Labelling online illegal wildlife advertisements with Active learning-SVM

Sai Manohar Reddy P

The Problem

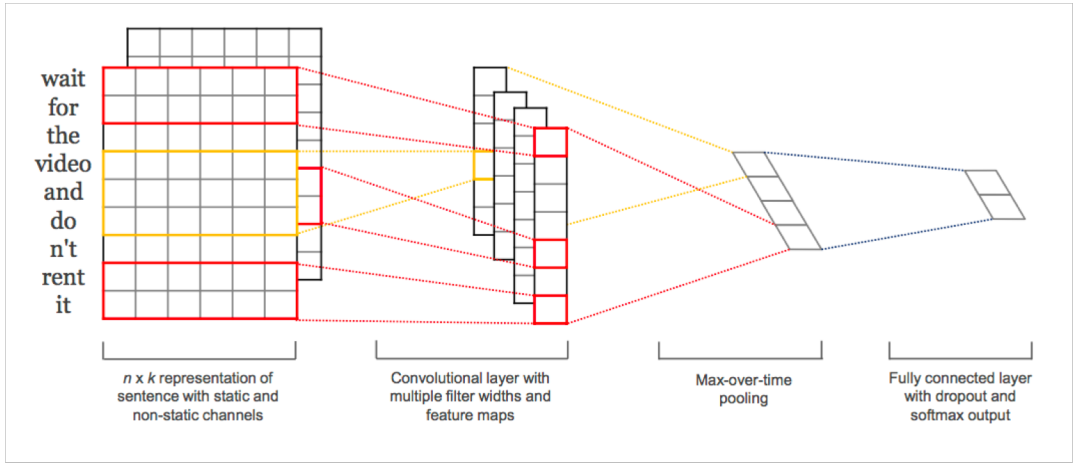
- Today, there are many wildlife species which are endangered.
- Illegal wildlife trade on internet has been a major problem to deal with for governments and international animal conservation organizations.
- Examples of these products in trade are ivory tusks, tiger skin etc.

Approach

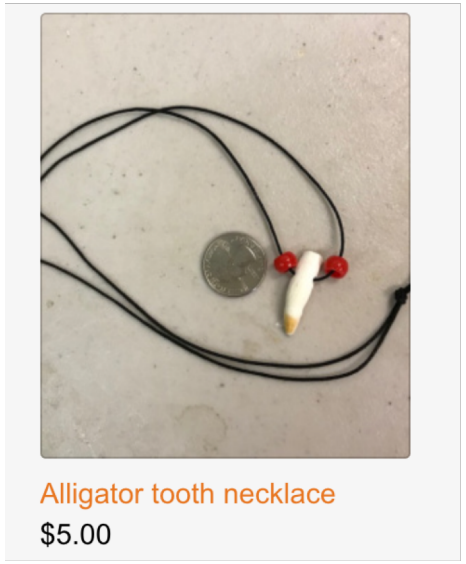
- Dr. Chakraborty(my advisor) from school of informatics has been working on this problem with various governments. For this, we have collected over 2.4 million advertisements by web-craping and web-crawling 37 listing websites like eBay, craigslist etc.
- This problem is split in two parts, First part is to build a binary classifier to classify between wildlife and non-wildlife products. Second part is to identify the illegal wildlife from the advertisements.

Prior work

- I have been working on the first part of the problem, we spent a large amount of time cleaning the data and experimenting basic machine learning algorithms to look for key features.
- So far I am able build a 1-D CNN and a bag of words – MLP (multi-perceptron network).



Dataset



2.4 million of these kinds of ads are collected and converted from html to json format.


```
{'description': '',
 'image_url': 'https://www.boneroom.com/uploads/4/8/1/1/48118243/s521972503441136676_p4592_i1_w320.jpeg',
 'price': {'amount': 5.0, 'currency': '$'},
 'title': 'Alligator tooth necklace',
 'url': 'https://www.boneroom.com/store/p4592/Alligator_tooth_necklace.html'},
```


- Then the data is cleaned and preprocessed.
- Each data point has features like title, price, title, URL, image.
- For this project, I am only using title and NLP machine learning methods.
- A sample of 20,000 ads are being used for this project.
- I labeled around 700 ads myself for the training dataset.

TF-IDF

$$w_{x,y} = tf_{x,y} \times \log\left(\frac{N}{df_x}\right)$$

TF-IDF

Term x within document y

$tf_{x,y}$ = frequency of x in y

df_x = number of documents containing x

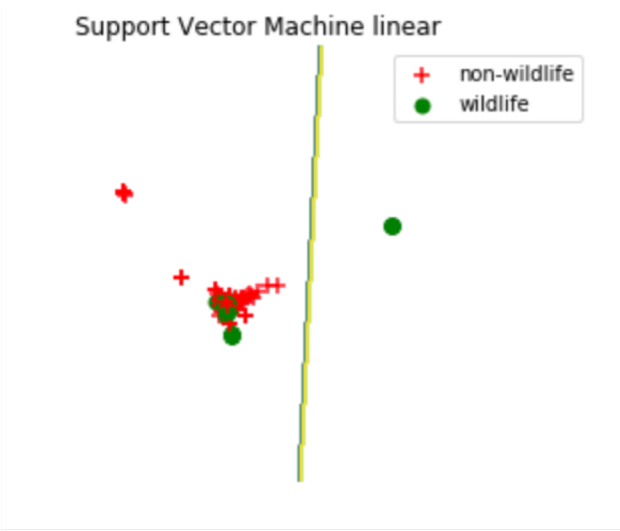
N = total number of documents

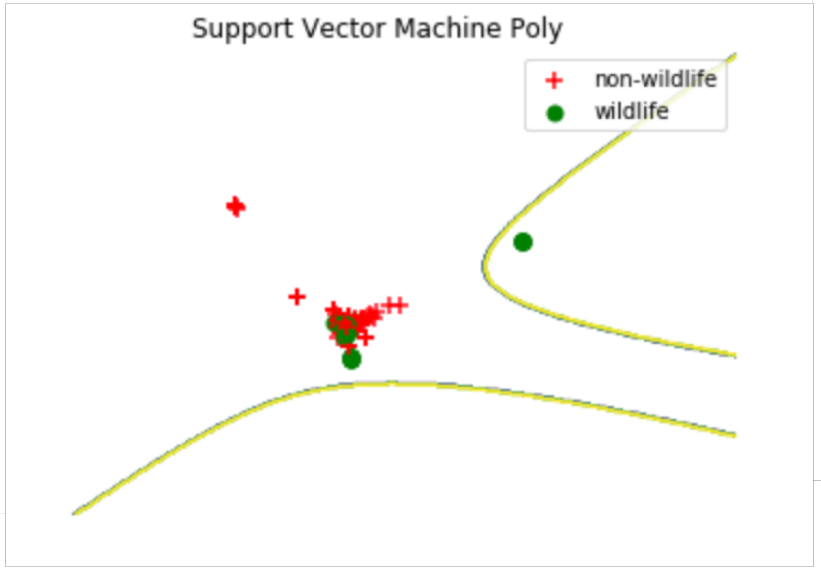
- Term frequency – inverse document frequency (TF-IDF) is used to score each data point i.e. title in the dataset.
- This created a vocabulary of 21664 words.
- The final dataset which is to trained is now 20000 x 21664 sparse matrix.
- This includes the initial training dataset.

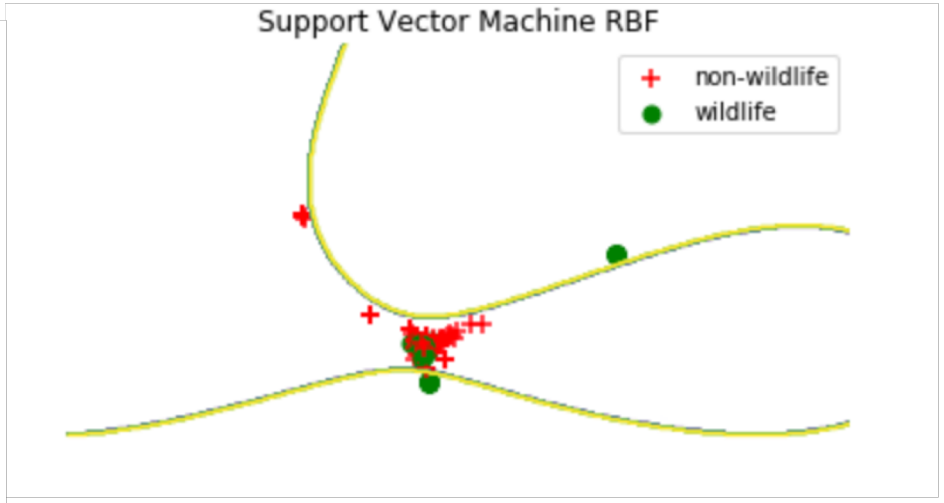
SVM

Kernels:

- Linear
- Poly
- RBF

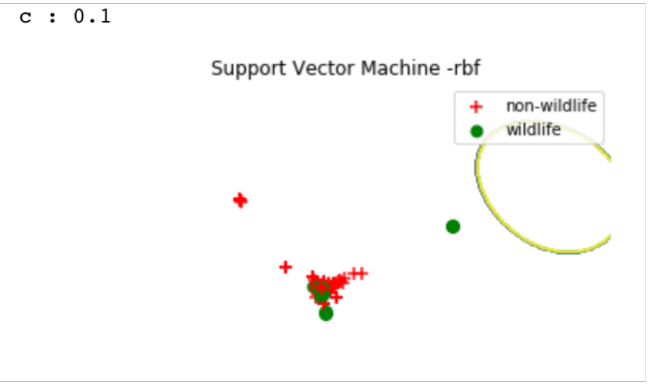


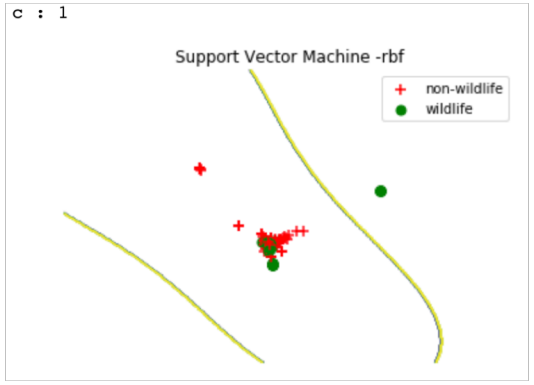


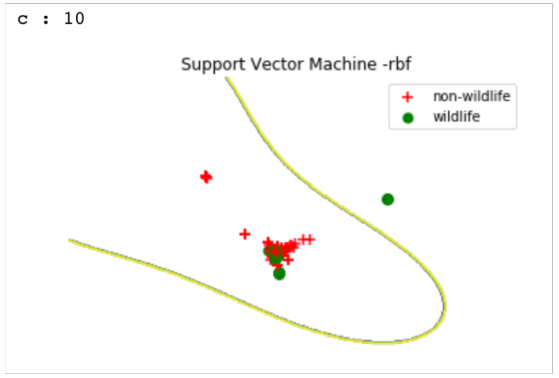


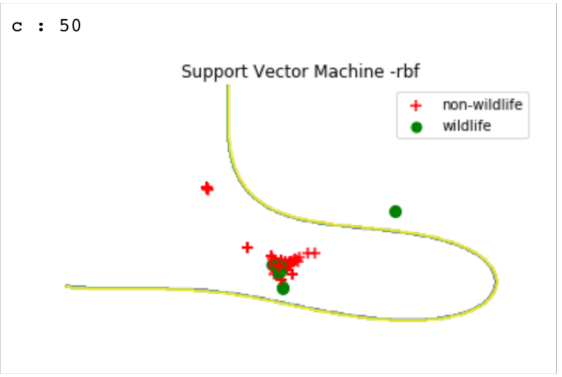
C=1000

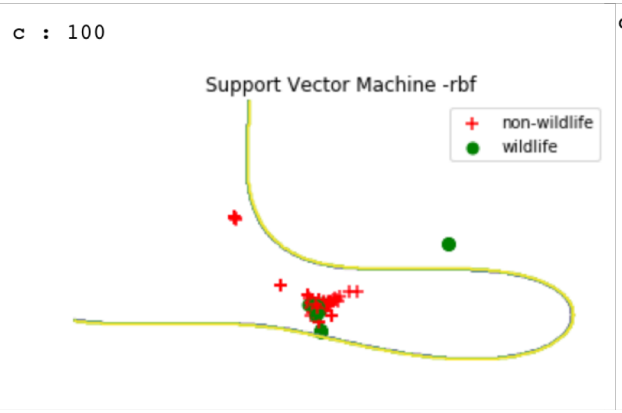
Model's reaction to regularization parameter C

