Finding Lemons: Traditional Algorithms vs Self Organizing Maps

- Volak Sin

Business Problem

- Junk cars are a significant loss
- Predicting a junk car has been fairly difficult
- The current company error rate is 12.30%

Data Summary

The purpose of the analysis is to classify the cars based on their likely hood of being a lemon/ a junk car. The data provided came from two auctions, ADESA and Manheim, had the following characteristics:

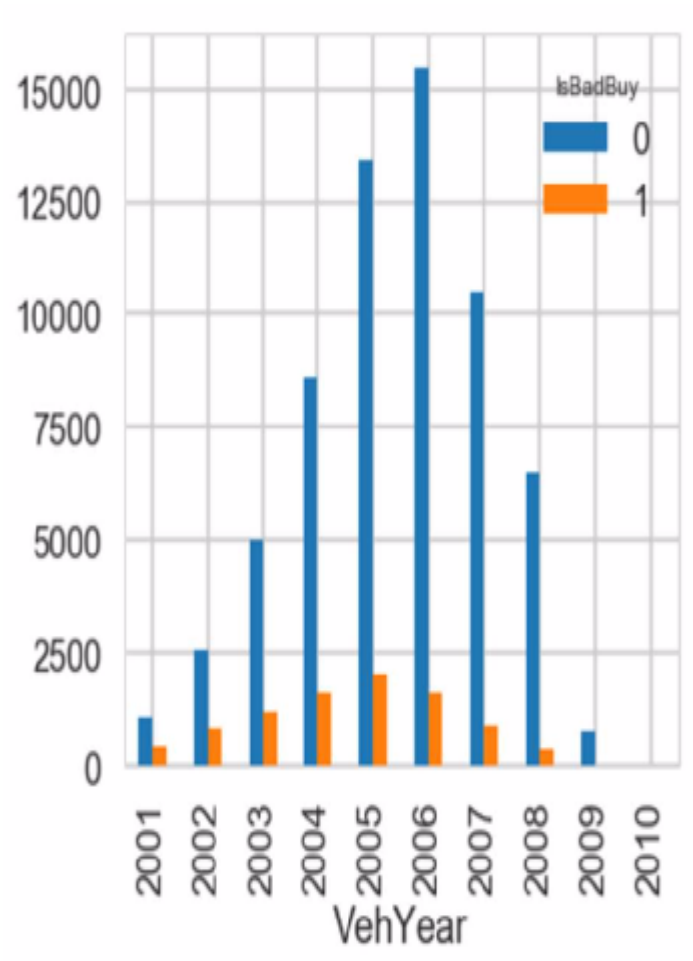
- 32 total features containing both categorical and continuous data
- A combination of float64, int64 and object data types
- Many features have 95% of their values missing

