# Deployments of ML models

1. Overview

* Process of making ML models available in prod environments
* One of the last stages in ML lifecycle – prolly the most challenging stage

1. Challenges of Model deployment

* Challenges of traditional software
* Reliability
* Reusability
* Maintainability
* Flexibility
* Additional challenges specific to ML
* Reproducibility – same result with same data across systems
* Why challenging?
* Needs coordination of DS, IT teams, software devs and business professionals
  + Ensure model works reliably
  + Ensure model delivers the intended result
* Potential discrepancy between programming language used to develop the model and the prod system language

1. Why important?

* Maximize the value of the ML model

# Deployment of ML pipelines

1. ML models

* Data from DB, Cloud, 3rd party APIs -> Model -> prediction

1. Data format and quality

* Data is almost never ready to be used to train the model
* Missing values
* Contains strings
* Distribution
* Outliers
* Needs to transform variables, extract features, create features
* Preprocess the data

1. ML pipeline

* Data preprocessing – feature engineering
* Variable selection – feature selection
* ML model building
* Obtain results

1. Deployment of ML pipeline

* Need to deploy the entire pipeline of feature engineering, model training, and scoring

# Research and production environments

1. Environments

* The setting or state of a computer where software or other products are developed or put in operation

1. Research environments

* A setting with tools, programs, and software suitable for data analysis and the development of ML models
* Here, we develop the ML models and identify their value

1. Production environment

* A real-time setting with running programs and hardware setups that allow the org’s daily operations
* Where the ML models are actually available for business use
* Allows orgs to show clients a ‘live’ service

# Building reproducible ML pipelines

1. Deployment of ML models & pipelines
2. Deploying the value
3. Why reproducibility matters?

* Without the ability to replicate prior results, it is difficult to determine if a new model is truly better than the previous one

1. Challenges to reproducibility

* Reproducibility is the ability to duplicate a ML model exactly – given the same raw data as input, both models return the same result
* ML pipeline overview

Data gathering -> Data analysis -> Data pre-processing -> Variable selection -> ML model building -> Model deployment

1. Reproducibility during Data Gathering

* Challenges
* Training dataset can’t be reproduced
* Databases are constantly updated and overwritten
* Order of data while loading is random (SQL)
* Solutions
* Save a snapshot of training data
  + Simple
  + Potential conflict with GDPR
  + Not suitable for big data
* Design data sources with accurate timestamps
  + Ideal situation
  + Big effort to redesign the data sources

1. Reproducibility during feature creation

* Lack of reproducibility may arise from
* Replacing missing data with random extracted values
* Removing labels based on percentages of observations
* Calculating stat values like mean to replace missing value
* More complex equations to extract features
* Solution?
* Code on how a feature is generated should be tracked under version control and published with auto-incremented or timestamp hashed versions
* Many of params extracted for feature engineering depend on the data used for training -> ensure data is reproducible
* If replacing by extracting random samples, always **set a seed**

1. Reproducibility during Model training

* Challenges
* ML models rely on randomness for training
  + Data and feature extraction for trees
  + Weight initialization for neural nets, etc.
* ML model implementations work with arrays agnostic to feature names
  + Need to be careful to feed data in the correct order
* Solutions
* Record the order of the features
* Record applied feature transformations
* Record hyperparams
* Set a seed for models requiring randomness
* If final model is a stack of models, record the structure of the ensemble

1. Reproducibility during Model deployment

* Challenges
* Feature not available in live env
* Different programming languages
* Different software
* Live populations don’t match those used for training
* Solutions
* Software versions should match exactly – list all 3rd party library dependencies and their versions
* Use a container and track software specifications
* Research, develop, and deploy utilizing the same language
* Prior to building the model, understand how the model will be integrated with other systems

# Streamlining Model deployment with open-source

1. Challenges

* Lots to code
* Time consuming
* Different versions across team
* Repetitive
* Multiple copies of same code
* Different versions of same code
* Difficult to keep track
* Learn and store parameters
* Multiple intermediate files with params
* Config or param file

1. Reproducibility in deployment

* Re-write code
* Include tests
* Reproducibility

1. Team performance
2. Open source

* No more coding
* Version tracking for reproducibility
* Classes and functions include tests – no need to recode for production