INDIANA UNIVERISTY BLOOMINGTON

Final Project: Don't get KICKED

I 526 Applied Machine Learning

PROJECT REPORT

Submitted by

Nihar Khetan
Masters Candidate in Computer Science
School of Informatics and Computing
Indiana University
nkhetan@indiana.edu

Ghanshyam Malu
Masters Candidate in Computer Science
School of Informatics and Computing
Indiana University
gmalu@indiana.edu

Xiao Liang
Masters Candidate in Health Informatics.
School of Informatics and Computing
Indiana university
liang25@indiana.edu

Under the Guidance of:

Professor Sriraam Natarajan
Asst. Professor, School of Informatics and Computing
Indiana University

ACKNOWLEDGMENT

Our project was the result of the encouragement of many people who helped in shaping it and provide feedback, direction valuable support. It is with hearty gratitude that we acknowledge their contributions to our project.

We would like to first thank Professor **Sriraam Natarajan,** Asst. Professor, School of Informatics and Computing, Indiana University, for the constant help, support and feedback extended towards us during the course of the project.

We are also grateful to the Associate Instructor, **Devendra Dhami,** Phd. Candidate, School of Informatics and Computing, Indiana University for his tips to get better results.

Ghanshyam Malu

Nihar Khetan

Xiao Liang

Table of Contents

1. DATASET DESCRIPTION

1.1 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.**

1.2 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.

1.3 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.

1.4 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)

1.5 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

1.6 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

2. RESULTS AND APPROACH

2.1 Class Imbalance

There was class imbalance in dataset. It was addresses by creating datasets (shown in 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
2. 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
3. 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 Test Set Generation

We generated test set by sampling 30 percent data from the complete dataset randomly. We made sure that test set is consistent for all Datasets mentioned in section 2.3.

2.6 Evaluation Criteria

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

2.7 Results

We did experiments for all datasets defined in section 2.3 and recorded Accuracy, F measure and AUC ROC Score for all Classifiers mentioned in section 2.4.

F measure represents Precision and Recall and AUR ROC Score Sensitivity and Specivity.

