ML Design

Tuesday, 26 April 2022 1:49 PM

https://blog.fennel.ai/p/two-types-of-information-leakage?s=r

https://blog.fennel.ai/p/real-world-recommendation-system?s=r

https://tech.instacart.com/lessons-learned-the-journey-to-real-time-machine-learning-at-in

https://www.instacart.com/company/how-its-made/griffin-how-instacarts-ml-platform-trip
year/

https://huyenchip.com/2020/12/27/real-time-machine-learning.html
https://huyenchip.com/blog/

https://towardsdatascience.com/deploying-ml-models-in-distributed-real-time-data-stream
applications-217954a0b423

https://eugeneyan.com/writing/practical-guide-to-maintaining-machine-learning/

Apache Flink:
https://flink.apache.org

Seldon Core:
https://www.youtube.com/watch?v=L746MuYzX1c
https://github.com/yinondn/seldon-core-tutorial/blob/main/quickstart.md

GPU:
Inside the Matrix: Unraveling the Intricacies of Neural Networks on GPUs

MLOPS:
https://neptune.ai/blog/optimizing-models-for-deployment-and-inference
https://stanford-cs329s.github.io/syllabus.html

https://www.evidentlyai.com/blog/embedding-drift-detection
https://www.evidentlyai.com/mlops-tutorials

