Table of Contents

[1. DATASET DESCRIPTION 2](#_Toc437635954)

[1.1 Problem Statement 2](#_Toc437635955)

[1.2 Background 2](#_Toc437635956)

[1.3 Motivation 2](#_Toc437635957)

[1.4 Feature Description 2](#_Toc437635958)

[1.5 Key Observations in data 3](#_Toc437635959)

[1.6 Data Preprocessing 4](#_Toc437635960)

[2. RESULTS AND APPROACH 4](#_Toc437635961)

[2.1 Class Imbalance 4](#_Toc437635962)

[2.2 Feature Engineering 4](#_Toc437635963)

[2.3 Dataset Samples 4](#_Toc437635964)

[2.4 Classifiers Tried 5](#_Toc437635965)

[2.5 Evaluation Criteria 5](#_Toc437635966)

[2.6 Results 6](#_Toc437635967)

[3. DETAILED ANALYSIS 14](#_Toc437635968)

[3.1 Accuracy Balanced v/s Unbalanced Data 14](#_Toc437635969)

[3.2 Bagging 15](#_Toc437635970)

[3.3 k Nearest Neighbor 15](#_Toc437635971)

[3.4 All Classifiers v/s each other 17](#_Toc437635972)

[3.5 Random Forest and Ensemble 19](#_Toc437635973)

[3.6 Naïve Bayes 20](#_Toc437635974)

[3.7 Data Balancing Techniques 20](#_Toc437635975)

[3.7 SVM 20](#_Toc437635976)

[3.8 Decision Trees 20](#_Toc437635977)

[3.9 Logistic Regression 20](#_Toc437635978)

[4. CONCLUSION 20](#_Toc437635979)

[5. REFERENCES 20](#_Toc437635980)

# DATASET DESCRIPTION

## Problem Statement

We are trying to ***predict*** if a car bought at an auction by an auto dealer is a **Good buy** or a **Bad buy.**

## Background

` When we go to buy a car at auto dealership we expect to get a good selection of car. Also we expect to trust in the condition of the car we are buying. These auto dealerships buy these cars from auctions and they have the same intent as us. However, the problem which dealers face is with the cars which have some serious conditions and they turn out to be bad buys. These are called “kicks”, and can happen due to variety of reasons.

## Motivation

It would greatly benefit both the auto-dealers and the end buyers if there is a way to determine a car will be a kicked car. A simple analysis of the same is presented below.

|  |  |  |
| --- | --- | --- |
| **Legends** | **Amounts** | **Information** |
| Average number of cars bought and sold by a dealer | 15000 | By Auction Direct |
| Kicked cars (%) | 12.3 % | By Dataset |
| Number of kicked cars | 1845 | By Calculation |
| Average price of car sold | $ 10000 | By Auction Direct |
| Profit on good sale | $ 2000 | Average profit = 20% |
| Profit on bad sale (kicked car) | $ 500 | Due to repairs, etc. |
| Loss of potential profit | $ 1500$ |  |
| Total loss | **$ 2767500** |  |

*Note: All values are assumed values as per Auction Direct (a company which deals in second hand cars)*

This huge amount of potential profit can be converted into actual profit if there exists a model to predict a kicked car. Thus we chose this dataset.

## Feature Description

Dataset contained 32 unique features with 73,041 samples.

