Aprendizado de máquina com o uso de Feature Store. Fonte: [11]

modelos ou predição de modelos em inferência e fornecendo métricas de qualidade que auxiliam na detecção de erros e análise de modelos e características[11].

Feature Stores aceleram uma das etapas mais trabalhosas no desenvolvimento de aprendizado de máquina, o preparo de dados. Em uma Feature Store, características podem ser reutilizadas sem nenhum trabalho necessário para processá-las novamente, e também é possível procurar características relacionadas a alguma característica específica. Para isso, é necessário um registro preciso de como as características foram geradas, sendo importante a utilização de proveniência para garantir a reutilização de características de maneira segura.

2.4. Proveniência

O termo proveniência, ou linhagem, descreve de maneira geral de onde um dado se originou e quais mudanças foram feitas ao longo do tempo[5], sendo uma maneira de descrever o processo de produção de um dado do começo ao fim[4].

Inicialmente, o conceito era utilizado para assunto de pesquisas na comunidade de banco de dados, porém, logo se mostrou útil em outras áreas como segurança e aprendizado de máquina[3].

Um exemplo simples de proveniência pode ser visto na figura 5, que mostra um grafo de transformações[5]. Nesse grafo, transformações T_i são aplicadas em conjuntos de dados de entrada I_j resultando em conjuntos de dados de saída O. A partir do grafo, é possível formular duas questões que são respondidas pela proveniência:

- Dada uma saída O_k, de quais entradas I o conjunto de dados O se originou
- Dada uma saída O_k, quais transformações T geraram o conjunto de dados O

Oferecer recursos que permitam responder a estas perguntas potencializa a confiança em utilizar-se um conjunto de dados coletado e transformado anteriormente em novas tarefas, além de ser fundamental para a interpretação dos dados e de resultados de operações sobre esses dados.

2.4.1. Proveniência em aprendizado de máquina

Um campo de pesquisa que se beneficiou com o conceito de proveniência é o aprendizado de máquina[3]. Ao incluir informações intrínsecas do aprendizado de máquina,

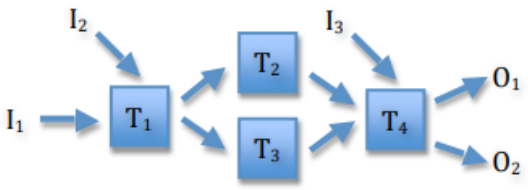

Figura 5. Grafo de transformação. Fonte: [5]

como características e hiper-parâmetros de modelos, a proveniência auxilia no entendimento de resultados, reprodução de modelos e análise de qualidade de dados[3][15][4], demonstrando ser importante para diferentes etapas do ciclo de vida de aprendizado de máquina[14].

Proveniência é normalmente alcançada por meio de duas maneiras. A primeira requer que o usuário especifique explicitamente quando deseja capturar proveniência, possibilitando um uso mais flexível. Essa maneira é normalmente alcançada por meio inserções no código-fonte[14]. A segunda maneira disponibiliza uma proveniência mais genérica, com a vantagem de poder ser feita de maneira automática sem alteração de código-fonte[9].

Durante o preparo dos dados para consumo por um modelo, é comum uma ampla utilização de diferentes técnicas de engenharia de características, incluindo junção de dados de diferentes conjuntos de dados, criação de novas características e transformações em características existentes. A proveniência de características[15] é responsável por manter um registro das técnicas utilizadas para o processamento de uma característica e qualquer atualização que venha a ser feita posteriormente.

Proveniência de modelos[15] é, junto com a proveniência de características, um dos principais tipos de proveniência para o aprendizado de máquina. Ela é a responsável por capturar informações durante o processo de aprendizado de um modelo, como as características e hiper-parâmetros utilizados para o aprendizado, sendo essencial para a reprodução de modelos.

Proveniência é um conceito muito importante para o aprendizado de máquina, com diversas tecnologias fornecendo mecanismos para a utilização deste conceito. Feature Stores especificamente beneficiam-se de proveniência, pois além de necessária para auxiliar na reprodução de modelos, proveniência é fundamental para garantir o reúso das características armazenadas em uma Feature Store. Entretanto a adoção de proveniência em MLOps ainda é um assunto incipiente, demandando trabalhos que explorem de forma ampla proveniência neste contexto, particularmente em Feature Stores.

3. Trabalhos Correlatos

Proveniência é um conceito bem explorado na literatura, com diversos artigos abordando o tema. O artigo de Herschel et al.[4] explica o conceito de proveniência, detalha sua importância e aplicações e apresenta quatro tipos de proveniência, com um detalhamento mais profundo em dois tipos específicos: proveniência de dados e proveniência de fluxo de trabalho. Porém, este artigo aborda proveniência de uma maneira mais geral, sem focar

em alguma área específica, como aprendizado de máquina.