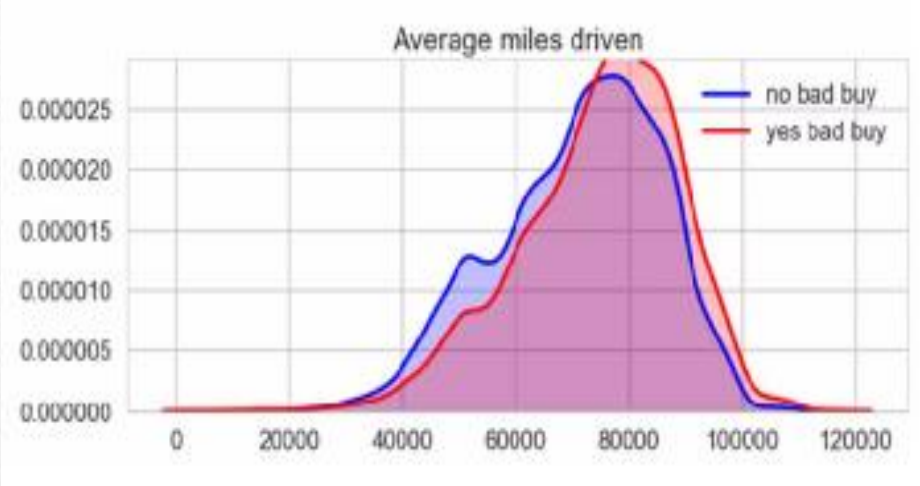
Data Wrangling

Field Name	value Type	Data Type	# Unic	Unique values	# Mis	% Missi	Impute	Definition
Auction	object	Categorical	3	ADESA' 'OTHER' 'MANHEIM'	0	0	n/a	Auction provider at which the vehicle was purchased
Make	object	Categorical	33	MAZDA' 'DODGE' 'FORD'	0	0	n/a	Vehicle Manufacturer
Model	object	Categorical	1064	MAZDA3' '1500 RAM PICKUP 2WD' '	0	0	n/a	Vehicle Model
Trim	object	Categorical	134	ST' 'SXT' 'ZX3' 'ES'	2360	3.23%	'UNK'	Vehicle Trim Level
Color	object	Categorical	16	RED' 'WHITE' 'MAROON'	8	0.01%	mode	Vehicle Color
Transmission	object	Categorical	3	['AUTO' 'MANUAL' nan 'Manual']	9	0.01%	mode	Vehicles transmission type (Automatic, Manual)
WheelType	object	Categorical	3	['Alloy' 'Covers' nan 'Special'	3174	0.043	'UNK'	The vehicle wheel type description (Alloy, Covers)
Nationality	object	Categorical	4	['OTHER ASIAN' 'AMERICAN' 'TOP	5	7E-05	all b	The Manufacturer's country
Size	object	Categorical	12	MEDIUM' 'LARGE TRUCK' 'COMPACT'	5	7E-05	mode	The size category of the vehicle (Compact, SUV, etc.)
TopThreeAmericanName	object	Categorical	4	CHRYSLER' 'FORD' 'GM'	5	7E-05	mode	Identifies if the manufacturer is one of the top three American manufacturers
PRIMEUNIT	object	Categorical	2	[nan 'NO' 'YES']	69564	0.953	"OTHE	Identifies if the vehicle would have a higher demand than a standard purchase
AUCGUART	object	Categorical	2	'GREEN' 'RED'	69564	0.953	"OTHE	The level guarntee provided by auction for the vehicle (Green light - Guaranteed
VNZIP1	int64	Categorical	153	20166 50111 72117	0	0	n/a	Zipcode where the car was purchased
VNST	object	Categorical	37	IA' 'AR' 'MN'	0	0	n/a	State where the the car was purchased
IsOnlineSale	int64	Categorical	2	[0 1]	0	0	n/a	Identifies if the vehicle was originally purchased online

**** There were significant issues with missing values and wrong data types. The sample table above describes some of the data transformation that took place before analysis.***

Exploratory Analysis





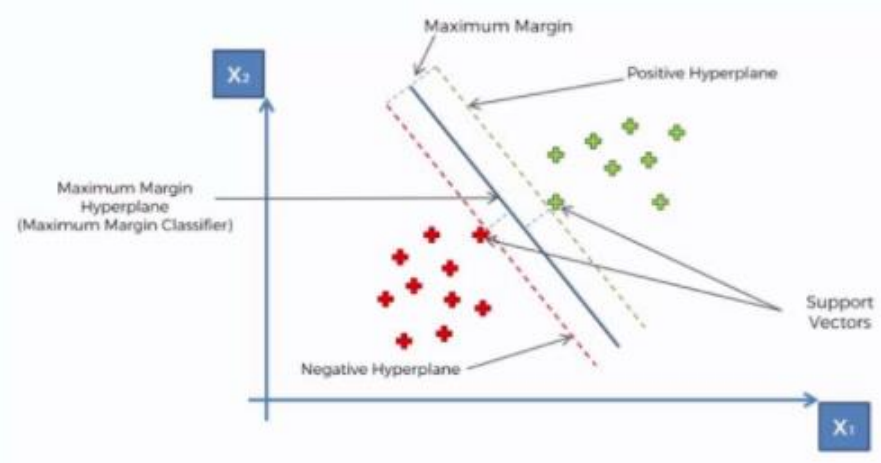
- The distribution of bad buys and non bad buys are somewhat normal
- As expected, the average miles driven for "bad buys" are higher than non "bad buys"

