Otto Group Product Classification Challenge

CS 514 - Applied Artificial Intelligence

Project 5

Skynet

Introduction

Background

Retail is one of the important business domains for machine learning applications. Recent application of machine learning in retail include customer segmentation, optimal pricing, customer churn detection and recommendation. Product Classification and categorization is one such area which can be modeled as supervised learning problem. Product categorization is important in two ways

1.It is crucial for a customer to find what they are looking for. The retail might end up losing sales because of poor classification and categorization. Studies have shown that typically 8 – 10% of product URLs available on a given shelf are misclassified.

2. Quality product analysis depends heavily on the ability to accurately cluster similar products.Many identical products get classified differently which makes further analysis difficult.

Dataset

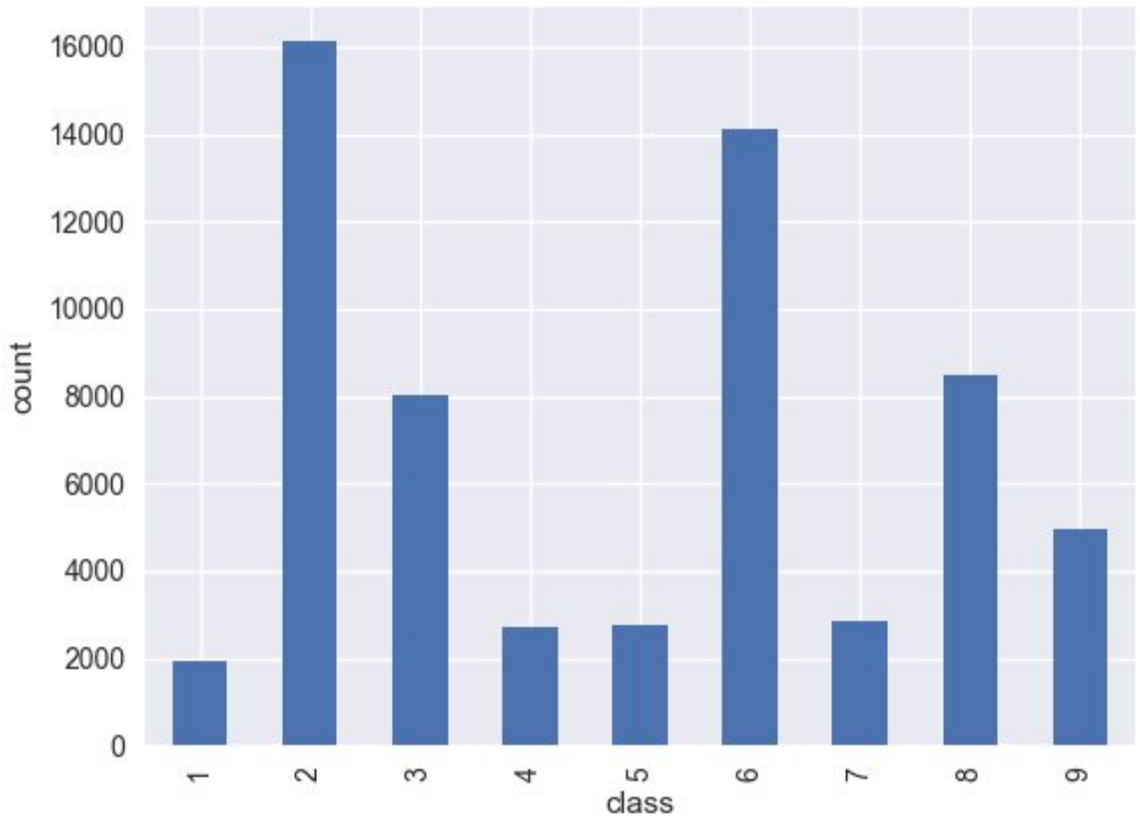
We are using otto Group Product Classification Dataset from Kaggle.Each row corresponds to a single product. There are a total of 93 numerical features

Objective

The objective is to build a predictive model which is able to distinguish between main product categories. The model on feeding product features should give one of the nine categories as output.

Exploratory Data Analysis

Class distribution

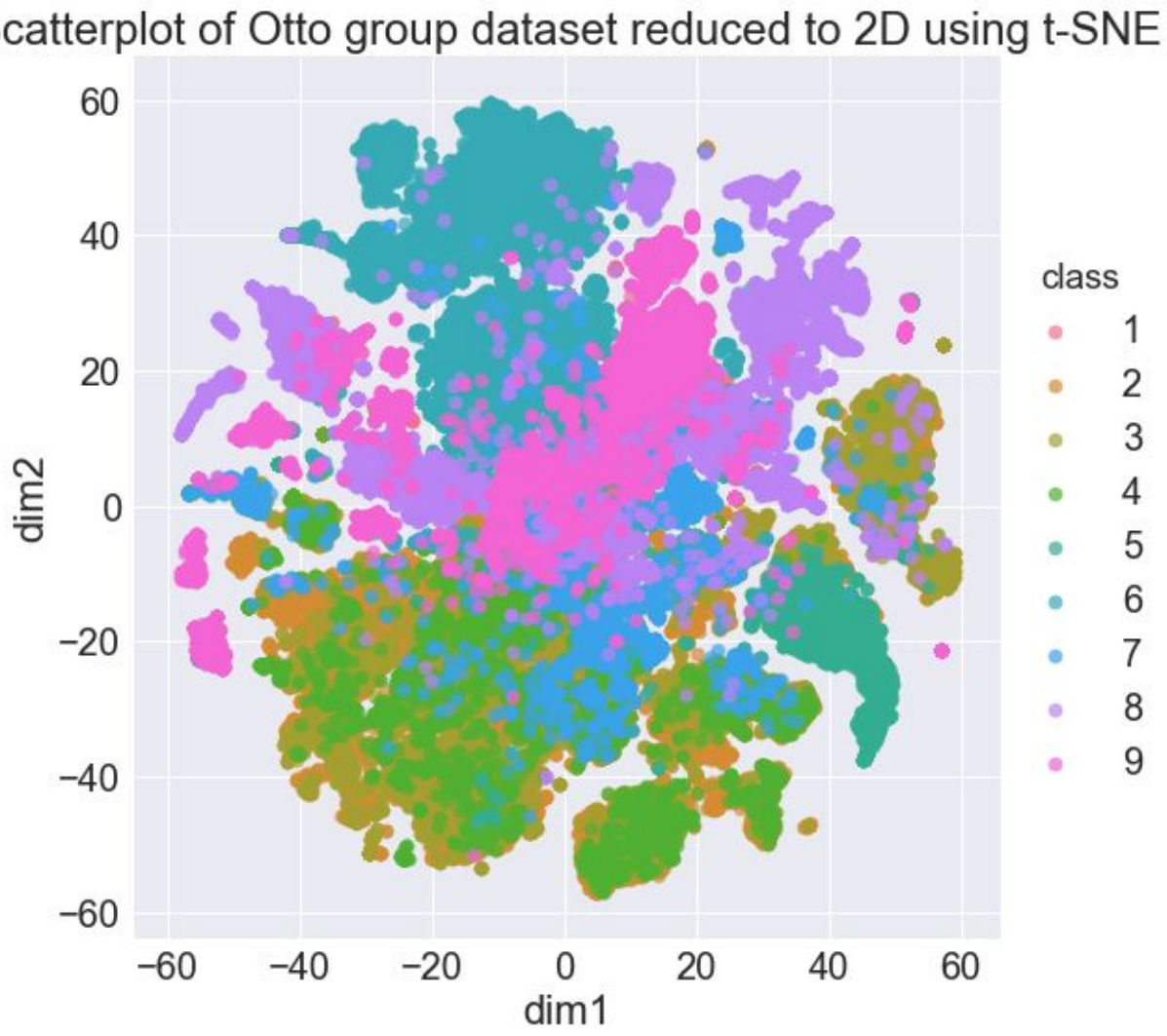


As seen in the bar chart , the dataset has somewhat imbalanced distribution of class labels.

Dimensionality Reduction

Since we have high dimensional data with as much as 93 features , we need to project it to lower dimension for intuitive visualization. I have used two methods for dimensionality reduction which are listed below

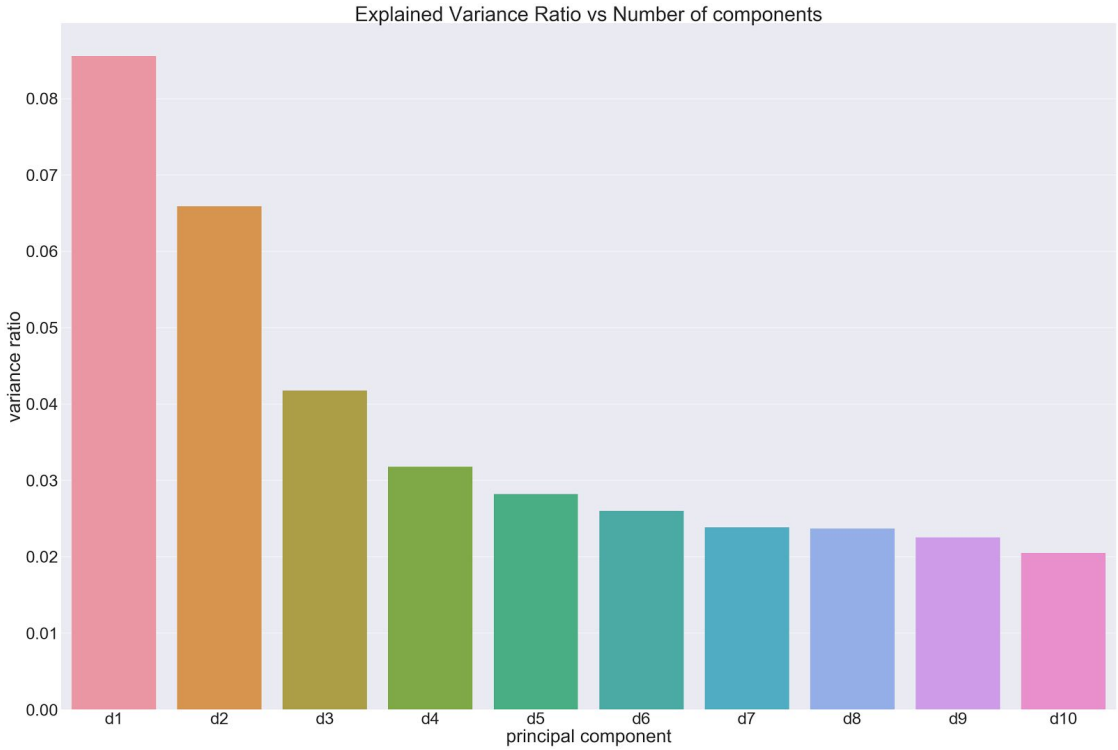
T-sne (t - distributed Stochastic Neighbor Embedding)



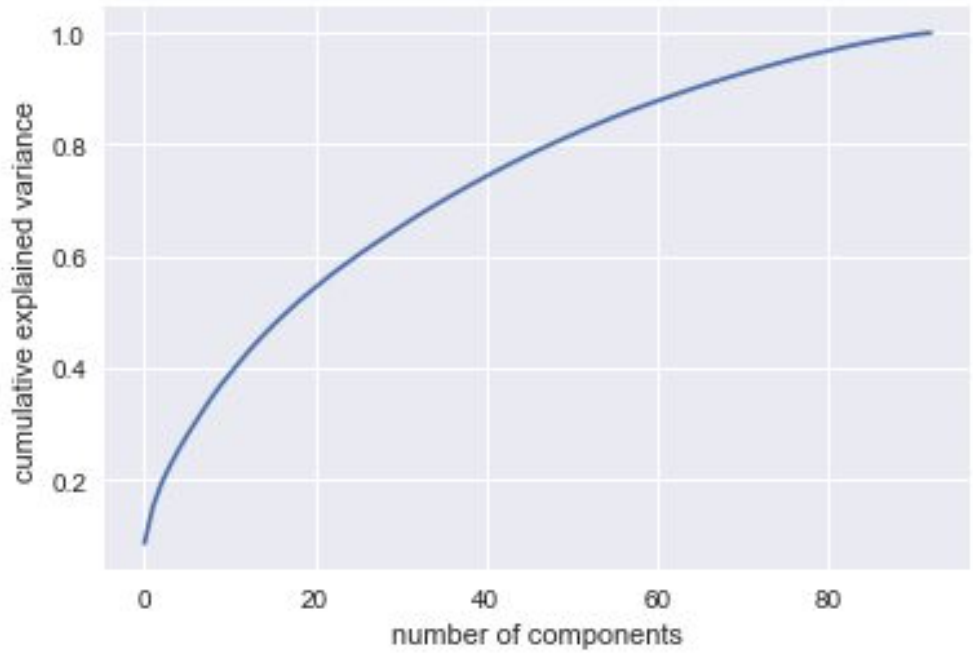
Principal Component Analysis

ScatterPlot in 2 Dimension

Explained Variance Ratio Bar Plot



Cumulative Variance Plot



As seen in plots, first few components (let's say 20) only explained around 50% variance in data. So dimensionality reduction technique won't probably help in predictive modeling.