stacart-942f3a656af3

led-ml-applications-in-a-

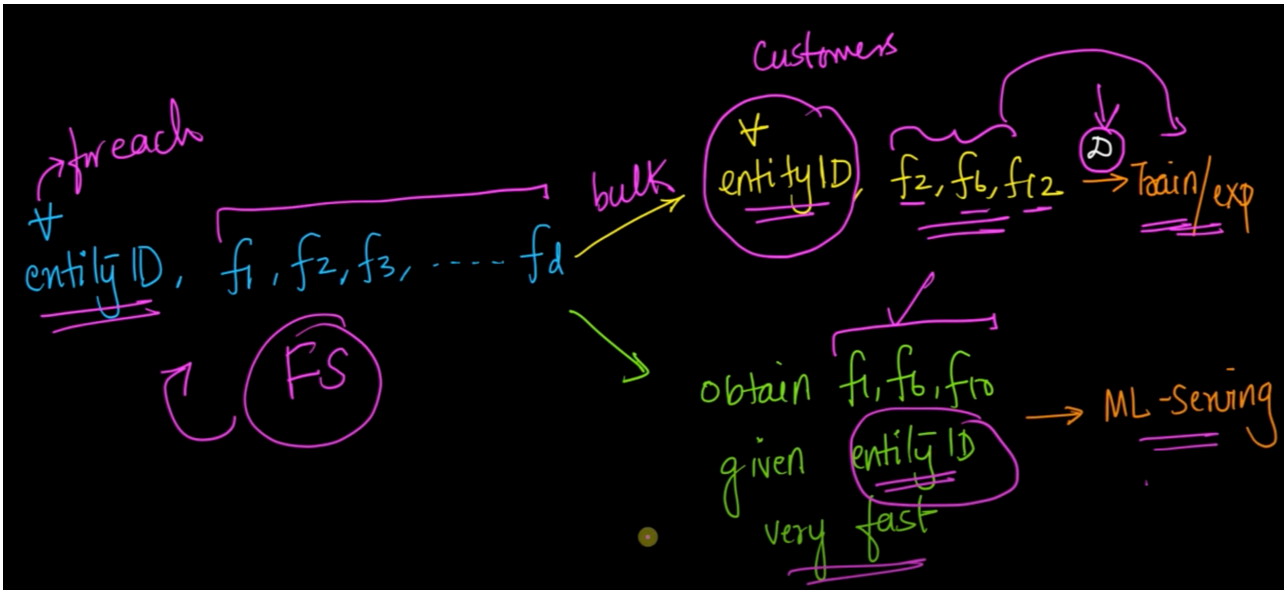
ing-

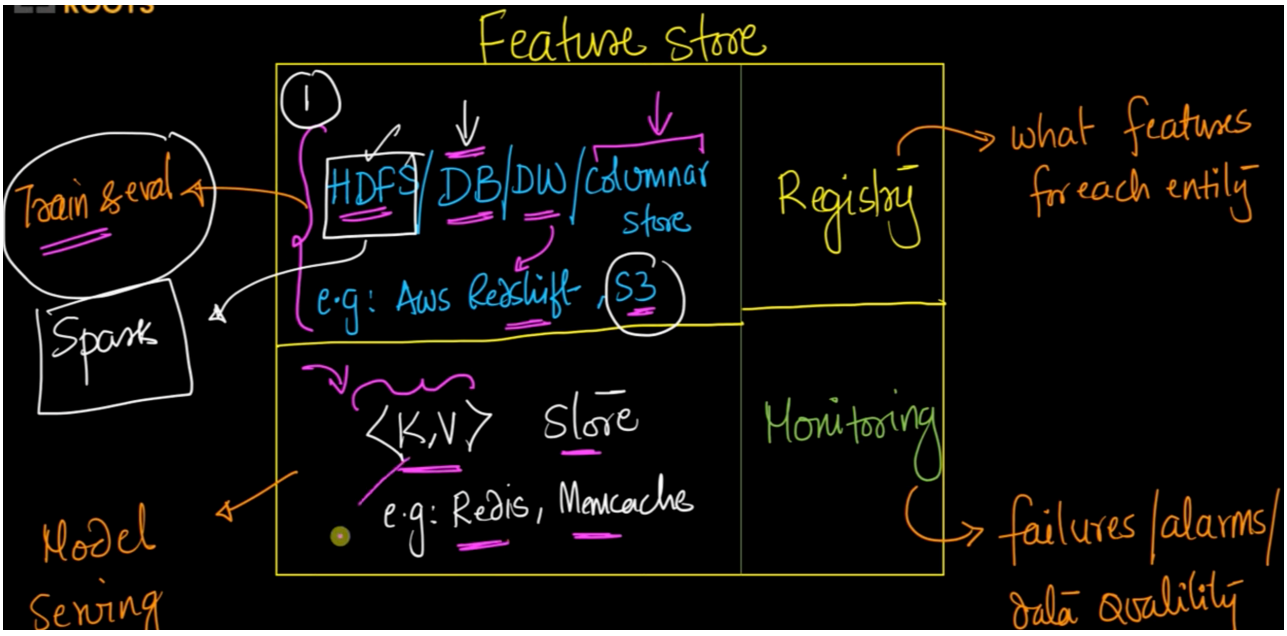
https://bytes.swiggy.com/enabling-large-scale-pytorch-model-inference-9af260538f5f

https://towardsdatascience.com/a-framework-for-building-a-production-ready-feature-eng
f0b29609b20f

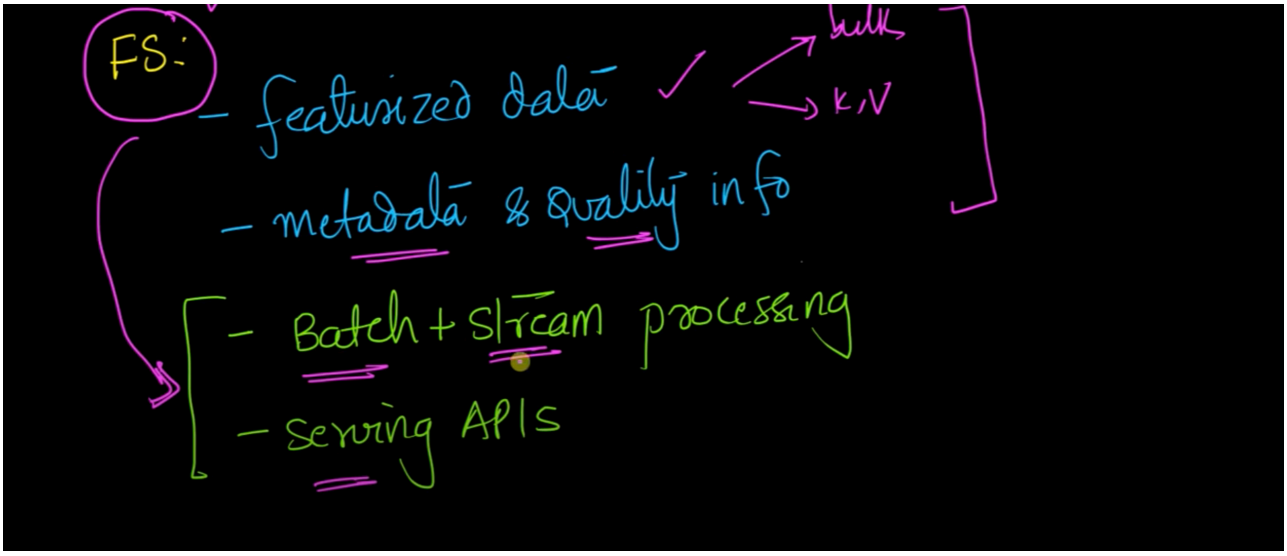
LLMoPS:
https://github.com/tensorchord/Awesome-LLMOps/tree/main

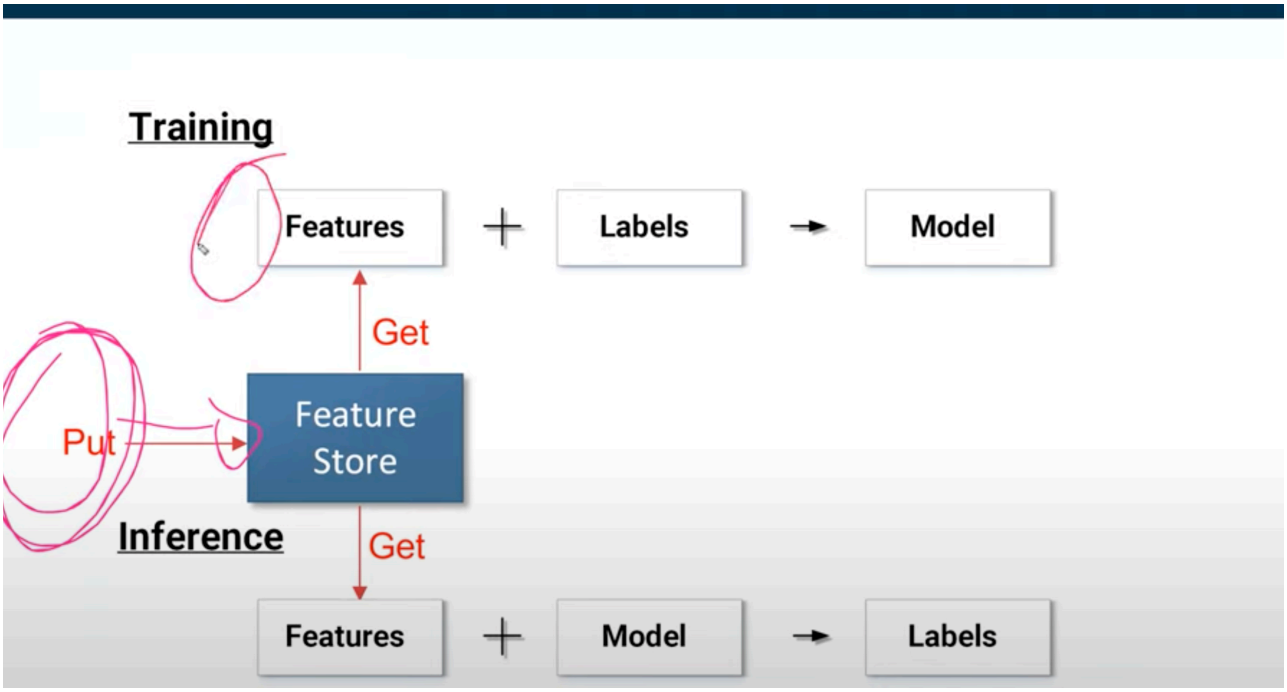
Feature Store:

