Week 1

- Plan
    - Motivate Machine Learning
    - Introduce notation used throughout course
    - Plan for initial lectures
        - *What*: Introduce, motivate a model
        - *How*: How to use a model: function signature, code (API)
        - *Why*: Mathematical basis -- enhance understanding and ability to improve results

- Course Overview (Course_overview.ipynb)
- Getting Started (Getting_Started.ipynb)
- Machine Learning: Overview (ML_Overview.ipynb)
- Intro to Classical ML (Intro_Classical_ML.ipynb)

Week 2

- Plan
    - Introduce a model for the Regression task: Linear Regression
    - Introduce the Recipe for Machine Learning: detailed steps to problem solving

- Recap: Intro to Classical ML (Recap_of_Intro_Classical_ML.ipynb)

- Our first model: Linear Regression (Overview) (Linear_Regression_Overview.ipynb)

- A *process* for Machine Learning
    - Go through the methodical, multi-step process
        - Quick first pass, followed by Deeper Dives
    - This will be a code-heavy notebook !
    - Illustrate Pandas, Jupyter, etc
    - Recipe for Machine Learning: Overview (Recipe_Overview.ipynb)
        - Linked notebook (Recipe_for_ML.ipynb)

- The Loss function for Linear Regression

    - Linear Regression: Loss Function (Linear_Regression_Loss_Function.ipynb)

- Deeper dives

    - Iterative improvement
        - When to stop: Bias and Variance (Bias_and_Variance.ipynb)
            - Regularization
    - Fine tuning techniques (Fine_tuning.ipynb)

Week 3

- Deferred from Week 2

        - Prepare Data step: Introduction to Transformations
            - Transforming data (featuring engineering) is a key step in the Recipe
            - We introduce transformations
                - Focus on the *how*; subsequent lecture will cover the *why*
            - Prepare Data: Intro to Transformations (Prepare_data_Overview.ipynb)
            - Transformation pipelines in `sklearn` (Transformations_Pipelines.ipynb)

- Plan
    - Introduce a model for the Classification task: Logistic Regression
    - How to deal with Categorical (non-numeric) variables
        - classification target
        - features
    - **Classification intro**
        - Classification: Overview (Classification_Overview.ipynb)
        - Classification and Categorical Variables (Classification_Notebook_Overview.ipynb)
            - linked notebook (Classification_and_Non_Numerical_Data.ipynb)

        - **Non-numeric variables (categorical)**
            - Categorical variables (Categorical_Variables.ipynb)
            - Titanic using categorical features (Classification_and_Non_Numerical_Data.ipynb#Titanic-revisited:-OHE--features)
        - **Classification, continued**
            - Multinomial Classification (Multinomial_Classification.ipynb)
            - Classification Loss Function (Classification_Loss_Function.ipynb)
- Deeper dives
    - Baseline model for Classification (Classification_Baseline_Model.ipynb)
    - Entropy, Cross Entropy, and KL Divergence (Entropy_Cross_Entropy_KL_Divergence.ipynb)
    - Log odds (Classification_Log_Odds.ipynb)
    - Dummy variable trap (Dummy_Variable_Trap.ipynb)
- Deferred modules
    - Deferred from week 2
        - Fine tuning techniques (Fine_tuning.ipynb)

Week 4

- Deferred from Week 3
    - Deeper dives
        - Baseline model for Classification (Classification_Baseline_Model.ipynb)
        - Dummy variable trap (Dummy_Variable_Trap.ipynb)
- Plan
    - Error Analysis
        - We explain Error Analysis for the Classification Task, with a detailed example
        - How Training Loss can be improved
    - Transformations
        - One of the most important parts of the Recipe: transforming raw data into something that tells a story
    - Loss functions
        - We look at the mathematical logic behind loss functions
- Recap and Clarification: Classification (Recap_of_Classification.ipynb)
- Error Analysis (Error_Analysis_Overview.ipynb)
    - Error Analysis for Classification (Error_Analysis.ipynb)
        - Summary statistics
        - Conditional statistics
        - Worked example (Deeper Dive)

- Loss Analysis: Using training loss to improve models (Training_Loss.ipynb)
- Transformations Overview (Transformations_Overview.ipynb)
    - Transformations: the how (Transformations_Overview.ipynb)
    - Transformations: the why (Transformations.ipynb)
- Deeper dives
    - Loss functions: the math (Loss_functions.ipynb)
        - Maximum likelihood
        - Entropy, Cross Entropy, and KL Divergence (Entropy_Cross_Entropy_KL_Divergence.ipynb)
        - Preview: custom loss functions and Deep Learning
    - Examining errors in MNIST classification (Error_Analysis_MNIST.ipynb)
- Deferred modules
    - Deferred from week 2
        - Fine tuning techniques (Fine_tuning.ipynb)