Approach

I plan to use several ML algorithms and ensemble of the algorithms to see if we can improve the performance of the model.

The base estimators (ML) algorithms used are

1. Logistic Regression
2. K-Nearest Neighbor

Similarly, the ensemble methods used are

1. Bagging (Random Forest)
2. Boosting (XGBoost)
3. Weighted Combination of Classifiers (minimization function used is SLSQP)

Performance Evaluation

Since this is multiclass classification problem , we will be using mlogloss (multiclass logarithmic loss) as performance evaluation metrics.Log Loss quantifies the accuracy of a classifier by penalising false classifications.

Mathematically, it is equal to,

$$-\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{M} y_{ij}\log p_{ij}$$

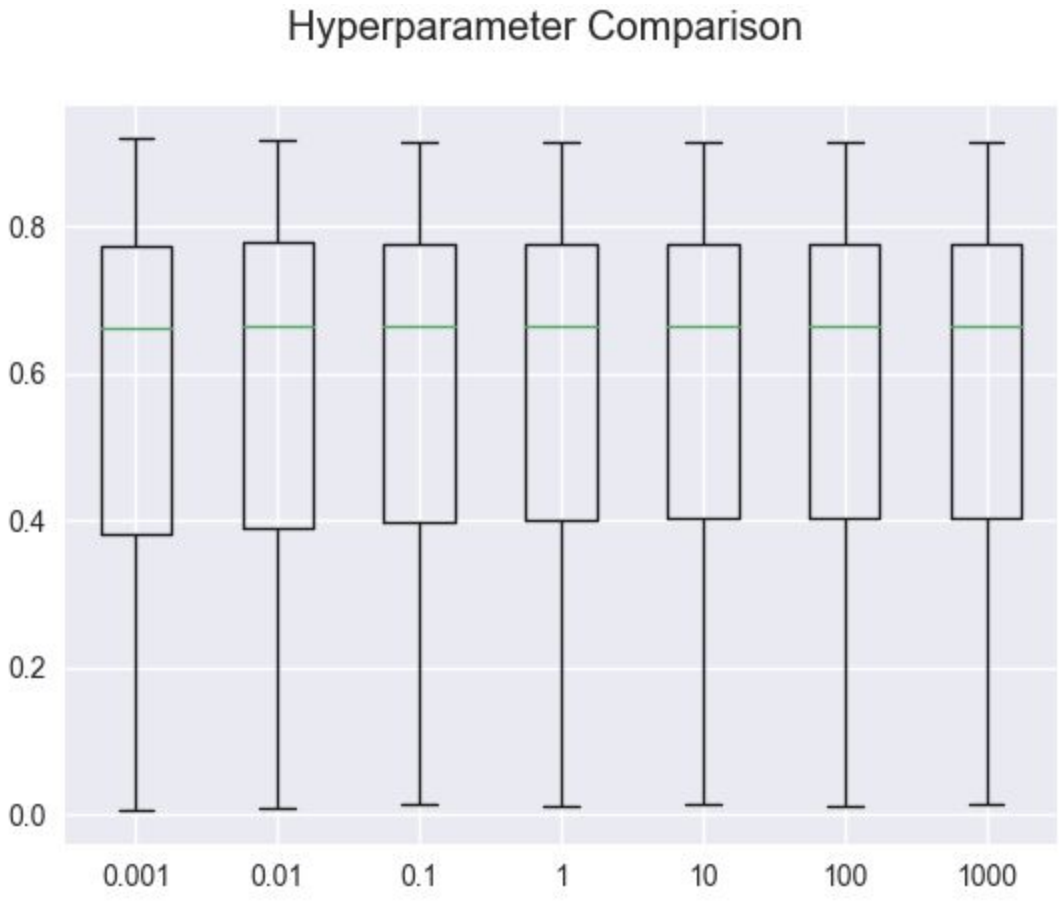
Y_{ij} is 1 if example i is classified as class j. Now if the point is misclassified, very small probability is assigned forexample 'i' for class 'j' which adds large value to cumulative sum. Largest value would occur if 0 probability is assigned to the example for its true label .It will be log(0) = -1. However if the predictor is confident and assigns it a probability as 1, it won't contribute towards overall logloss as log(1) = 0

The idea is to train the model using training set and create the submission file for kaggle submission. The mlogloss as reported by kaggle is included in the report.