|  |  |
| --- | --- |
| **Field Name** | **Definition** |
| RefID | Unique (sequential) number assigned to vehicles |
| **IsBadBuy** | **Identifies if the kicked vehicle was an avoidable purchase** |
| PurchDate | The Date the vehicle was Purchased at Auction |
| Auction | Auction provider at which the vehicle was purchased |
| VehYear | The manufacturer's year of the vehicle |
| VehicleAge | The Years elapsed since the manufacturer's year |
| Make | Vehicle Manufacturer |
| Model | Vehicle Model |
| Trim | Vehicle Trim Level |
| SubModel | Vehicle Submodel |
| Color | Vehicle Color |
| Transmission | What type the transmission of the car Auto or Manual |
| WheelTypeID | The type id of the vehicle wheel |
| WheelType | The vehicle wheel type description (Alloy, Covers) |
| VehOdo | The vehicles odometer reading |
| Nationality | The Manufacturer's country |
| Size | The size category of the vehicle (Compact, SUV, etc.) |
| TopThreeAmericanName | Identifies if the manufacturer is one of the top three American manufacturers |
| MMRAcquisitionAuctionAveragePrice | Acquisition price for this vehicle in average condition |
| MMRAcquisitionAuctionCleanPrice | Acquisition price for this vehicle in the above Average condition |
| MMRAcquisitionRetailAveragePrice | Acquisition price for this vehicle in the retail market in average condition at time of purchase |
| MMRAcquisitonRetailCleanPrice | Acquisition price for this vehicle in the retail market in above average condition at time of purchase |
| MMRCurrentAuctionAveragePrice | Acquisition price for this vehicle in average condition as of current day |
| MMRCurrentAuctionCleanPrice | Acquisition price for this vehicle in the above condition as of current day |
| MMRCurrentRetailAveragePrice | Acquisition price for this vehicle in the retail market in average condition as of current day |
| MMRCurrentRetailCleanPrice | Acquisition price for this vehicle in the retail market in above average condition as of current day |
| PRIMEUNIT | Identifies if the vehicle would have a higher demand |
| AcquisitionType | Identifies how the vehicle was aquired (Auction buy, trade in, etc) |
| AUCGUART | The level guarntee provided by auction for the vehicle |
| KickDate | Date the vehicle was kicked back to the auction |
| BYRNO | Unique number assigned to the buyer that purchased the vehicle |
| VNZIP | Zipcode where the car was purchased |
| VNST | State where the the car was purchased |
| VehBCost | Acquisition cost paid for the vehicle at time of purchase |
| IsOnlineSale | If the vehicle was sold online |
| WarrantyCost | Warranty price (term=36month and millage=36K) |

## Key Observations in data

* Redundant data: VehYear and VehAge mean the same thing
* Poor Quality of variables: PRIMEUNIT only 4.6% records were no
* Class Imbalance: 87.7 % Good Buys, only 13.3 % Bad buys
* There were no Manual transmission vehicles which were bad buys
* Only 0.11% records with RED category in AUCGUART

## Data Preprocessing

* Removed redundant features
* Removed features with more than 95% missing values
* Handles Null/Missing values
  + Continuous data: took average
  + Discrete data: created new category NULL
* Normalized all the continuous values in range [0, 1]
* We were left with 22 features to work with

# RESULTS AND APPROACH

## 2.1 Class Imbalance

There was class imbalance in dataset. It was addresses by creating datasets (shown in [2.3](#_2.3__)) using the below techniques:

1. Oversampling with replacements
2. SMOTE (Synthetic Minority Oversampling Technique)
3. Undersampling of majority label

## 2.2 Feature Engineering

We tried multiple ways to get the best features for the predictions:

* + - * 1. An expert we met from Auction Direct recommended the following best features:
* VehOdo
* VehicleAge
* MMRCurrentAuctionCleanPrice
* MMRCurrentRetailAveragePrice
* Transmission
  + - * 1. We tried Chi Sqaure Ranks which gave us the following results

Unbalanced Data : Best Score for All 22 features

Balanced Data : Best Score for 17 features

* + - * 1. We tried Recursive Feature Elimination which gave us best features to be:
* MMRAcquisitionAuctionAveragePrice
* MMRAcquisitionretailCleanPrice
* MMRCurrentAuctionCleanPrice
* MMRCurrentAuctionAveragePrice
* WarrantyCost

Recursive Feature Elimination did not consider discrete features. However it was correct with respect to MMRCurrentAuctionCleanPrice and MMRCurrentAuctionAveragePrice as they indeed were important features.

## 2.3 Dataset Samples

|  |  |
| --- | --- |
| ***Dataset1*** | UnBalanced Data |
| ***Dataset2*** | Balanced Data by Oversampling |
| ***Dataset3*** | Balanced Data by SMOTE |
| ***Dataset4*** | Balanced Data by Undersampling |
| ***Dataset5*** | Unbalanced Data; \*Selected features |
| ***Dataset6*** | Balanced Data by Oversampling; \*Selected features |
| \*Auction, VehicleAge, Make, Transmission, WheelType, VehOdo, Nationality, Size, TopThreeAmericanNames, MMRAcquisitionAuctionCleanPrice, MMRAcquisitionRetailAveragePrice | |

