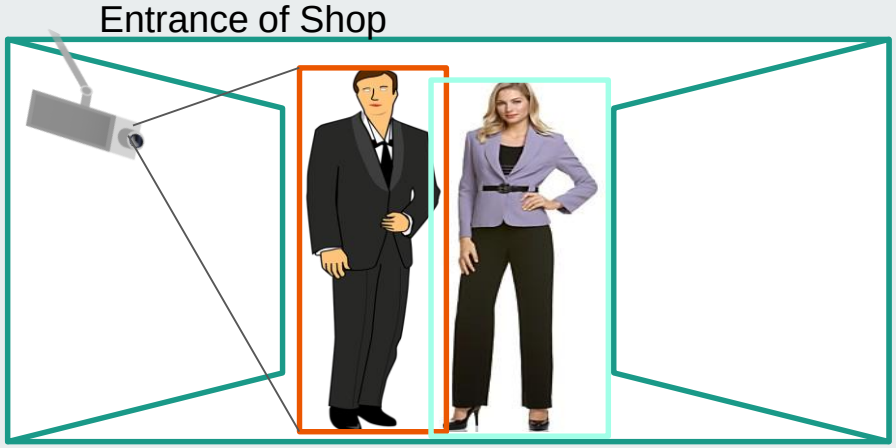
Machine Learning Based Image Classification System to Predict Changing Fashion Trends

Motivation and Problem Definition

Who ever enters the shop following information is captured using, person detection and many classification models



- A garment shop owner wants to keep stock of such garments which people are currently liking.
- To keep appropriate stock of garments, shop owner need to know changing fashion trends.
- This will help in converting the visitor into a customer.
- Fashion trends change from place to place.
- Local fashion trends can be extracted using video analytics, image processing and ML.

Out of all possible classification models, we will focus on Model which does classification based on Sleeve Length.

Method

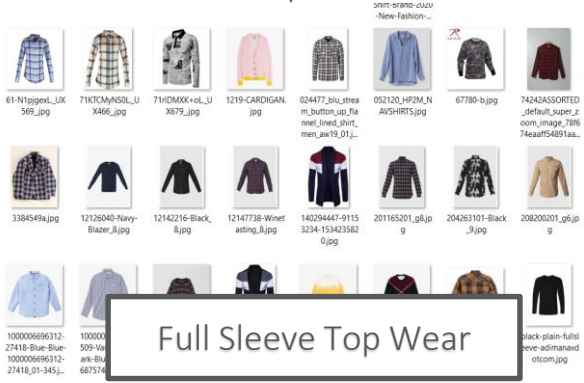
- Image Data Collection
- Data Pre-Processing
- ML Algorithms along with Hyperparameter tuning- Results & Performance Measures

Image Data Collection:

Images Downloaded using following websites.

Google amazon Flipkart YAHOO! Bing

Images of following 3 Categories were downloaded.



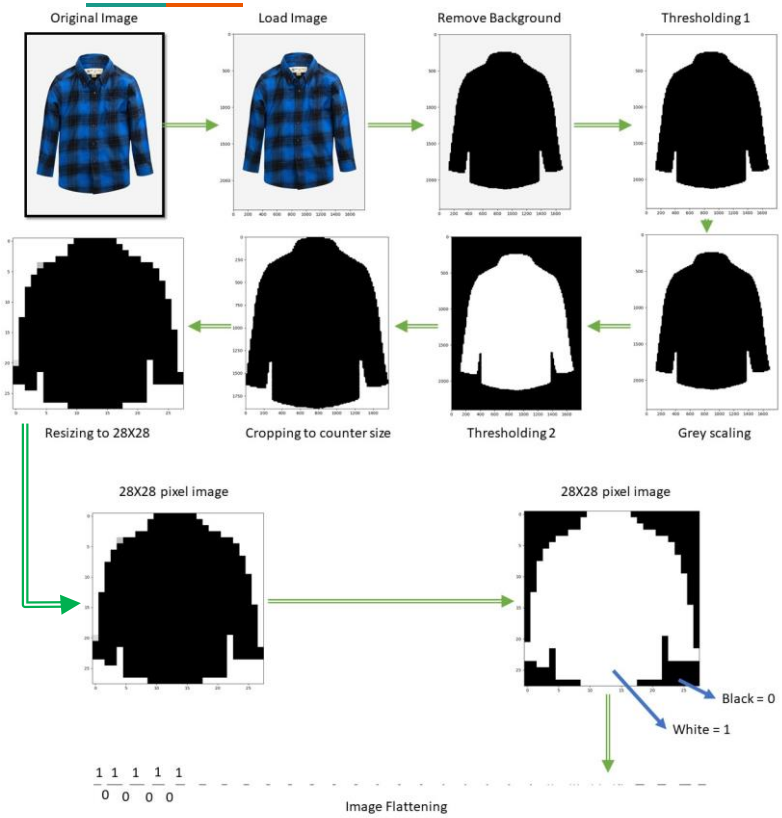
Images in each category is mixture images from categories as well:

