Model comparison

Data description

The data is composed of 44419 images with size varying from 2400X1800 and 1600X1200. As the models work with images with varying sizes, we chose a standard size for resizing that can be used by all the model.The standard size we used is 600X450 as it is of the same ratio as the original except for NasnetLarge as it only takes image of size 331X331. For time issues we're not using all the data. We're using images with articleTypes that have 500 or more images for all the models except NasnetLarge.For NasnetLarge, we're using images with articleTypes that have 650 or more for memory issues in Kaggle. With that filtering we'll use 34719 images for all other models and for NasnetLarge we'll use 31284 images.

Accuracy metric

We can use 3 accuracy metric:

- Accuracy
- Precision
- Recall

As this is a recommendation system, so we can't have wrong predictions even if some good recommendations are omitted. So we need to be precise with the recommendation. So we use precision.

Features used for calculating

We are mainly focusing on the article type and gender as it would be embarrassing if we recommended a pair of heels when a man's shoe is given as a reference item or if we recommended a shirt when a watch is given. So we are emphasizing on articleType and secondly gender.

Comparison

Model	Top N	Features	Tolerance	Precision
Xception	10	articleType	.2	80.96%
		articleType	.1	75.56%
		articleType	0	66.38%
		gender	.2	75.98%
		gender	.1	68.27%
		gender	0	56.40%
		gender and articleType	.2	63.16%
		gender and articleType	.1	54.46%
		gender and articleType	0	42.41%
InceptionV3	10	articleType	.2	78.15%
		articleType	.1	71.85%
		articleType	0	61.81%
		gender	.2	75.04%
		gender	.1	66.62%
		gender	0	53.77%
		gender and articleType	.2	60.23%
		gender and articleType	.1	51.04%
		gender and articleType	0	38.41%

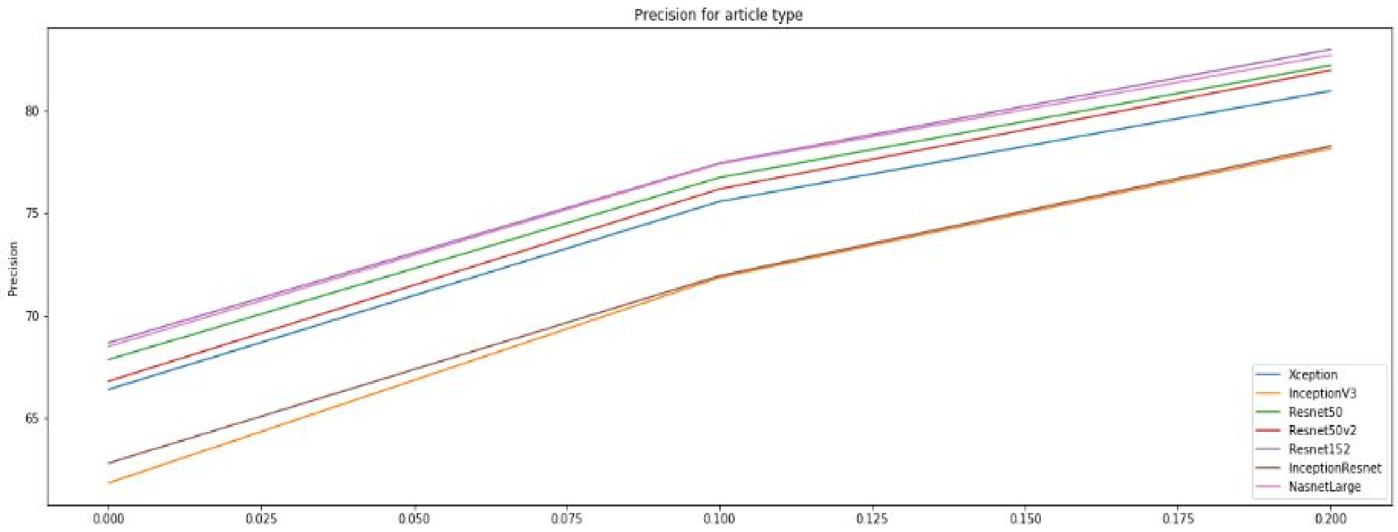
Model	TopN	Feature	Tolerance	Precision
Resnet50	10	articleType	.2	82.21%
		articleType	.1	76.73%
		articleType	0	67.84%
		gender	.2	81.02%
		gender	.1	74.24%
		gender	0	63.67%
		gender and articleType	.2	67.50%
		gender and articleType	.1	58.92%
		gender and articleType	0	47.03%
Resnet152	10	articleType	.2	82.99%
		articleType	.1	77.44%
		articleType	0	68.66%
		gender	.2	81.39%
		gender	.1	74.75%
		gender	0	63.94%
		gender and articleType	.2	68.48%
		gender and articleType	.1	59.98%
		gender and articleType	0	47.80%