Logistic Regression

Confunsion Matrix		Accrucacy	89.57
15739	571	Precision	69.03
1632	604	Recall	27.01

- Even with a simple logistic regression we get an improvement in accuracy to 89.57%
- This represents a 2% improvement from past performances. Normally, we would assume a performance about human error to be ideal, but since the nature of predicting junk vehicles is complex for humans, we will seek to improve our model performance

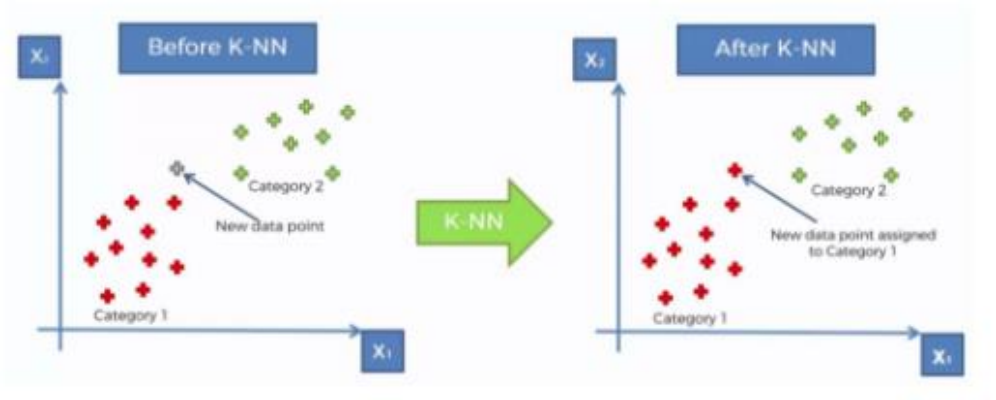
Supervised Model: K Nearest Neighbor



Confusion Matrix		Accrucacy	0.883317
15524	486	Precision	0.549583
1643	593	Recall	0.265206

- The performance of the K Nearest Neighbor is about the same to the current performance model

Support Vector Machine



Confusion Matrix		Accrucacy	0.895868
15743	267	Precision	0.693103
1633	603	Recall	0.269678

- The performance of the K Nearest Neighbor is about the same to the current performance model