Results

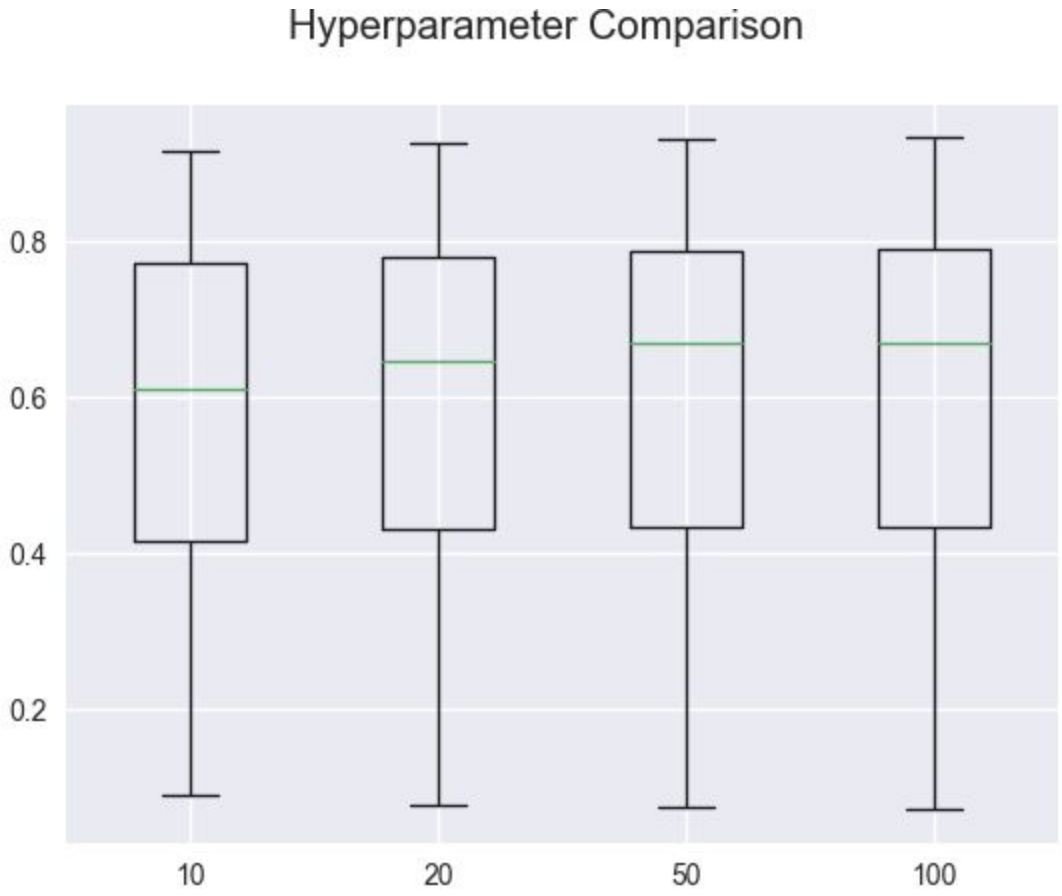
Hyperparameter Tuning using cross validation

First we observe the result of hyperparameter tuning for logistic regressionHyperparameter for logistic regression is regularization parameter which helps to prevent overfitting the data .



Here y-axis shows the accuracy of the model while x-axis shows the value of C (regularization parameter). As seen in the plot every choice of C almost gave the similar accuracy in the validation set.

Next we observe the similar plot for random forest. It shows Hyperparameter tuning using k-fold cross validation for number of trees



Following Table shows the result of several methods used for prediction

Method	Estimators	Hyperparameter	Validation logloss	Test Dataset (kaggle) logloss	
				Private Score	Public Score
Parametric Base Estimator	Logistic Regression	C = 10		0.66909	0.66943
		C = 1000		0.66908	0.66946
Nonparametric Base Estimator	KNN	K=5		2.41600	2.34618
Ensemble: Bagging	Random Forest	N= 100		0.60173	0.58657
		N =1000		0.55586	0.55392
Ensemble: Boosting	XGBoost	default		0.64939	0.64972
Ensemble : RF and KNN	Random Forest	n=1000	0.49086578	0.52310	0.51738
	KNN	K =5	2.56764		
	Ensemble		0.4831		
Ensemble: XGBoost and KNN	XGBoost	Default parameters	0.50927	0.47449	0.47018
	KNN	K=5	5.20		
	Ensemble		0.491		

The best performance was achieved with the combination of XGBoost and KNN algorithm

Limitations

Didn’t use feature engineering, feature elimination and data transformation to improve performance of the model.