# Machine Learning Final Exam


**Department of Computer Science, University of Copenhagen**

Dhruv Chauhan


January 18, 2017

## 1 In a galaxy far, far away

### 1.1

The variance of the red-shifts in the spectroscopic training data was calculated to be:

$$0.0106$$

(where from now on, unless specified, values are shown to 3 significant places).

The MSE on the test SDSS predictions was calculated to be:

$$0.000812$$

### 1.2

The linear regression was done in Python, using the `sklearn` linear regression package. This performs an ordinary least squares linear regression. The error function is a Mean Squared Error.

The parameters of the model were calculated to be:

[ -2.82898070e+11, 6.79638352e+11, -7.30280682e+11, 3.84379194e+11, -5.08387940e+10, -2.44829466e+11, 6.20014394e+10, 4.45567121e+11, -3.89498568e+11, 1.26759474e+11, 2.82898070e+11, -3.96740282e+11, 3.33540400e+11, -5.08387940e+10, 2.44829466e+11, 1.82828027e+11, -2.62739094e+11, 1.26759474e+11]

The error on the training data was calculated to be 0.00187, and on the test data was 0.00187 also. The errors noramlised by the variance, $\sigma^2_{red}$ were equal to 0.176 for both the test and the training data.

This normalised error falling below one signifies that...

## 1.3

For the non-linear regression, I chose to apply the K-nearest neighbours (KNN) algorithm. I chose this method for its simplicity (following Occam's razor), and therefore its ease of understanding. The simplicity of the algorithm is also reflected in the single hyperparameter, $k$, which means that there is less computation in tuning the hyperparameter.

I utilised the `neighbours` library from the `sklearn` package.

The KNN algorithm uses a distance metric to calculate the distance between a (set of) training point(s). I used the Euclidian distance, given by $||\mathbf{x} - \mathbf{x}'||$, or $\sqrt{\mathbf{x}^T\mathbf{x}'}$.

My method involved doing the following:

1. Given a certain K,