1. Gender – Male and Female
2. Size
3. Pattern
4. Type of Top Wear – Shirt, T-Shirt, Kurta, Jacket, Blazer, Sweater, etc.

This is to make model more generalised

Data Pre-Processing:

We have Image Data and we are not going to use CNN so data-pre-processing become extremely important, so as to get good model accuracy.



1. Category wise store images in 3 different folders.
2. Load Image one by one into algorithm from folder.
3. Remove Background so as to get only shadow of top wear.
4. Use threshold to convert it into pure black and white image. (image is still in RBG format)
5. Use grey scaling to convert RBG image into single channel grey scaled image.
6. Find contours in the image which is top wear in our case. Crop image to bounding box size of contour.
7. Resize image to 28X28 pixel. (This ensure all images will have same size, also this reduces number of features)
8. Apply Max Min scaling to convert 0-255pixel data to 0-1 data.
9. Image flattening to convert 28X28 data array into 1X784 data.
10. Save this image as row in csv file along with image label as folder name.

Some example output of 28X28 pixel representation of actual image from each category:

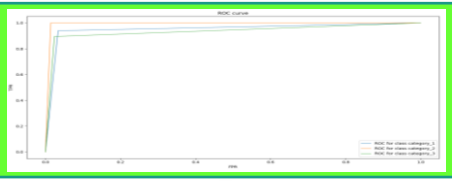
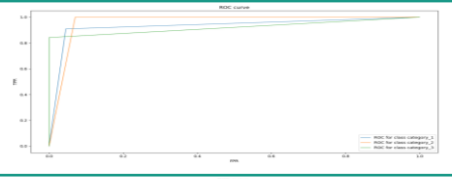
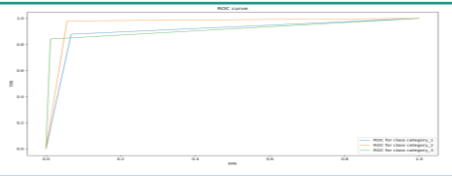
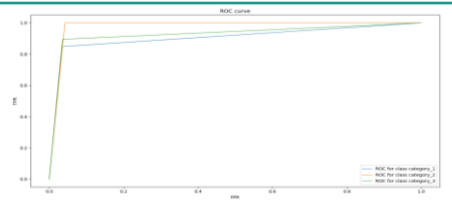
Full Sleeve Top Wear

3/4th Sleeve Top Wear

Half Sleeve Top Wear

ML Algorithms along with Hyperparameter tuning- Results & Performance Measures

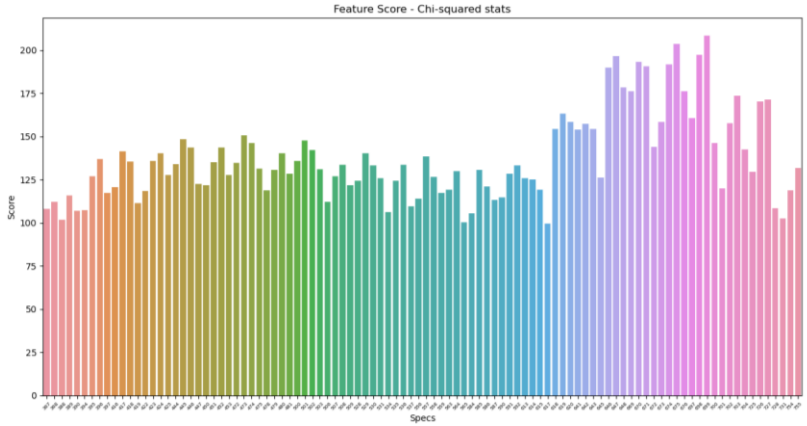
All Features

Model	Model Score (Train)	Model Score (Test)	Confusion Matrix (Test)	Area under curve of ROC (Test)	Tuned Hyper parameter (Optimized using test data set)	ROC Curve
K Nearest Neighbours	0.9876	0.9508	[[31 0 2] [0 51 0] [3 1 34]]	0.9641	K=1	
SVM with RBF kernel	0.9545	0.9262	[[30 3 0] [0 51 0] [4 2 32]]	0.94231	Gamma=0.004 C=1.5	
Decision Tree Classifier	0.9421	0.9098	[[28 3 2] [1 50 0] [4 1 33]]	0.9318	max_depth = 5 min_samples_leaf = 3	
Gradient Boosting Classifier	0.9586	0.9262	[[28 2 3] [0 51 0] [3 1 34]]	0.9441	min_samples_leaf = 1 max_depth =1 n_estimators=200	

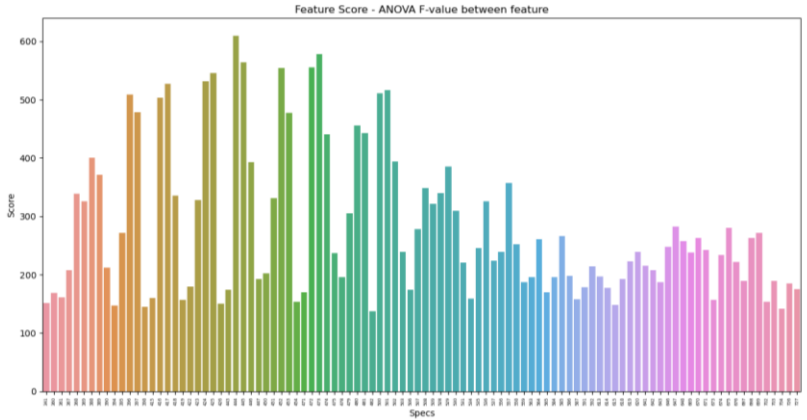
Why other models not tried:

1. Deep Neural Network is not used due to lack of data
2. Logistic Regression is not used as it is a simple linear model and we need non-linear boundary in the given problem
3. Generative Classification Models such as LDA and QDA are not used as we don't know considerable data generation process. Also data does not follow gaussian distribution

Feature Selection



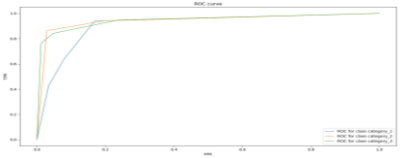
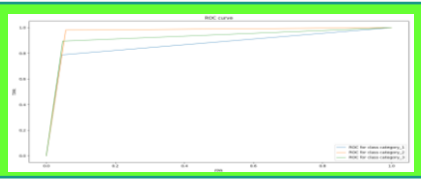
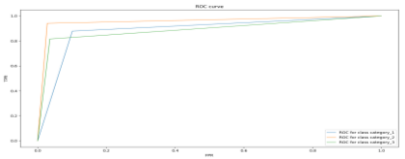
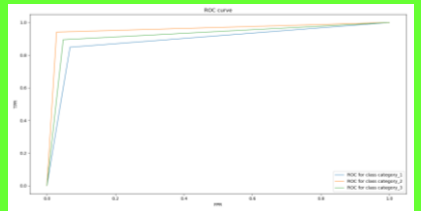
Chi-squared stats has been calculated between each feature and class.



F-test estimate the degree of linear dependency between any feature and discrete target variables (class).

- 111 features out of total 784 features has been selected.
- As all features are binary and have value either 0 or 1, we choosen these feature selection technique.

ML Algorithms along with Hyperparameter tuning- Results & Performance Measures

Model / Selected Features	Model Score (Train)	Model Score (Test)	Confusion Matrix (Test)	Area under curve of ROC (Test)	Tuned Hyper parameter (Optimized using test data set)	ROC Curve
K Nearest Neighbours	0.8780	0.7950	[[29 1 3] [6 44 1] [5 1 32]]	0.8735	K=3	
SVM with RBF kernel	0.8636	0.9016	[[26 3 4] [1 50 0] [3 1 34]]	0.9255	Gamma=0.003 C=1.45	
Decision Tree Classifier	0.9152	0.8852	[[29 1 3] [3 48 0] [6 1 31]]	0.9174	max_depth = 5 min_samples_leaf = 2	
Gradient Boosting Classifier	0.8842	0.9016	[[28 1 4] [3 48 0] [3 1 34]]	0.9283	min_samples_leaf = 1 max_depth =1 n_estimators=200	

Conclusion:

As we can see K Nearest Neighbours with K = 1, when trained with all features gives best result. So we select this model for classification.