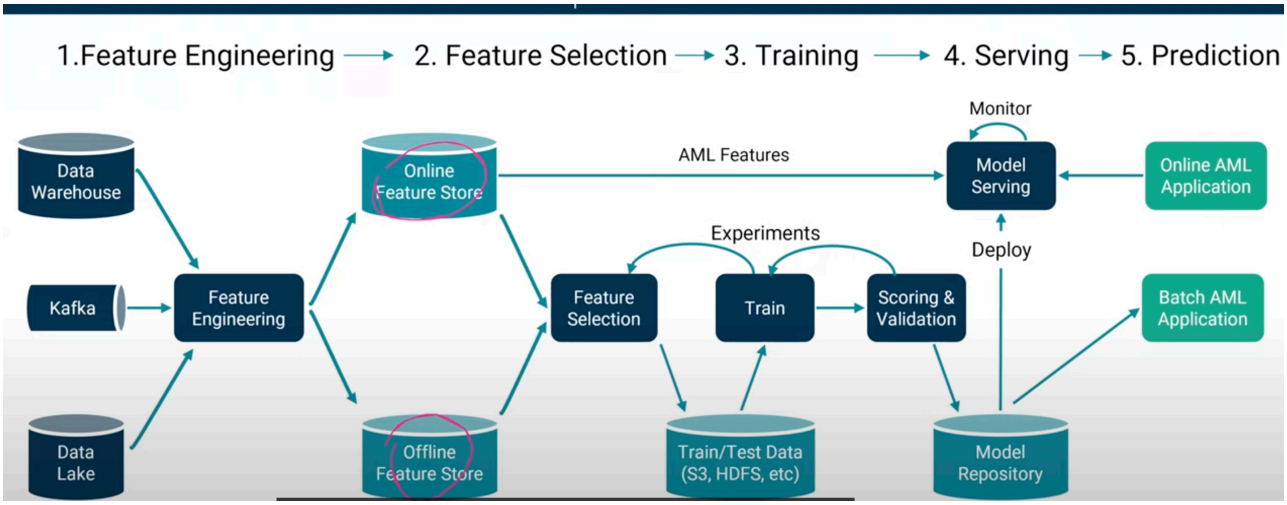


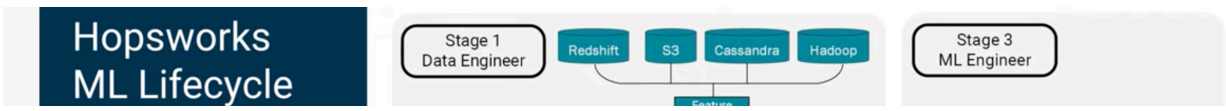


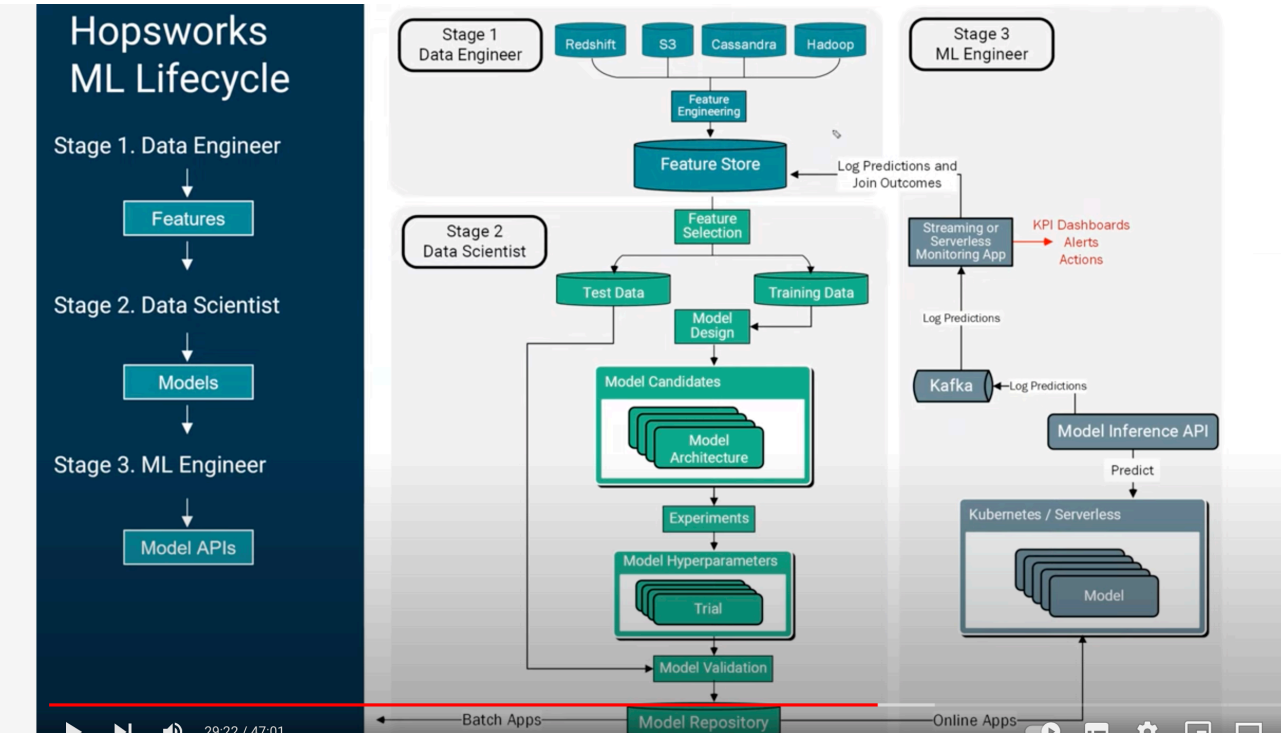
ineering-pipeline-


FS:
- featurized data
- metadata & quality info
- Batch + Stream processing
- serving APIs
bulk
k,v


Training
Features
Labels
Model
Get
Put
Feature Store
Inference
Get
Features
Model
Labels


1.Feature Engineering
2. Feature Selection
3. Training
4. Serving
5. Prediction
Monitor
Data Warehouse
Online Feature Store
AML Features
Model Serving
Online AML Application
Experiments
Deploy
Kafka
Feature Engineering
Feature Selection
Train
Scoring & Validation
Batch AML Application
Data Lake
Offline Feature Store
Train/Test Data (S3, HDFS, etc)
Model Repository


Hopsworks ML Lifecycle
Stage 1 Data Engineer
Redshift
S3
Cassandra
Hadoop
Stage 3 ML Engineer



1) Deployment Types:

➡️ **Batch**:

👉 You apply your trained models as a part of ETL/ELT Process on a given schedule.
👉 You load the required Features from a batch storage, apply inference and save the results to a batch storage.
👉 It is sometimes falsely thought that you can't use this method for Real Time Predictions.
👉 Inference results can be loaded into a real time storage and used for real time applications.

➡️ **Embedded in a Stream Application**:

👉 You apply your trained models as a part of Stream Processing Pipeline.
👉 While Data is continuously piped through your Streaming Data Pipelines, an application with a loaded model continuously applies inference on the data and returns it to the system - most likely another Streaming Storage.
👉 This deployment type is likely to involve a real time Feature Store Serving API to retrieve additional Static Features for inference purposes.
👉 Predictions can be consumed by multiple applications subscribing to the Inference Stream.