Model	TopN	Feature	Tolerance	Precision
Resnet50V2	10	articleType	.2	81.97%
		articleType	.1	76.17%
		articleType	0	66.78%
		gender	.2	79.32%
		gender	.1	71.74%
		gender	0	60.26%
		gender and articleType	.2	65.50%
		gender and articleType	.1	56.42%
		gender and articleType	0	44%
InceptionResnet	10	articleType	.2	78.27%
		articleType	.1	71.93%
		articleType	0	62.78%
		gender	.2	73.73%
		gender	.1	65.49%
		gender	0	53.68%
		gender and articleType	.2	59.01%
		gender and articleType	.1	49.99%
		gender and articleType	0	38.61%

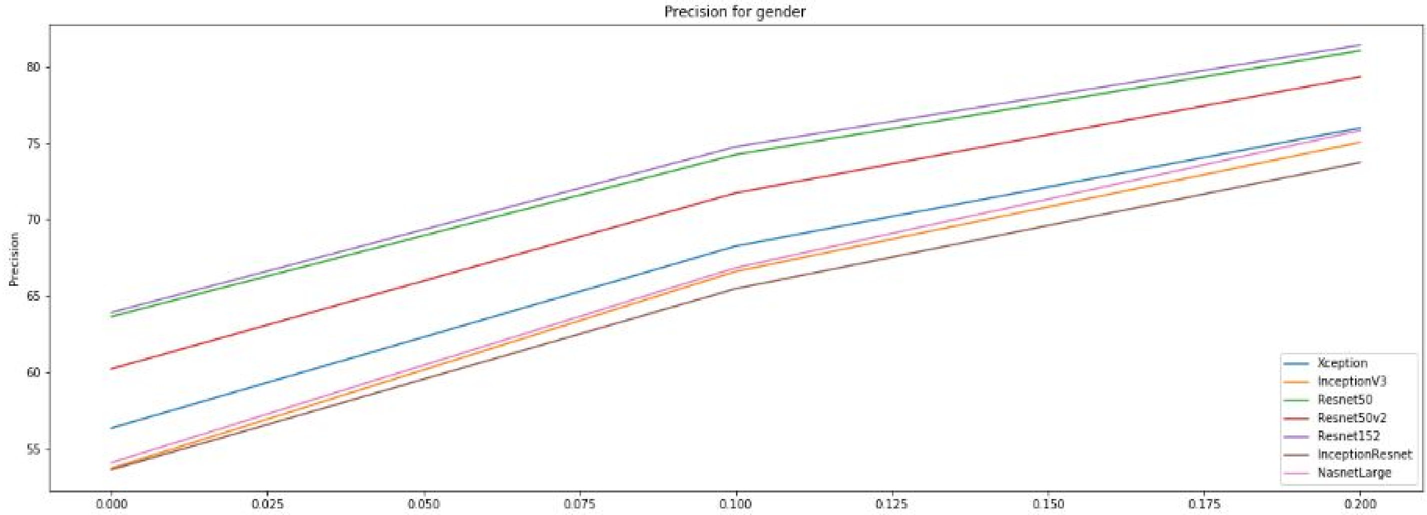
Model	TopN	Feature	Tolerance	Precision
NasnetLarge	10	articleType	.2	82.71%
		articleType	.1	77.40%
		articleType	0	68.48%
		gender	.2	75.81%
		gender	.1	66.86%
		gender	0	54.14%
		gender and articleType	.2	64.91%
		gender and articleType	.1	56.12%
		gender and articleType	0	43.75%

