Data Visualization

* Generate a boxplot for the data, per class
* Include count of step per class
  + Discuss maintaining distribution of classes for train and test set
* Include range of each coordinate per class
  + Discuss differing ranges for each feature, and the need for scaling
  + Discuss decision to use min/max scaling instead of standard

Correlation Analysis

* Include correlation heatmap
* Discuss low correlation between features
  + All features are highly independent, at least linearly
* Discuss high correlation with x-coordinate and step
* Discuss low correlation of z-axis and class
  + Could potentially be removed without significant impact on model
  + Makes sense physically, as humans down really translate about this axis, at least macroscopically

Data preparation

* Discuss min/max scaling decision

Model Development

* Discuss choice of classifiers
  + Some that can handle native multiclassification natively
  + Include sklearn decision flowchart
* General for each model
  + Discuss choice of parameters in parameter grid
  + Discuss using multi-scoring instead of just accuracy
  + Discuss process of tuning with GridSearch
    - Tweaking values until best parameters fall within range, ensuring model isn’t limited by range of chosen hyperparameters
* Random forest
  + Able to handle multiclass tasks natively
* Complement Naïve Bayes
  + Good for imbalanced datasets
  + Default values are even worse in terms of accuracy
  + Fit is almost instantaneous compared to other models
* SVM
  + Recommended by sklearn
* Gaussian naïve bayes
  + CNB had really poor performance
  + Want to compare to the base classifier

Model Analysis

* Generate scoring reports for each model
  + Test and train sets
    - Discuss if overfitting seems to be an issue
* Discuss scoring metrics
  + Meaning of each in context of problem
  + Which ones to favour
  + Why not recall
  + Weighted average
* Select best model
  + Generate confusion matrix for it
  + Discuss confusion matrix results
  + Discuss errors with the model
  + Evaluate on test set

Stacked Model

* Combine two models and evaluate performance
* Combined best performing models
* Slight higher accuracy from stacked model

Model Evaluation

Joblib