➡️ **Request - Response**:

> 👉 You expose your model as a Backend Service (REST or gRPC).
> 👉 This ML Service retrieves Features needed for inference from a Real Time Feature Store Serving API.
> 👉 Inference can be requested by any application in real time as long as it is able to form a correct request that conforms API Contract.
>
> ➡️ **Edge**:
>
> 👉 You embed your trained model directly into the application that runs on a user device.
> 👉 This method provides the lowest latency and improves privacy.
> 👉 Data in most cases is generated and lives inside of device significantly improving the security.

2)
Linear regression assumes that the model residuals (=actual-predicted) are normally distributed.

If the model is underperforming, it may be due to a violation of this assumption.

A QQ plot (short for Quantile-Quantile) is a great way to verify this and also determine the model's performance.

It depicts the quantiles of the observed distribution (residuals in this case) against the quantiles of a reference distribution. Reference is typically the standard normal distribution.

A good QQ plot will:
- Show minimal deviations from the reference line
- NOT reveal trends in residuals

A bad QQ plot will:
- Exhibit significant deviations
- Reveal patterns in residuals
- Show skewness with diverging ends

Thus, the more aligned the QQ plot looks, the more confident you can be about your model.

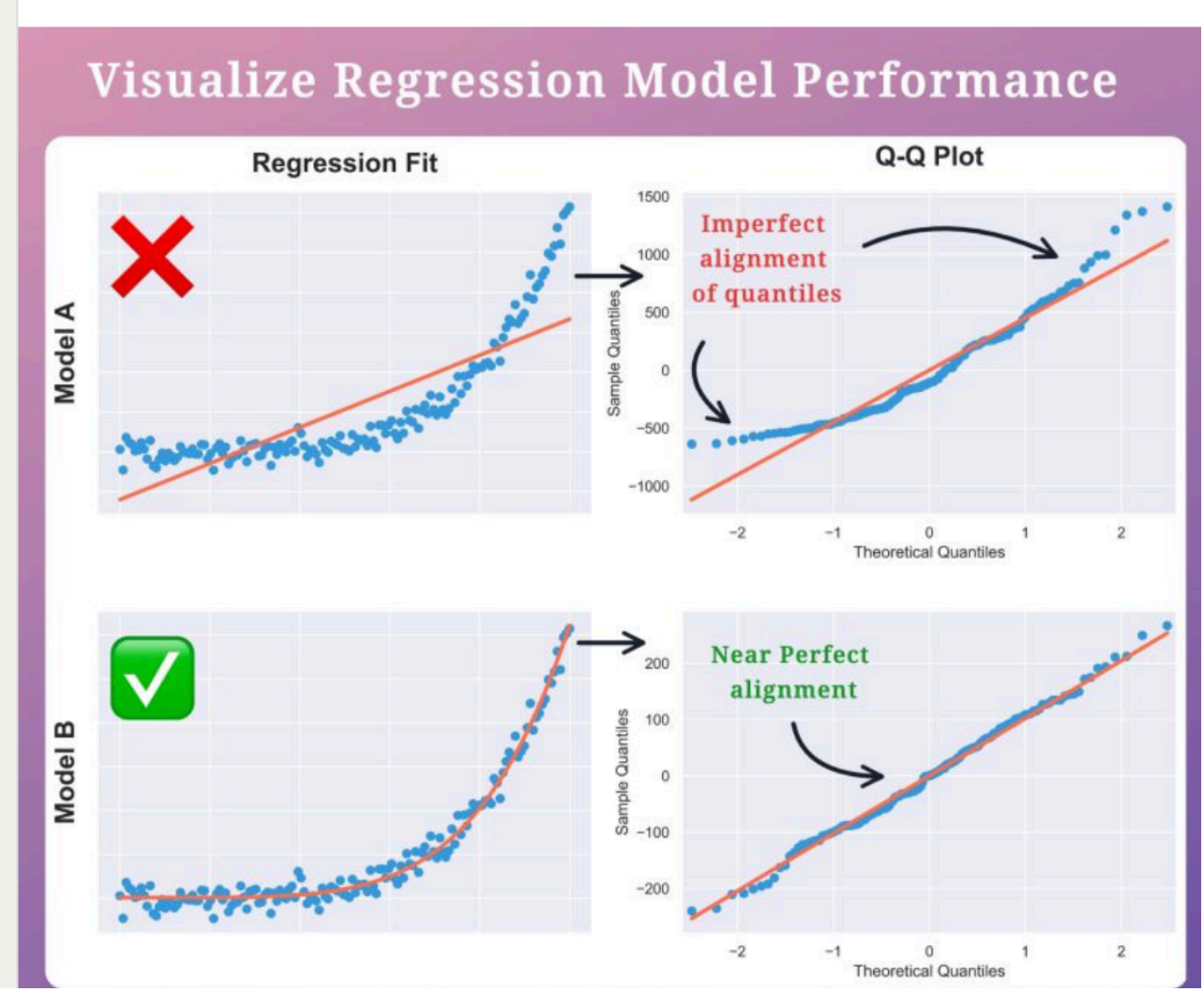
This is especially useful when the regression line is difficult to visualize, i.e., in a