## 2.4 Classifiers Tried

|  |  |  |
| --- | --- | --- |
| **J48 Pruned** | **J48-Unpruned** | **Logitic regression** |
| **Adaboost** | **LogitBoost** | **Bagging** |
| **Random Forest** | **Naïve Bayes** | **SVM with c = 1, 0.1 and 0.001** |
| **1,3 and 5-NN** | **Ensemble (Average Vote)** |  |

## 2.5 Evaluation Criteria

* For Balanced data
  + Accuracy is a good measure
* For Unbalanced data
  + F measure and AUC ROC Score is a good measure

## 2.6 Results

Figure 1

Figure 2

Figure 3

Figure 4

Figure 5

Figure 6

Figure 7

Figure 8

Figure 9

Figure 10

Figure 11

Figure 12

Figure 13

Figure 14

Figure 15

Figure 16

Figure 17

Figure 18

# DETAILED ANALYSIS

## 3.1 Accuracy Balanced v/s Unbalanced Data

Accuracy is usually considered a bad measure for unbalanced data thus it was expected to be lesser than balanced data. Clearly, this is not the case, we then hypothesize that classifiers are overfitting for balanced data.

Figure 19

## 3.2 Bagging

The hypothesis from [3.1](#_3.1_Accuracy_Balanced) was made sure by looking at the results for bagging.

Figure 20

We can see that the accuracy of Balanced data is at par with that of Unbalanced data. We know that bagging is known to reduce variance thus reducing the tendency to overfit.

## 3.3 k Nearest Neighbor

k-NN was run for the values of 1, 3 and 5 on the various datasets.

Figure 21

Figure 22

Figure 23

***For Unbalanced data:***

If we look at F Measure and ROC Area they tend to improve as we increase the k, this clearly shows that **KNN is overfitting for k = 1.**

***For Balanced data:***

If we look at accuracy for Balanced dataset, we would have expected accuracy to increase as value of k increases. However, **this does not happen**. We hypothesized that kNN works best for k = 1 for Balanced data because of duplicate data points. These points are generated by oversampling with replacement. Also when we use SMOTE, data synthetically generated is **close to the original points** in the sample. Thus we get these results.

## 3.4 All Classifiers v/s each other

Figure 24

Figure 25

## 3.5 Random Forest and Ensemble

Figure 26

Figure 27

Figure 28

If we look at Accuracy, F Measure and ROC for Balanced data both Ensemble and Random Forest do very well.

They can be considered to be best algorithms for synthetically generated balanced data

However, if we look at unbalanced data F Measure and ROC Scores of these two are **significantly** **less** as compared to other classifiers like **Decision trees and Logistic regression**.

We observed that though **the number of False Negatives are being reduced but in effect to this, there is a drastic increase in number of False Positives.**

## 3.6 Naïve Bayes

If we look at Naïve Bayes from figure we can clearly see that it performs really badly. This happens because the fundamental assumption of Naïve Bayes that features should be independent of each other is violated. Features in this dataset have clear dependence. For example **Odometer reading and Age of Vehicle are related**. Price of Vehicle, Vehicle Age, and Warranty Cost are related.

## 3.7 Data Balancing Techniques

Oversampling with replacement worked better than SMOTE and even outperformed Undersampling.

This shows that SMOTE cannot always give better results.

## 3.7 SVM

As it can be seen in the figure, changing values of c does not have a significant affect all three evaluation measures. Moreover, the performance of SVM is not good compared to other classifiers on these datasets. Thus, SVM does not work well.

## 3.8 Decision Trees

The comparison clearly shows that the decision trees perform very well and are suitable classifiers for this dataset. First split happens on Wheel type and next split on Auction implying that these are the features with relatively highest information gain.

## 3.9 Logistic Regression

Logistic regressions works very well and is the preferred classifier for this dataset as for unbalanced data it has highest F Measure and AUC ROC Score. It also have relatively better accuracy.

# CONCLUSION

# REFERENCES

* [SMOTE: Synthetic Minority Over-sampling Technique](https://www.cs.cmu.edu/afs/cs/project/jair/pub/volume16/chawla02a-html/chawla2002.html) - *Chawla, Bowyer, Hall and Kegelmeyer*

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |