Google Prediction API evaluation of predictions

From the 3 different approaches I used, these are the accuracies I got:

Approach 1(990k training and 10k testing): 64.52
Approach 2(same as approach 1 but with categories feature): 63.63
Approach 3(10 fold cross validation): 43.762

The cross validation method performs significantly lower than the other two approaches, because of the smaller training size (90k as opposed to 990k). Because of the low accuracy, I would discard it and focus on the first two approaches.

To dive deeper into analyzing the accuracy of the predictions, it would help to look at the confusion matrix.

Approach 1 :

[0,	41,	67,	41,	133]
[334,	0,	129,	98,	139]
[190,	57,	0,	262,	307]
[72,	22,	145,	0,	1082]
[83,	20,	48,	278,	0]

Approach 2:

[0,	60,	69,	45,	165]
[276,	0,	157,	83,	161]
[162,	79,	0,	214,	352]
[59,	43,	162,	0,	1133]
[66,	34,	76,	241,	0]

Number of data points for each rating in each training data:

1.0: 102239
2.0: 74934
3.0: 104703
4.0: 180661
5.0: 527463

So, in essence, more than half of the training data is comprised of data with a rating of 5.0 . This might reason out for the fact that most ratings other than 5 are wrongly labelled as 5 as the training data is heavily composed of 5's. Approach 2 improvises on approach 1 in the fact that less number of lowly rated products with either a 1 or 2 rating are rated a high rating of 4 or 5 and conversely a high rating of 4 or 5 has less predictions of 1 or 2.
However, I was expecting a significant increase in accuracy in approach 2 because of the suggestion given in the prediction API docs, the more number of features *generally* gives a better predictions. The inclusion of the categories feature might skew the predictions because the density of categories is really sparse as there are different combinations of hierarchies which make up a category. Here I would have liked if there was more control given to the user to change the weightage of parameters or tune them.

Overall, I think the prediction API does a decent job at predicting the ratings. Although I would have liked a better precision, maybe around the 72-75% mark, given the low number of features provided and the skewness of the training data, I think the accuracy is fair.