Coders Inc Design Decisions – Milestone 4

# Use of Abstract Classes in GUI

* Many classes used in the View and Controller were very similar, so to reduce code duplication and coupling, abstract classes were used. The following classes were refactored, with their abstract parent class in bold:
  + **FeaturePanelController:** FeaturePanelComplexController, FeaturePanelSimpleController
  + **ValueInputController:** ValuePromptFrameController, ValueTestFrameController
  + **FeaturePanel:** FeaturePanelComplex, FeaturePanelSimple
  + **PromptFrame:** PromptValueFrame,TestCaseFrame

# Use of Packages

* In order to keep track of many classes, they were sorted into packages based on their functionality and responsibility:
  + Maths
  + View
  + Controllers
  + DataModel
* A fifth package holds archive documents and files relevant to the project (such as this one)

# Distance Metrics for kNN Functions

* 4 different distance metrics have been implemented
  + Euclidean
  + Minkowski
  + Manhattan
  + Chebyshev
* These 4 metrics cover a good range of possibilities
  + Euclidean is a good standard and familiar metric
  + Manhattan gives a view of the absolute distance
  + Minkowski provides a more flexible version of Euclidean distance in terms of polynomial order
  + Chebyshev gives a view of the maximum difference of any dimensions which is useful in terms of how the points cluster

# Null Target Value Handling

* In the case where a potential neighbour does not have a value for the target value, that point is ignored by the kNN function
  + This is due to the fact that this point will not provide any useful information for when the target value is being calculated