high-dimensional dataset.

So remember...

After running a linear model, always check the distribution of the residuals.

This will help you:
- Validate the model's assumptions
- Determine how good your model is
- Find ways to improve it (if needed)



RL:
https://jonathan-hui.medium.com

https://www.rebellionresearch.com/accentures-chief-data-scientist

https://huyenchip.com/2023/05/02/rlhf.html

https://datamachines.xyz/the-hands-on-reinforcement-learning-course-page/. == best

https://datamachines.xyz/the-hands-on-reinforcement-learning-course-page/
https://datamachines.xyz/the-hands-on-reinforcement-learning-course-page/?trk=feed_main-feed-card_feed-article-content

https://datamachines.xyz/the-hands-on-reinforcement-learning-course-page/

https://www.linkedin.com/

KG:
https://medium.com/@peter.lawrence_47665/knowledge-graphs-large-language-models-the-ability-for-users-to-ask-their-own-questions-e4afc348fa72

https://medium.com/neo4j/creating-a-knowledge-graph-from-video-transcripts-with-gpt-4-52d7c7b9f32c

https://medium.com/@amine.dadoun/knowledge-graph-embeddings-101-2cc1ca5db44f

https://towardsdatascience.com/graph-embeddings-explained-f0d8d1c49ec
https://medium.com/@preeti.chauhan8/learn-a-z-of-knowledge-graphs-part-6-shallow-graph-embeddings-node2vec-3a3f33bf72a7
https://medium.com/@amine.dadoun/knowledge-graph-embeddings-101-2cc1ca5db44f

https://thedataquarry.com/posts/neo4j-python-1/

https://towardsdatascience.com/langchain-has-added-cypher-search-cb9d821120d5

https://www.linkedin.com/posts/sarath-chandra-nalluri_langchain-llms-promptengineering-activity-7067710956788711424--ojt

https://medium.com/neo4j/knowledge-graphs-llms-fine-tuning-vs-retrieval-augmented-generation-30e875d63a35

https://medium.com/neo4j/knowledge-graphs-llms-fine-tuning-vs-retrieval-augmented-generation-30e875d63a35