Supervised Model: Naïve Bayes

#4 Posterior Probability

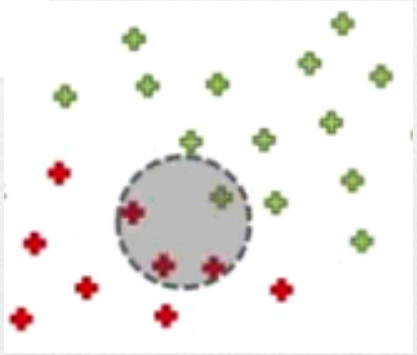
#3 Likelihood

#1 Prior Probability

$$P(Walks|X) = \frac{P(X|Walks) * P(Walks)}{P(X)}$$

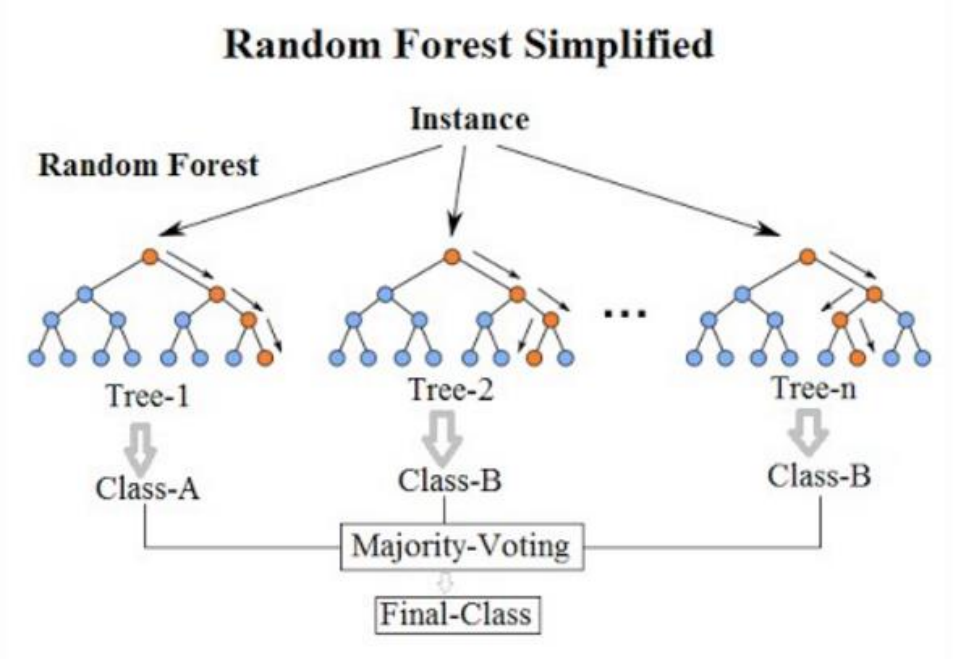
#2 Marginal Likelihood

Confunsion Matrix		Accrucacy	0.891154
15614	396	Precision	0.619962
1590	646	Recall	0.288909



- The Naïve Bayes Classifier underperformed the Logistic regression.

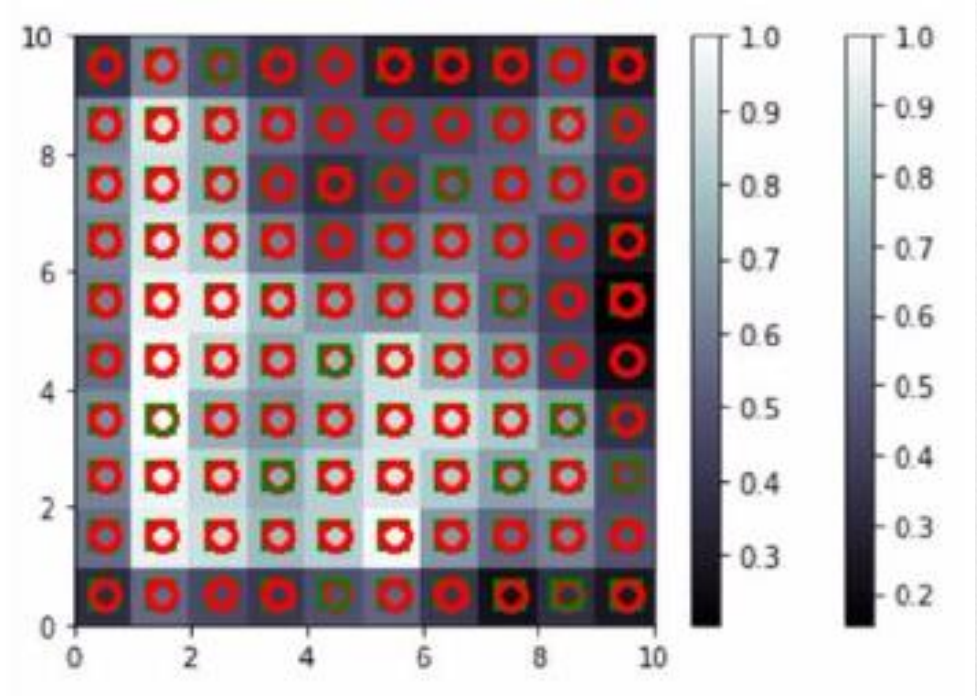
Random Forest



Confunsion Matrix		Accrucacy	0.841828
14716	1294	Precision	0.332301
1592	644	Recall	0.288014

- Surprisingly, the random forest algorithm has underperformed compared to the rest of the models ran. It also underperformed the original model.

Self Organizing Map



Self organizing maps are an unsupervised deep learning algorithm used for feature detection. The goal of the SOM is to group instances and in our case, hopefully group together junk vehicles.

Recommendations

We originally purposed the use of Self Organizing Maps as a algorithm to reduce the error selection rate of junk vehicles. Even though the algorithm was able to group vehicles that were 2 deviations away from the rest, the result of these grouping did not produce a level of accuracy greater than previous models. We were however able to used the reduced groupings and fed it into a neural network to create model that estimated probability that a vehicle would be a junk vehicle.