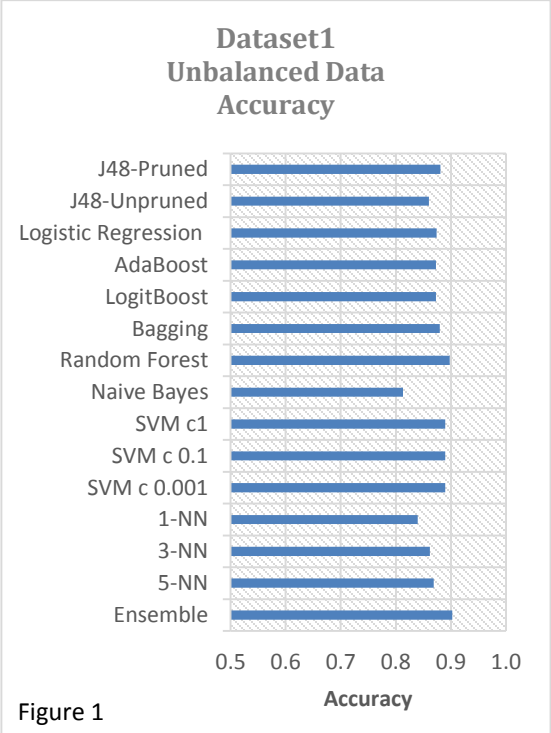


Figure 1

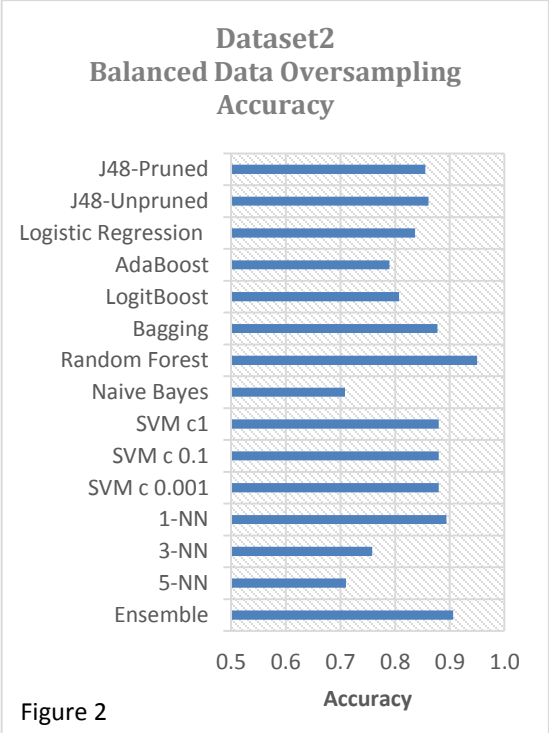


Figure 2

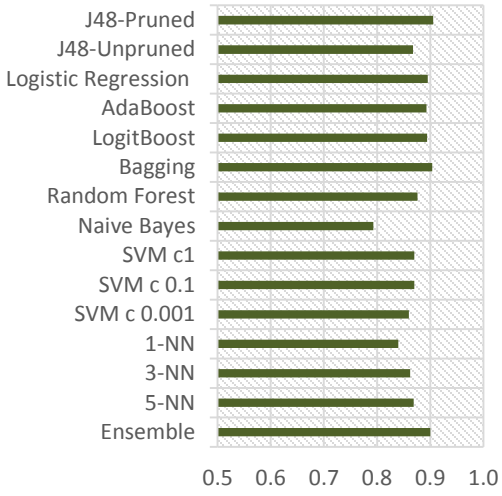


Figure 3

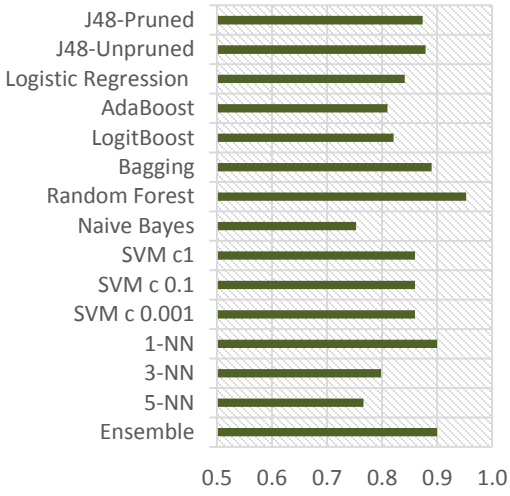


Figure 4

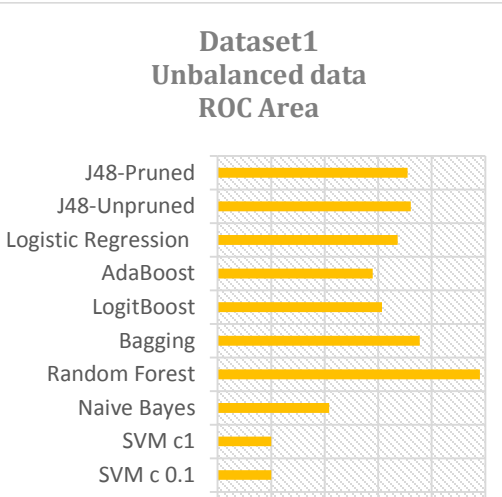


Figure 5

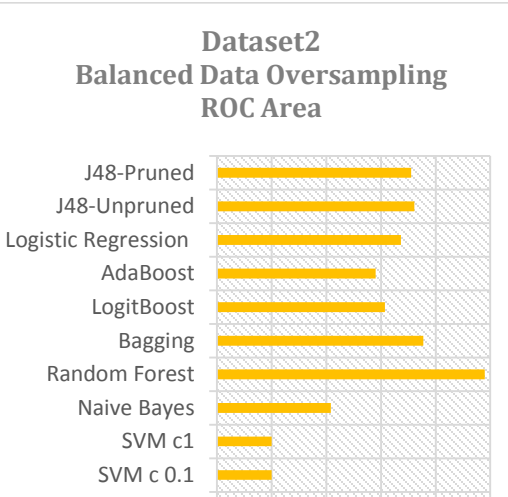


Figure 6

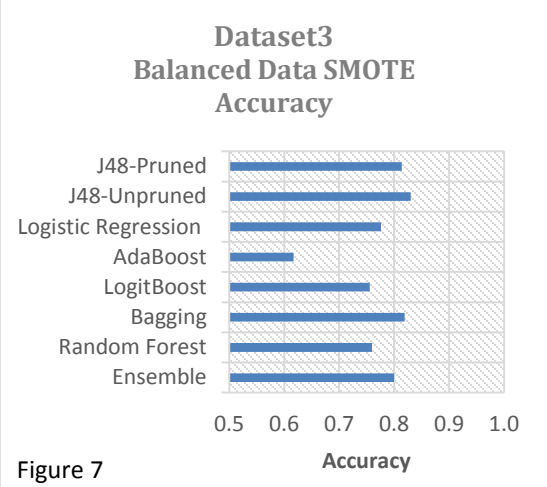


Figure 7

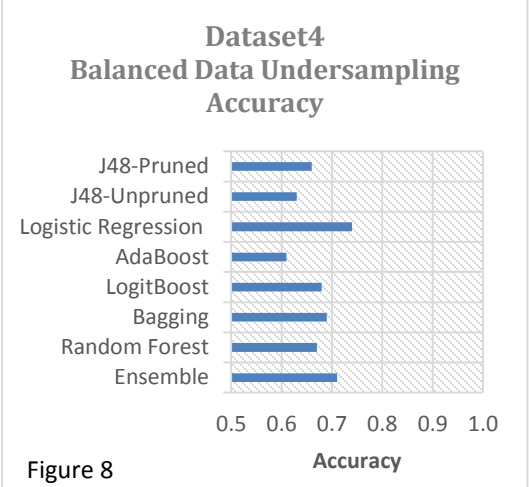


Figure 8

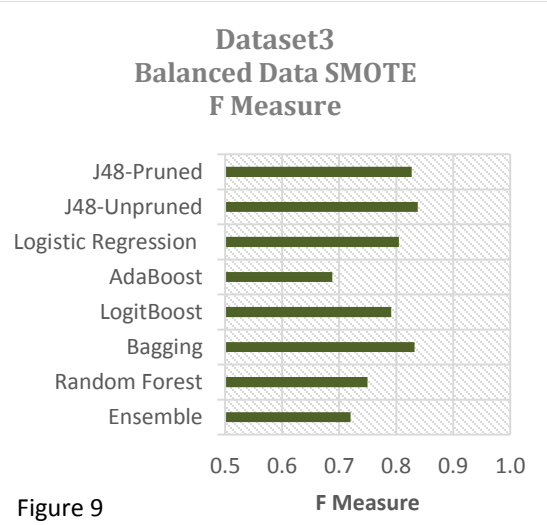


Figure 9

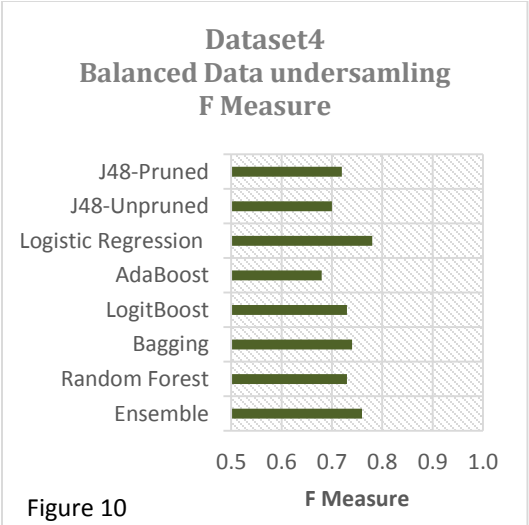


Figure 10

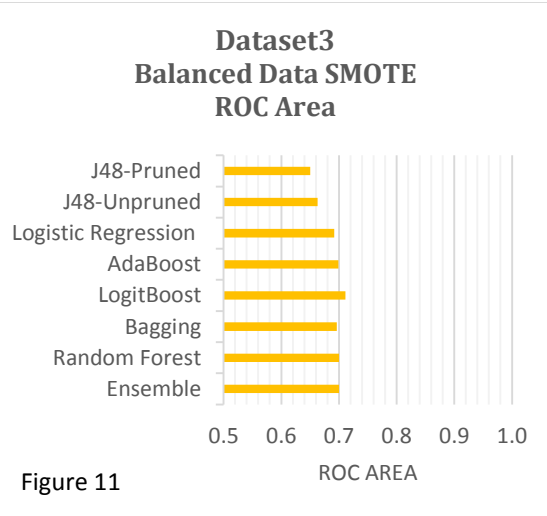


Figure 11

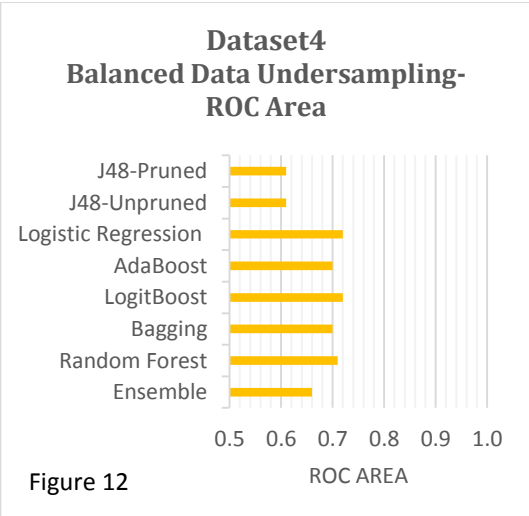


Figure 12

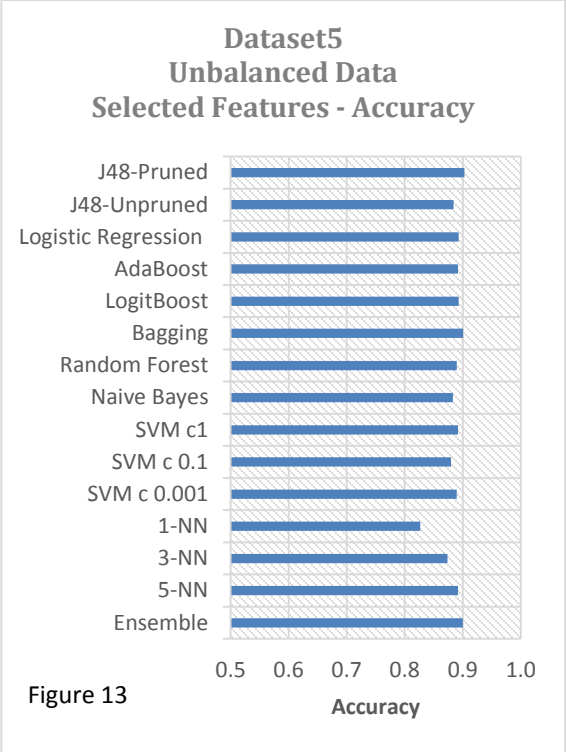


Figure 13

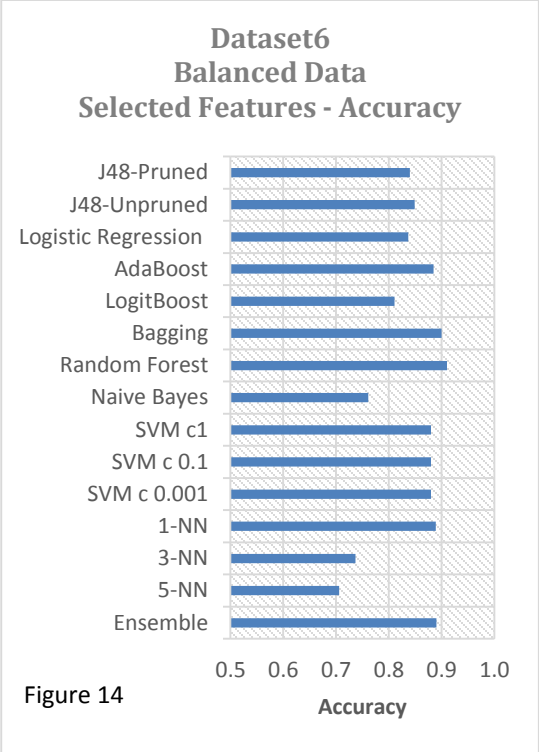


Figure 14

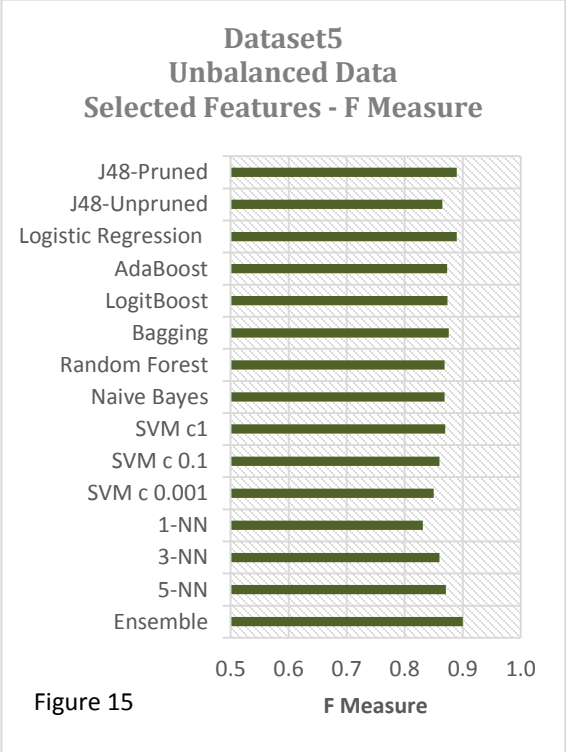


Figure 15

