 [**Planning and project setup**](https://www.jeremyjordan.me/ml-projects-guide/#planning)

* Goal of predicting the probability of a fastball, slider, etc., in a real-time environment.
* Classification problem. Output probability of each pitch
* Real time environment – needs to execute quickly
* build and evaluate a model that would be acceptable in a production environment after improvement
* actually delivering predictions with any degree of accuracy is unlikely in this short time span
  + Goal – Not perfection, but better than random guessing
* additionally provide any associated data analysis (plots, graphs etc..), feature engineering and code assembled during the 3-5 hours in the form of a pdf or .html file.
* Will attempt to implement XGBoost
  + Needed to install via pip
* Creating working directory with pitch data, dataset metadata and a jupyter notebook for easy analysis and iterative execution

 [**Data collection and labeling**](https://www.jeremyjordan.me/ml-projects-guide/#data)

* Define ground truth (create labeling documentation)
* Build data ingestion pipeline
* Validate quality of data
* Revisit Step 1 and ensure data is sufficient for the task

 [**Model exploration**](https://www.jeremyjordan.me/ml-projects-guide/#exploration)

* Start with a simple model using initial data pipeline
* Overfit simple model to training data
* Determine and implement simple baseline
* Stay nimble and try many parallel (isolated) ideas during early stages
* Find SoTA model for your problem domain (if available) and reproduce results, then apply to your dataset as a second baseline
* Revisit Step 1 and ensure feasibility
* Revisit Step 2 and ensure data quality is sufficient

 [**Model refinement**](https://www.jeremyjordan.me/ml-projects-guide/#refinement)

* Perform model-specific optimizations (ie. hyperparameter tuning)
* Iteratively debug model as complexity is added
* Perform error analysis to uncover common failure modes
* Revisit Step 2 for targeted data collection of observed failures

 [**Testing and evaluation**](https://www.jeremyjordan.me/ml-projects-guide/#testing)

* Evaluate model on test distribution
  + May necessitate additional training
  + Understand differences between train and test set distributions (how is "data in the wild" different than what you trained on)
* Revisit model evaluation metric
  + Ensure that this metric drives desirable downstream user behavior
* Write tests for:
  + Input data pipeline
  + Model inference functionality
  + Model inference performance on validation data
  + Explicit scenarios expected in production (model is evaluated on a curated set of observations)

 [**Model deployment**](https://www.jeremyjordan.me/ml-projects-guide/#deployment)

* Expose model via a REST API
* Deploy new model to small subset of users to ensure everything goes smoothly, then roll out to all users
* Maintain the ability to roll back model to previous versions
* Monitor live data and model prediction distributions

 [**Ongoing model maintenance**](https://www.jeremyjordan.me/ml-projects-guide/#maintenance)

* Understand that changes can affect the system in unexpected ways
* Periodically retrain model to prevent model staleness
* If there is a transfer in model ownership, educate the new team