What are the best *k* hashes and *m* bits values to store one million *n* keys (E.g. e52f43cd2c23bb2e6296153748382764) suppose we use the same MD5 hash key from pickle_hash.py and explain why?

In order to store one million n keys into an m bit array the number of size of the bitarray that is ‘m’ should be **6235224** and the number of hash functions to be used are **4**.
This calculation is based by assuming the probability score to be 0.05.
The size of the bit array is determined by using the formula:

$$m = - (n*log(p))/log(2)^2$$

The ‘m’ size obtained by this will be good enough to accommodate 1 million keys without giving more false positives. The number of hash functions is calculated by using:

$$k= m/n*log(2)$$

The number of hash functions to be used are picked as 4 to decrease the latency, as the number of hash functions used increases it adds the overhead time of calculating hash and assigning them to more bits in the bitarray resulting in more false positives.