O artigo de Souza et al.[14] descreve proveniência em um contexto de aprendizado de máquina, apresentando uma visão detalhada de diferentes etapas no ciclo de vida de sistemas de aprendizado de máquina e como proveniência pode ser utilizada para auxiliar cada etapa do ciclo de vida. O artigo também apresenta uma representação de dados, para proveniência em aprendizado de máquina, PROV-ML, e a utiliza na prática para averiguar sua utilidade e custo de captura. O artigo de Sugimura et al.[15] cita diversas dificuldades encontradas na reprodução de modelos de aprendizado de máquina, entre elas, proveniência de características e de modelos, e também apresta um sistema que busca superar tais dificuldades. Ambos os artigos, porém, não exploram a proveniência em Feature Stores, especificamente.

O artigo de Ormenisan et al.[9] cita brevemente uma maneira tipicamente utilizada para capturar proveniência em aprendizado de máquina, onde é necessário o usuário explicitar quando é desejado obter proveniência, apelidada de proveniência explícita. O artigo também apresenta uma maneira automática de captura de proveniência, sem a necessidade de interação por parte do usuário, chamada de proveniência implícita, utilizada no sistema de arquivos da Feature Store da plataforma Hopsworks. Este artigo, porém, não explica com detalhes o que a proveniência fornecida é capaz de alcançar.

Por ser um conceito recente, existem pouco artigos que abordam Feature Stores. Os artigos de Patel[12] e Orr et al.[11] explicam o conceito de Feature Store, como elas auxiliam no aprendizado de máquina e atuais limitações. O artigo de Cerar et al.[2] analisa diferentes Feature Stores disponíveis atualmente, listando opções de armazenamento e plataformas de instanciação disponíveis. Todos os artigos acima, porém, não exploram a proveniência nas Feature Stores.

O único trabalho encontrado que aborda proveniência em Feature Stores é o artigo de Ormenisan et al.[10]. Ese artigo fala sobre proveniência na Feature Store Hopsworks, como ela é utilizada para reproduzir a execução de pipelines e também ajudar na depuração de erros que podem acontecer durante a execução de um pipeline. Apesar de abordar o tema proveniência especificamente em Feature Stores, o artigo explica somente para uma Feature Store específica, e não apresenta nenhum caso de uso real. Diferentemente desse artigo, a presente proposta de TCC tem como foco uma análise mais aprofundada sobre proveniência em diferentes Feature Stores e também a utilização prática de Feature Stores de código aberto adotando estratégias de proveniência.

4. Objetivos

Este trabalho tem como objetivo geral realizar uma análise sobre os aspectos de proveniência para aprendizado de máquina utilizados em Feature Stores. Como objetivos específicos, esse trabalho busca:

- Verificar a importância do conceito de proveniência para as Feature Stores e como elas se beneficiam disso;
- Comparar mecanismos fornecidos por diferentes Feature Stores que possibilitam a utilização de proveniência, levantando vantagens e limitações;
- Prover uma instanciação de utilização de Feature Store de código aberto com proveniência.

5. Procedimentos metodológicos/Métodos e técnicas

Inicialmente, será feita uma revisão bibliográfica sobre proveniência em aprendizado de máquina, buscando compreender as técnicas utilizadas para obter proveniência, os custos ligados as técnicas e suas dificuldades. Paralelamente, serão realizados desenvolvimentos de modelos de aprendizados de máquinas simples buscando entender melhor o processo de desenvolvimento e as etapas envolvidas.

Após a revisão, será realizado um estudo sobre as Feature Stores de código aberto disponíveis atualmente, a princípio Feast e Hopsworks, a fim de compreender aspectos técnicos e a maneira como cada uma disponibiliza proveniência. Para o estudo, primeiro será explorado o funcionamento de cada Feature Store específica. Após isso, serão realizados pequenos testes na prática com cada Feature Store e inspeção no código-fonte.

As informações de proveniência descobertas pelo estudo das Feature Stores serão comparadas entre si e com outras ferramentas a fim de determinar pontos positivos e negativos de cada Feature Store. Para a comparação serão utilizados conceitos estudados na revisão bibliográfica buscando determinar a abrangência da proveniência fornecida, o custo computacional, entre outros fatores.

Por fim, para solidificar as conclusões obtidas anteriormente, será escolhida uma Feature Store para a realização de um estudo de caso, prefencialmente com conjunto de dados de agronomia, buscando confirmar a utilidade de Feature Stores e a importância da proveniência fornecida pelas Feature Stores no desenvolvimento de sistemas de aprendizado de máquina.

6. Cronograma de Execução

As atividades apresentadas na seção 5 são enumeradas a seguir.

1. Revisão bibliográfica sobre proveniência em aprendizado de máquina.
2. Desenvolvimento de modelos simples de aprendizado de máquina.
3. Estudo e experimentação de cada Feature Store específica.
4. Realização de testes e inspeção de código fonte em cada Feature Store.
5. Classificação da proveniência da cada Feature Store e levantamento de pontos positivos e negativos.
6. Execução do estudo de caso com uma Feature Store estudada.
7. Escrita do Trabalho de Conclusão de Curso.

