Question 4

1. Algorithm submitted.
Proof for time complexity of $O(nd)$

~~Calculate~~ Calculating the l2 distance for each training point per testing point : $O(n*d)$
Use quickselect to find kth smallest distance : $O(n)$
Finding all nearest neighbours to store their y-values : $O(n)$

$\therefore$ Total time complexity $\Rightarrow O(n*d + n + n)$
Since nd is more ~~larger~~ significant compared to n, being dominant, total time complexity = $\underline{O(n*d)}$ irrespective of k

2. When does each approach perform better, and why?

The least squares linear regression is better when the data to be predict continuous numeric data where there is a linear relationship between independent and dependent variables. Eg. Data in dataset D

The k-Nearest Neighbour can be used for classification as well as regression. For regression, it doesnt assume linearity and as such is very suitable for complex non linear dataset. However, computational cost becomes excessive for large datasets Eg. Dataset E.

3. Which approach is better and why?

The k-nearest neighbours perform better as they have ~~more~~ a ~~higher~~ higher k values. As the k value increase there are more near neighbours used to predict the y values rather than only using the neighbour with the lowest l_2 distance.
The linear regression prediction is better than k=1 or 2 but ~~remains~~ the MSE remains constant as the k value go down.

Inspecting the nearest distances between the training and test set data, a large number of nearest neighbours are quite close to the data point ranging from 2.8 to 3.2 because of which this approach is better than linear regression for this set of data points with d=20.