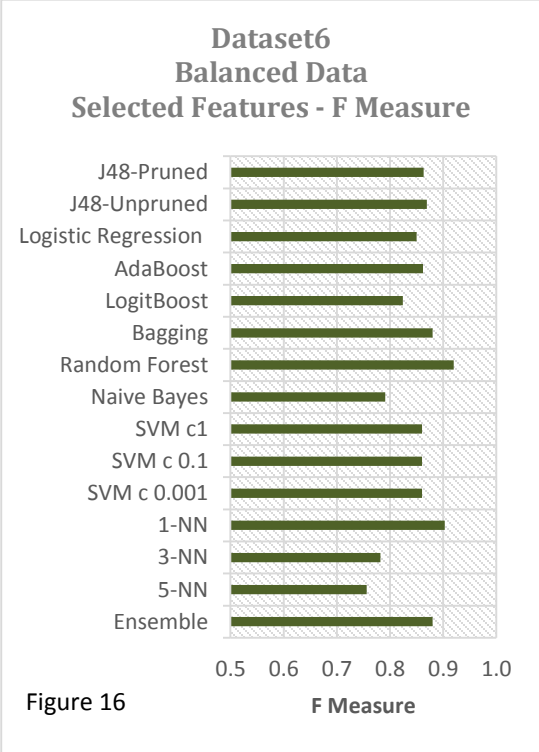


Figure 16

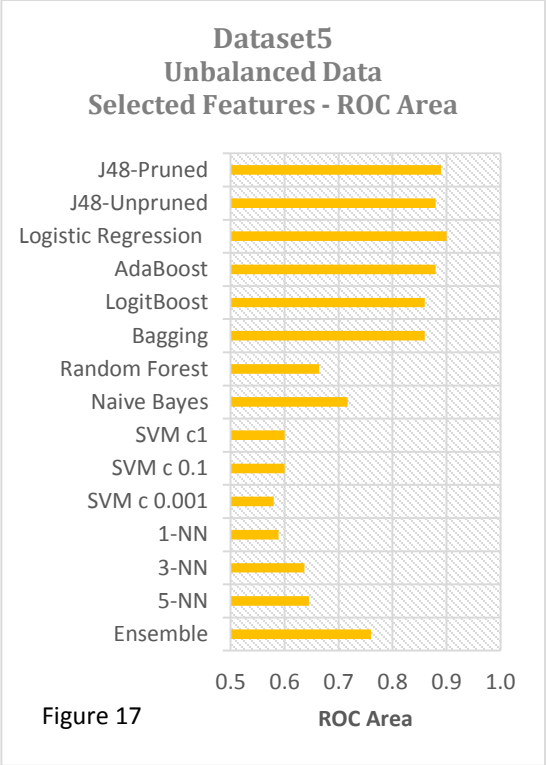


Figure 17

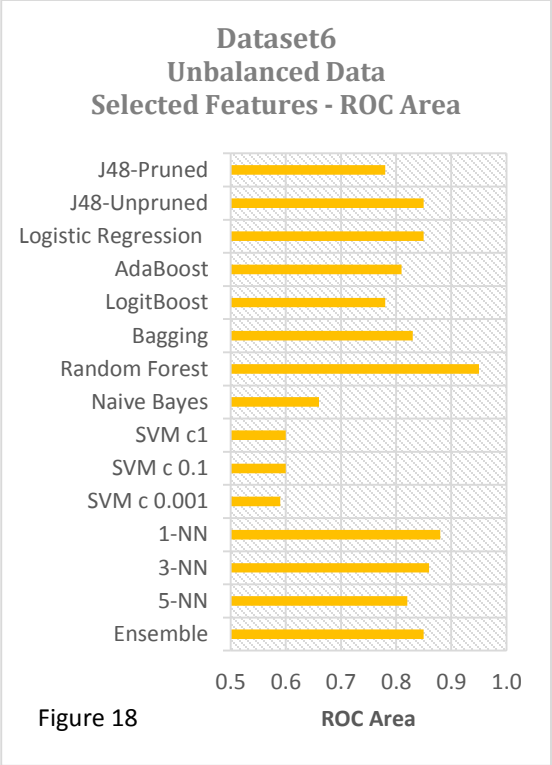


Figure 18

3. 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. This can be clearly seen in Figure 19 below.