Decision

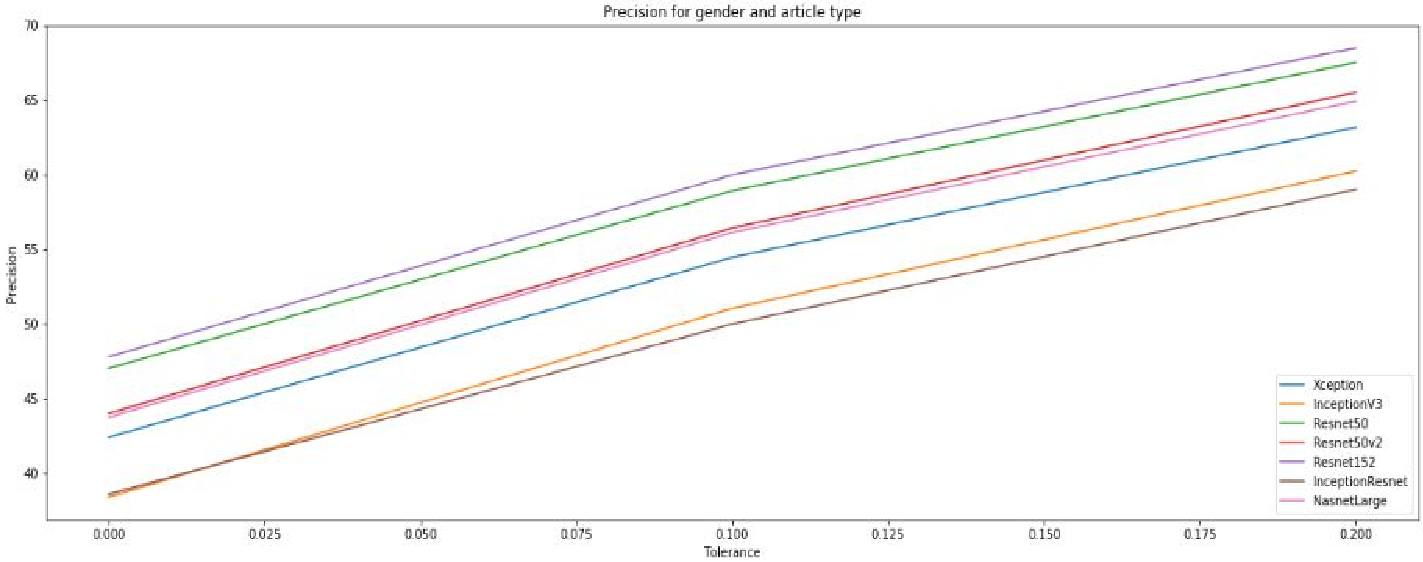
The plot for precision - tolerance for articleType as feature used for calculating precision is given below:



The plot for precision - tolerance for gender as feature used for calculating precision is given below:



The plot for precision - tolerance for both articleType and gender as feature used for calculating precision is given below:



As we can clearly see from the above plots, if we use the Resnet152 model for feature extraction, it gives the best results for all the important features chosen but the size of this model is 232MB. So it will take more space in the memory.That will be a problem as we need to keep 44k embeddings for getting faster recommendation for unknown images so memory issues will arise. But we can also see from the above plots that Resnet50 is a close second but it’s size is just 98MB that is almost 125% lesser than Resnet50.

So, we’ll be using ***Resnet50*** for feature extraction.