augmented-generation-30e875d63a35

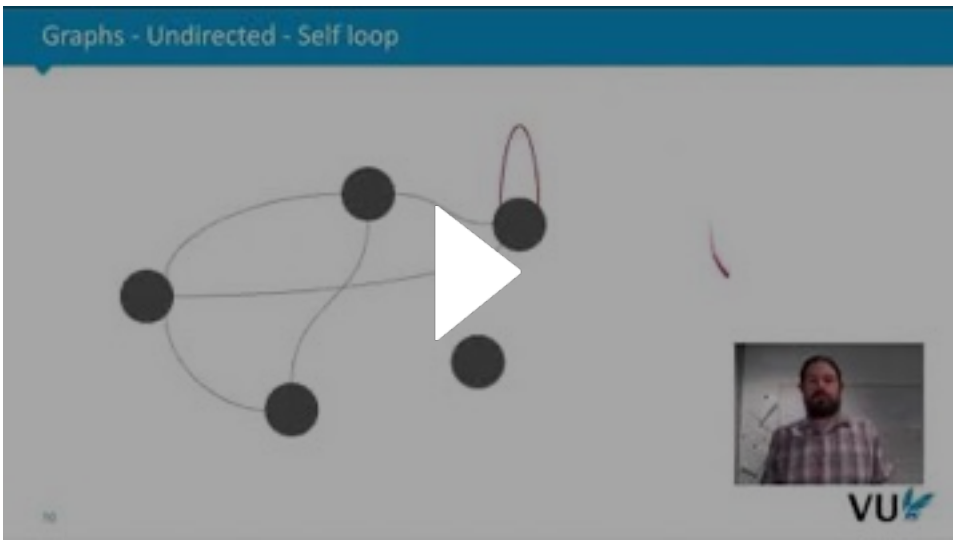
https://medium.com/neo4j/monitoring-the-cryptocurrency-space-with-nlp-and-knowledge-graphs-92a1cfaebd1a
https://github.com/tomasonjo/blogs/blob/master/nft/DiffbotNLP%20%2B%20Neo4j.ipynb

https://medium.com/neo4j/creating-a-knowledge-graph-from-video-transcripts-with-gpt-4-52d7c7b9f32c
https://towardsdatascience.com/extract-knowledge-from-text-end-to-end-information-extraction-pipeline-with-spacy-and-neo4j-502b2b1e0754
https://towardsdatascience.com/from-text-to-knowledge-the-information-extraction-pipeline-b65e7e30273e#:~:text=What%20exactly%20is%20an%20information,unstructured%20data%20such%20as%20text.

Langchain and cypher search:
https://towardsdatascience.com/langchain-has-added-cypher-search-cb9d821120d5

KG:
https://towardsdatascience.com/how-to-use-chat-gpt-and-python-to-build-a-knowledge-graph-in-neo4j-based-on-your-own-articles-c622bc4e2eaa

Lecture 8



HMM:
https://newsletter.theaiedge.io/p/modeling-customers-loan-default-with

For building a knowledge graph for Telekom Germany, you will need to use both RDF and OWL. RDF will be used to represent the data about the resources (such as products, services, and customers) and their properties in a machine-readable format. OWL will be used to create a formal ontology that defines the concepts and relationships within the domain of Telekom Germany.

Using RDF, you can represent the data in the form of subject-predicate-object triples, where the subject represents the resource, the predicate represents the property, and the object represents the value of the property. For example, you can create triples such as:

- Telekom Germany has a product offering named "MAGENTA ZUHAUSE S MAGENTA TV NETFLIX".
- "MAGENTA ZUHAUSE S MAGENTA TV NETFLIX" has a feature named "Netflix streaming".
- "MAGENTA ZUHAUSE S MAGENTA TV NETFLIX" is available in the region of Berlin.

Using OWL, you can define the concepts and relationships within the domain of Telekom Germany. For example, you can define a class named "Product Offering" and specify its subclasses such as "Fixed-line contracts" and "Mobile contracts". You can also define properties such as "hasFeature" and "isAvailableIn" to specify the relationships between different entities.