Week 5

- Plan

        - More models: Decision Trees, Naive Bayes
            - Different flavor: more procedural, less mathematical
            - Decision Trees: a model with *non-linear* boundaries
        - Ensembles
            - Bagging and Boosting
            - Random Forests
        - Naive Bayes: a simple but effective model

- Entropy, Cross Entropy, and KL Divergence (Entropy_Cross_Entropy_KL_Divergence.ipynb)

- Decision Trees: Overview (Decision_Trees_Overview.ipynb)

- Decision Trees (Decision_Trees_Notebook_Overview.ipynb)
    - linked notebook (Decision_Trees.ipynb)
- Trees, Forests, Ensembles (Ensembles.ipynb)

- Naive Bayes (Naive_Bayes.ipynb)

- Deeper Dives

    - Feature importance (Feature_Importance.ipynb)

- Deferred modules

    - Deferred from week 4
        - Examining errors in MNIST classification (Error_Analysis_MNIST.ipynb)
    - Deferred from week 3
        - Log odds (Classification_Log_Odds.ipynb)

        - Log odds (Classification_Log_Odds.ipynb)
    - Deferred from Week 2
        - Fine tuning techniques (Fine_tuning.ipynb)

Week 6

- Plan
    - Support Vector Classifiers: a classifier with an interesting twist
    - Interpretation: understanding models
    - Gradient Descent: our tools for solving optimization problems
- Support Vector Machines: Overview (SVM_Overview.ipynb)
- SVC Loss function (SVM_Hinge_Loss.ipynb)
- SVC: Large Margin Classification (SVM_Large_Margin.ipynb)
- SVM: Kernel Transformations (SVM_Kernel_Functions.ipynb)
- SVM Wrapup (SVM_Coda.ipynb)
- Gradient Descent (Gradient_Descent.ipynb)
- Interpretation: Linear Models (Linear_Model_Interpretation.ipynb)

Deeper Dives

- SVC Loss function derivation (SVM_Derivation.ipynb)
- Missing Data (Missing_Data.ipynb)
- Imbalanced data (Imbalanced_Data.ipynb)

- Deferred modules
    - Deferred from week 4
        - Examining errors in MNIST classification (Error_Analysis_MNIST.ipynb)
    - Deferred from week 3
        - Log odds (Classification_Log_Odds.ipynb)
    - Deferred from Week 2
        - Fine tuning techniques (Fine_tuning.ipynb)

Week 7

- Plan
    - Unsupervised Learning
- Unsupervised Learning: Overview (Unsupervised_Overview.ipynb)
- PCA Notebook Overview (Unsupervised_Notebook_Overview.ipynb)
    - linked notebook (Unsupervised.ipynb)

Deeper dives

- Other matrix factorization method (Unsupervised_Other_Factorizations.ipynb)

Week 8

- Set up your Tensorflow environment (Tensorflow_setup.ipynb)
- Neural Networks Overview (Neural_Networks_Overview.ipynb)
- Coding Neural Networks: Tensorflow, Keras
    - Intro to Keras (Tensorflow_Keras.ipynb)
    - Using Tensorflow
        - Two choices
            - Local: DNN Tensorflow example Notebook local (DNN_TensorFlow_example.ipynb) (local)
                - Tensorflow version 2+ only !
            - DNN Tensorflow example Notebook from github (https://colab.research.google.com/github/kenperry-public/ML_Spring_2020/blob/master/DNN_TensorFlow (**Colab**)
- Practical Keras (Tensorflow_Keras_Practical.ipynb)
- Practical Colab

- Colab: Practical Colab Notebook from github (https://colab.research.google.com/github/kenperry-public/ML_Fall_2019/blob/master/Colab_practical.ipynb)

Deeper Dives

- A neural network is a Universal Function Approximator (Universal_Function_Approximator.ipynb)
- History/Computation Graphs: Tensorflow version 1 (DNN_TensorFlow_Using_TF_version_1.ipynb)
- Raw_TensorFlow example Notebook from github (https://colab.research.google.com/github/kenperry-public/ML_Spring_2020/blob/master/Raw_TensorFlow.ipynb) (**Colab**)
- Computation Graphs (Computation_Graphs.ipynb)

Assignments

Assignment Guidelines (assignments/Assignment_Guidelines.ipynb)

Assignment 1

- Assignment 1 notebook (assignments/Assignment_1.ipynb)
- Assignment 1 data (assignments/data/assignment_1)

Assignment 2

- Assignment 2 notebook (assignments/Assignment_2.ipynb)
- Assignment 2 data (assignments/data/assignment_2)