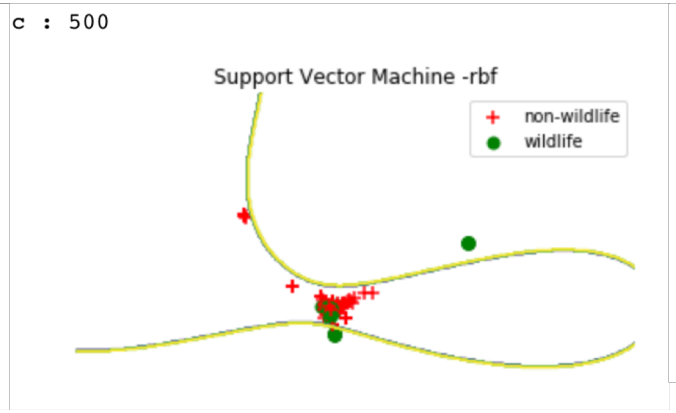


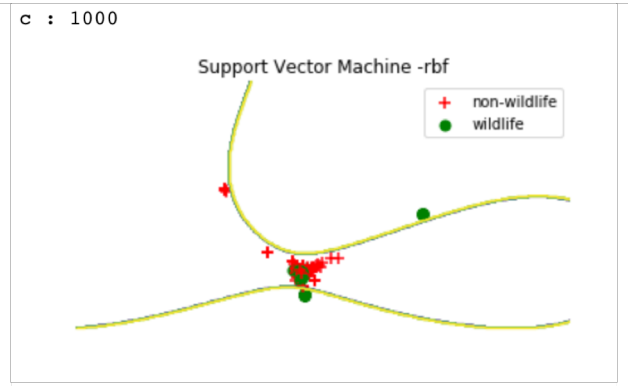


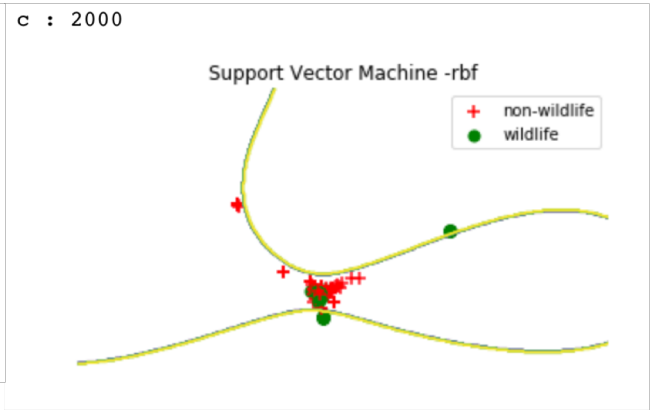




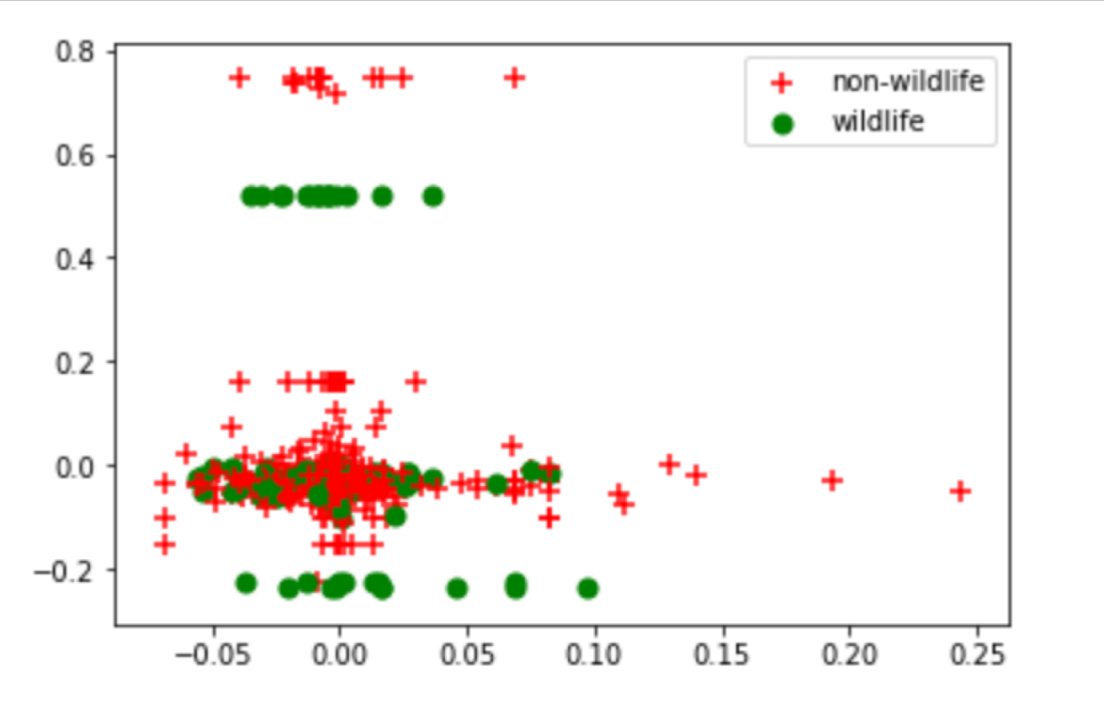


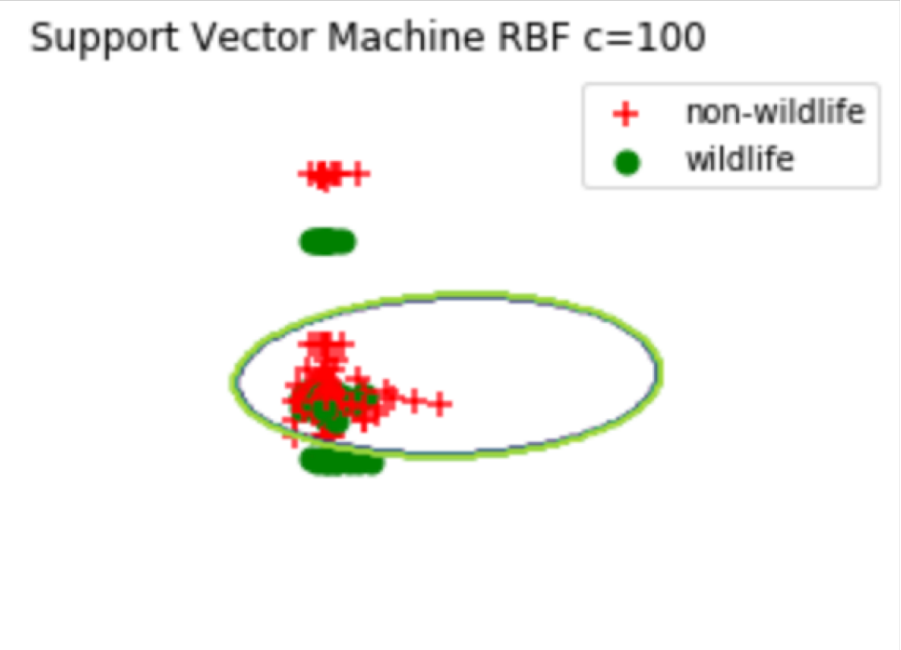






What's happening when we build the model without outliers – PCA 2D



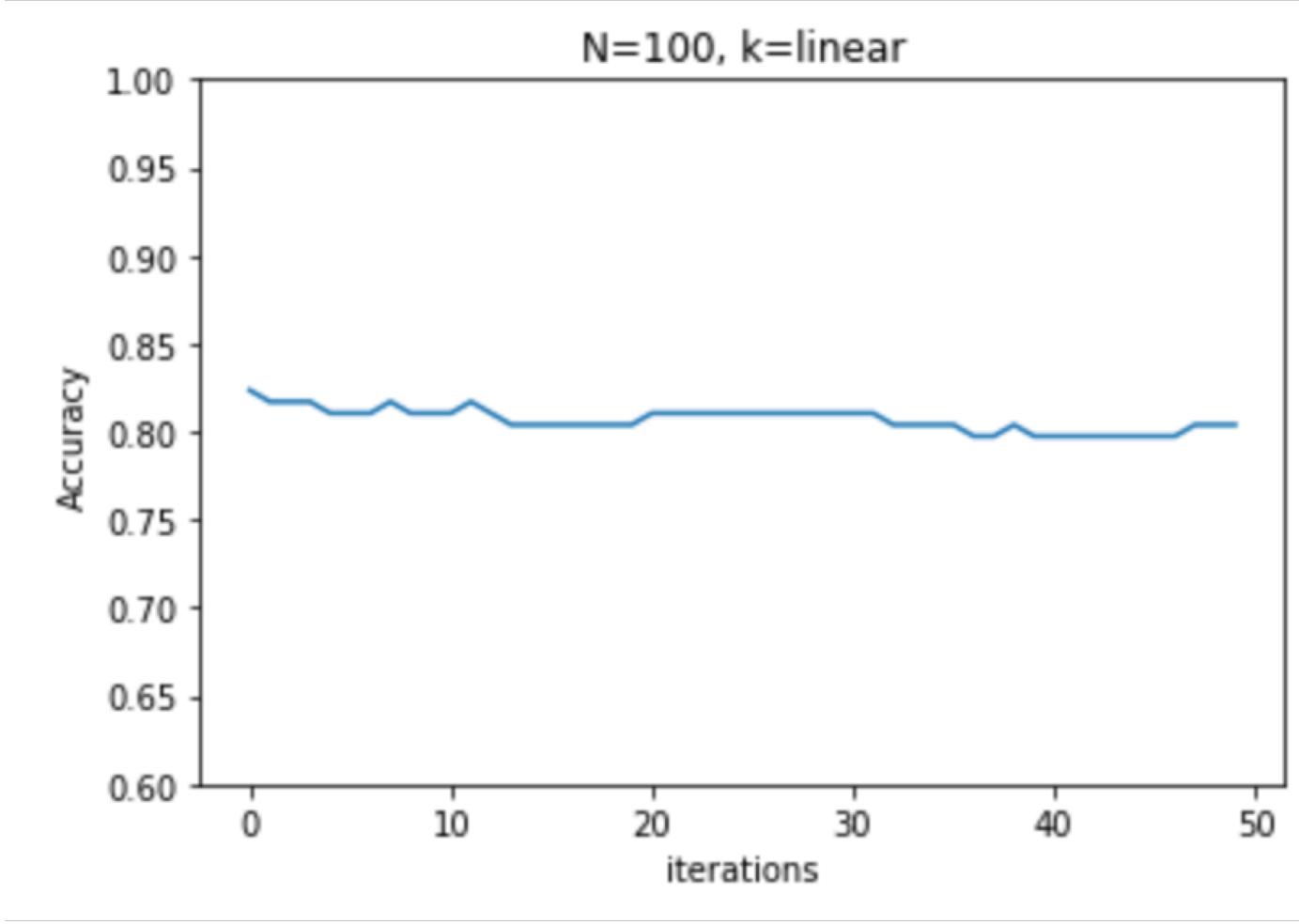


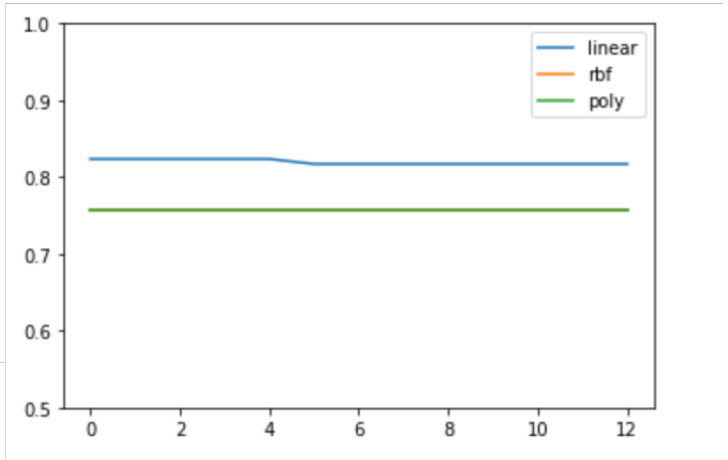
This means reducing the dimensionality is removing lot of important features.

Active learning

- The Model first is trained on the initially labelled dataset
- Next, N number of randomly chosen data points from the unlabeled dataset are predicted and the model is retrained and updated with the newer hyperplane.
- This process is continued till all data points are labeled while accuracy was logged.

Model Accuracy





Linear kernel in higher dimensions is giving better accuracy from the experiments.

Accuracy on an average stayed at: 82.64%

Comments

- Still working on cross validation, hopefully will submit with that done before final submission next week.
- We are able to detect extreme data points accurately.
- Less false positives, but higher false negative.
- We are using only ‘title’ to perform our classification, because of missing data in ‘description’ and prices in different currencies eytc. If we can add more features better results can be expected
- How can I use this model? I can add this to current models to give an ensembled decision.

Accuracy compared to prior models

1.	1-D CNN	89.2%
2.	BOW MLP	83.4%
3.	**SVM-Linear**	**82.64%**

Questions