Chapter 13 - Deploying a Machine Learning API

- Prediction = Machine Learning → Train various models then use them to make prediction = inference → Model processes and entire group of prediction at once = batch inference → May be done by a scheduled script or job → Problem: prediction change minute-by minute → Solution: Real-Time Inference = Calling a model to get a single prediction immediately $\xrightarrow{\text{tool}}$ deploying model as an API
- Concepts that you will come across:
  - Classification:
    - Type of a model that predicts what category a value will fall into.
    - Classifiers = Model that perform classification
  - Decision Trees:
    - Type of a ML algorithm that creates a recursive tree structure to perform classification or regression.
  - Evaluating a model:
    - Comparing the models' predictions to test data to see how well it would have predicted past events.
  - Gradient Boosting:
    - ML technique that combines multiple models to create a model that is more effective than the individual models.
  - Regression:
    - Type of model that predicts a continuous numeric value
    - Regressor = Models that perform regression
  - Training a Model:
    - Using the training portion of historical data to create a model that can make inferences based on new data.

Training Machine Learning Models

- Supervised Learning: a method of creating a models by processing existing data where the expected values are known (labeled/categorized data).
- Inference Process: When model has been trained, it can be used to read inputs and predict output.

Tools

- ONNX Runtime: A cross-platform tool for using models from a variety of different frameworks
- Scikit-learn: An ML framework for training models
- Sklearn-onnx: A library that converts scikit-learn models to ONNX format

ONNX Runtime

- The Open Neural Network Exchange (ONNX) is an open standard for ML models
- ONNX is a standard format that models from different programming languages and different frameworks can be converted to and run in a standard way.
    - Allow greater interoperability
        - You can easily deploy model with ONNX Runtime
            - ONNX Runtime include acceleration that can improve model inference performance

Scikit-learn

- Python framework that allow ML implementations
- Competitors: PyTorch, TensorFlow, XGBoost

Sklearn-onnx

- Because we are using scikit-learn to create model, we will use sklearn-onnx library to convert the model into ONNX format

Using the CRISP-DM Process

- A useful method of organizing an ML modeling project is the Cross-Industry Standard Process for Data Mining (Shearer, 2000)
  Stages of CRISP-DM:
- Business Understanding
    - Team identifies business objectives and assesses tools and techniques available.
- Data Understanding
    - Collecting data that is available to solve the problem, explore it, verify the data quality.
- Data Preparation
    - Data Scientists select specific data elements to be used, format them, merge with any additional sources needed

- Modeling
    - Select a Modeling Technique and building a model that answers your business question
- Evaluation
    - Review the model for its ability to solve the question and its readiness for production
- Deployment
    - Models are deployed in an environment where they can be consumed by the customer, Monitor and maintain model.

Business Understanding

- Understanding the Problem that you are trying to solve.
- Question: How much will it cost to acquire this player on waivers?
    - Fantasy manager add players to their rosters through a waiver request.
        - Blind bidding auction is performed to decide who gets the best available players
            - Manager decide which player to bid for, and how much they are willing to spend, which is hidden from others
                - Each manager has a set amount of money we call Free Agent Acquisition budget (FAAB)
                    - We want to bid high enough, not overspent
                    - We give different ranges of prediction
                        - Low-end cost (10th percentile)
                        - Median cost (50 percentile)
                        - High-end cost (90th percentile)

Data Understanding

`player_training_data_full.csv` contain columns:

- Fantasy regular season weeks remaining
    - How many weeks are left in the regular season.
- League budget percentage remaining
    - The percent of total dollars available in the league
- Player season number
    - The number of seasons this player has been in the league
- Position:

    - The fantasy football position of the players that was acquired
- Waiver value tier
    - A qualitative measure of how valuable an individual player is.

Data Preparation

- Instead of trying all possible variables you need to select specific features for your model to learn on, and this selection must have a reason or theory that support it
    - League budget percentage remaining
        - higher budget remaining leads to higher bids → linear feature: the output variable goes up or down at a consistent rate as this value change
    - Fantasy regular season weeks remaining
        - Players cost more at different points of the season, is not strictly linear
    - Waiver value tier
        - Fact: higher value players will cost more $\xrightarrow{\text{uestions}}$ How much more? How each tier affected? → We want to model be able to detect this

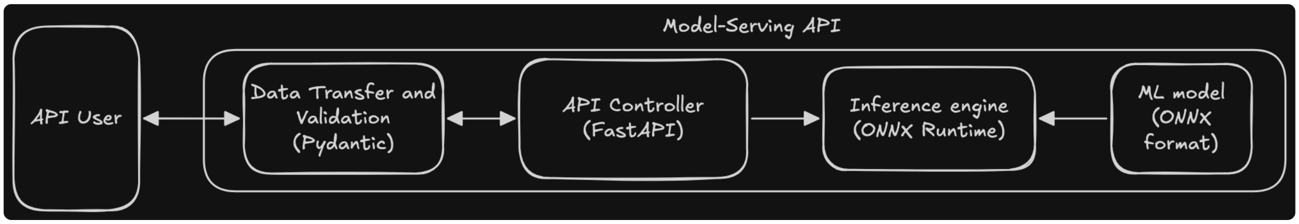
Modeling

- Select Algorithm + ML Framework → Combination of Technical Limitation & Modeling factors
    - Technical Limitation: Python Framework to Implement ML + Can be converted to ONNX + prediction for 10th, 50th and 90th percentile
    - Modeling Factors: Output is numerical (regressor) + features are not linear (budget remaining (linear) value tier (categorical) weeks remaining (slightly complicated)) → Decision Tree Regressor
- Gradient Boosting Regressor: A way to combine multiple decision trees into an ensemble model that is more predictive than using individual decision trees by themselves + support multiple prediction
- We do 80-20 split for train and test
- Fitting: Process of training your model, where library takes a general algorithms and fits/apply it to your training data to make a specialized model.

Evaluation

- Iterative Process that we evaluate models with formal metrics such as accuracy, fairness, and etc.
- Checkout Designing Machine Learning Systems by Chip Huyen

Deployment



Documenting Machine Learning Models

- Model Cards proposed by Google
  - model's operations, risks and biases
- System Cards proposed by Meta
  - Holistically across an AI system, versus one-off models

Additional Resources:

- Practical Data Science with Python | Data | eBook
- Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow, 2nd Edition [Book]
- Designing Machine Learning Systems [Book]
- ONNX Runtime | Getting-started
- Tutorial - sklearn-onnx 1.19.1 documentation