Tabela 1. Cronograma de Execução

	set	out	nov	dez	jan	fev	mar	abr	mai
Atividade 1	x	x							
Atividade 2		x							
Atividade 3			x						
Atividade 4			x	x					
Atividade 5					x	x			
Atividade 6						x	x		
Atividade 7				x	x	x	x	x	x

7. Contribuições e/ou Resultados esperados

Dentre os resultados esperados com esse trabalho, destaca-se os seguintes:

- Compreender aspectos de Feature Stores que podem se beneficiar de proveniência;
- Disponibilizar uma categorização de implementações específicas de Feature Stores de código aberto que tornam possível a obtenção de proveniência;
- Apresentar vantagens e limitações de Feature Stores de código aberto;
- Apresentar uma instanciação e utilização prática de uma Feature Store de código aberto.

8. Espaço para assinaturas

Londrina, *12 de setembro de 2022*.

Aluno	Orientador

Referências

[1] Jie Cai, Jiawei Luo, Shulin Wang, and Sheng Yang. Feature selection in machine learning: A new perspective. *Neurocomputing*, 300:70–79, 2018.

[2] Gregor Cerar, Blaž Bertalanič, Anže Pirnat, Andrej Čampa, and Carolina Fortuna. On designing data models for energy feature stores. *arXiv preprint arXiv:2205.04267*, 2022.

[3] Boris Glavic et al. Data provenance. *Foundations and Trends® in Databases*, 9(3-4):209–441, 2021.

[4] Melanie Herschel, Ralf Diestelkämper, and Houssem Ben Lahmar. A survey on provenance: What for? what form? what from? *The VLDB Journal*, 26(6):881–906, 2017.

[5] Robert Ikeda and Jennifer Widom. Data lineage: A survey. Technical report, Stanford InfoLab, 2009.

[6] Samina Khalid, Tehmina Khalil, and Shamila Nasreen. A survey of feature selection and feature extraction techniques in machine learning. In *2014 science and information conference*, pages 372–378. IEEE, 2014.

[7] Huan Liu and Hiroshi Motoda. Feature transformation and subset selection. *IEEE Intell Syst Their Appl*, 13(2):26–28, 1998.

[8] Hiroshi Motoda and Huan Liu. Feature selection, extraction and construction. *Communication of IICM (Institute of Information and Computing Machinery, Taiwan)*, 5(67-72):2, 2002.

[9] Alexandru A Ormenisan, Mahmoud Ismail, Seif Haridi, and Jim Dowling. Implicit provenance for machine learning artifacts. *Proceedings of MLSys*, 20, 2020.

[10] Alexandru A Ormenisan, Moritz Meister, Fabio Buso, Robin Andersson, Seif Haridi, and Jim Dowling. Time travel and provenance for machine learning pipelines. In *2020 USENIX Conference on Operational Machine Learning (OpML 20)*, 2020.

[11] Laurel Orr, Atindriyo Sanyal, Xiao Ling, Karan Goel, and Megan Leszczynski. Managing ml pipelines: feature stores and the coming wave of embedding ecosystems. *Proceedings of the VLDB Endowment*, 14(12):3178–3181, 2021.

[12] Jayesh Patel. Unification of machine learning features. In *2020 IEEE 44th Annual Computers, Software, and Applications Conference (COMPSAC)*, pages 1201–1205. IEEE, 2020.

[13] David Sculley, Gary Holt, Daniel Golovin, Eugene Davydov, Todd Phillips, Dietmar Ebner, Vinay Chaudhary, Michael Young, Jean-Francois Crespo, and Dan Dennison. Hidden technical debt in machine learning systems. *Advances in neural information processing systems*, 28, 2015.

[14] Renan Souza, Leonardo Azevedo, Vítor Lourenço, Elton Soares, Raphael Thiago, Rafael Brandão, Daniel Civitarese, Emilio Brazil, Marcio Moreno, Patrick Valduriez, et al. Provenance data in the machine learning lifecycle in computational science and engineering. In *2019 IEEE/ACM Workflows in Support of Large-Scale Science (WORKS)*, pages 1–10. IEEE, 2019.

[15] Peter Sugimura and Florian Hartl. Building a reproducible machine learning pipeline. *arXiv preprint arXiv:1810.04570*, 2018.

[16] Jesper E Van Engelen and Holger H Hoos. A survey on semi-supervised learning. *Machine Learning*, 109(2):373–440, 2020.

[17] Rizgar Zebari, Adnan Abdulazeez, Diyar Zeebaree, Dilovan Zebari, and Jwan Saeed. A comprehensive review of dimensionality reduction techniques for feature selection and feature extraction. *Journal of Applied Science and Technology Trends*, 1(2):56–70, 2020.

[18] Yue Zhou, Yue Yu, and Bo Ding. Towards mlops: A case study of ml pipeline platform. In *2020 International conference on artificial intelligence and computer engineering (ICAICE)*, pages 494–500. IEEE, 2020.

[19] Zhi-Hua Zhou. *Machine learning*. Springer Nature, 2021.