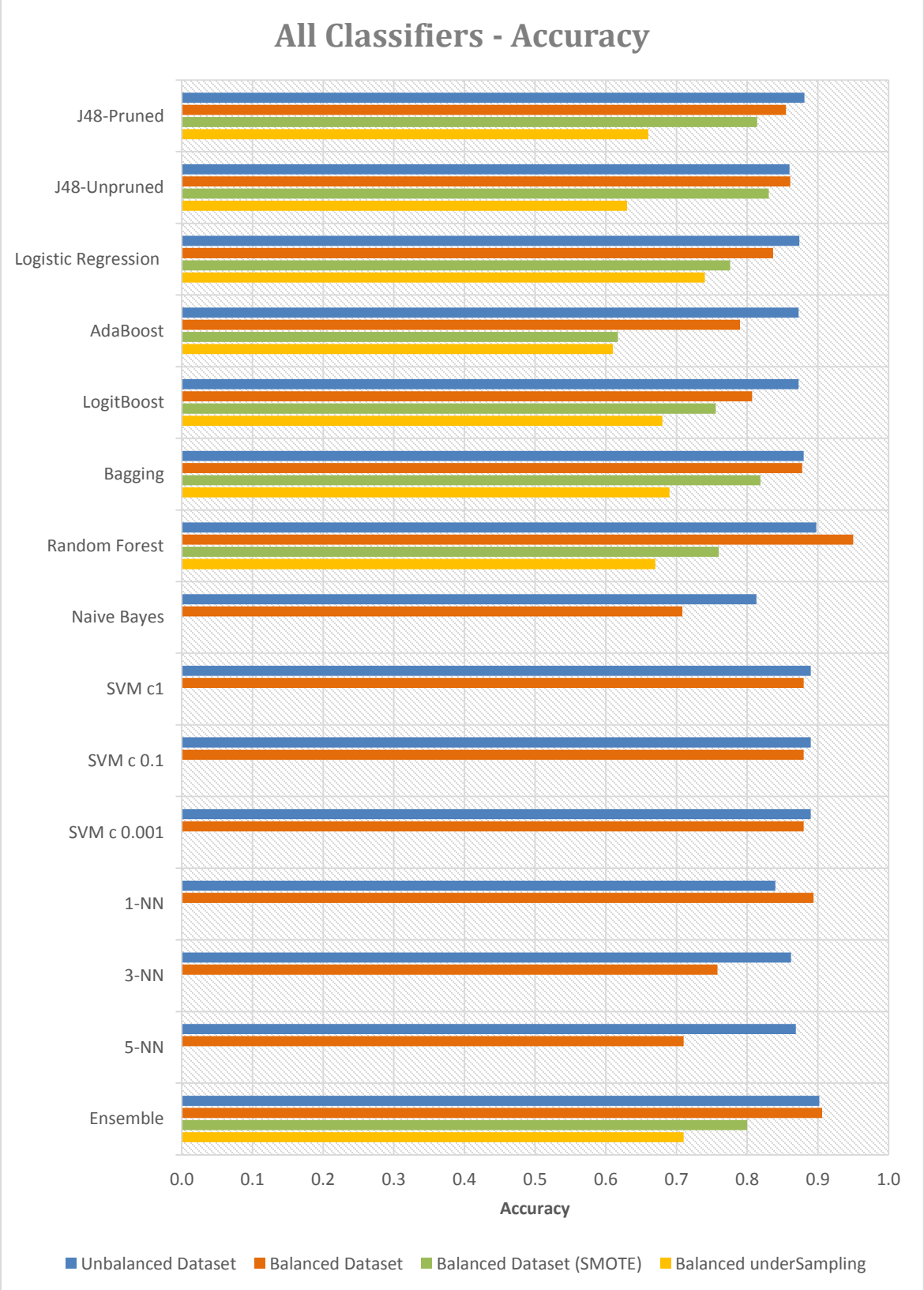

All Classifiers - Accuracy
J48-Pruned
J48-Unpruned
Logistic Regression
AdaBoost
LogitBoost
Bagging
Random Forest
Naive Bayes
SVM c1
SVM c 0.1
SVM c 0.001
1-NN
3-NN
5-NN
Ensemble
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0
Accuracy
Unbalanced Dataset
Balanced Dataset
Balanced Dataset (SMOTE)
Balanced underSampling

Figure 19

3.2 Bagging

The hypothesis from 3.1 was confirmed by looking at the results for bagging.

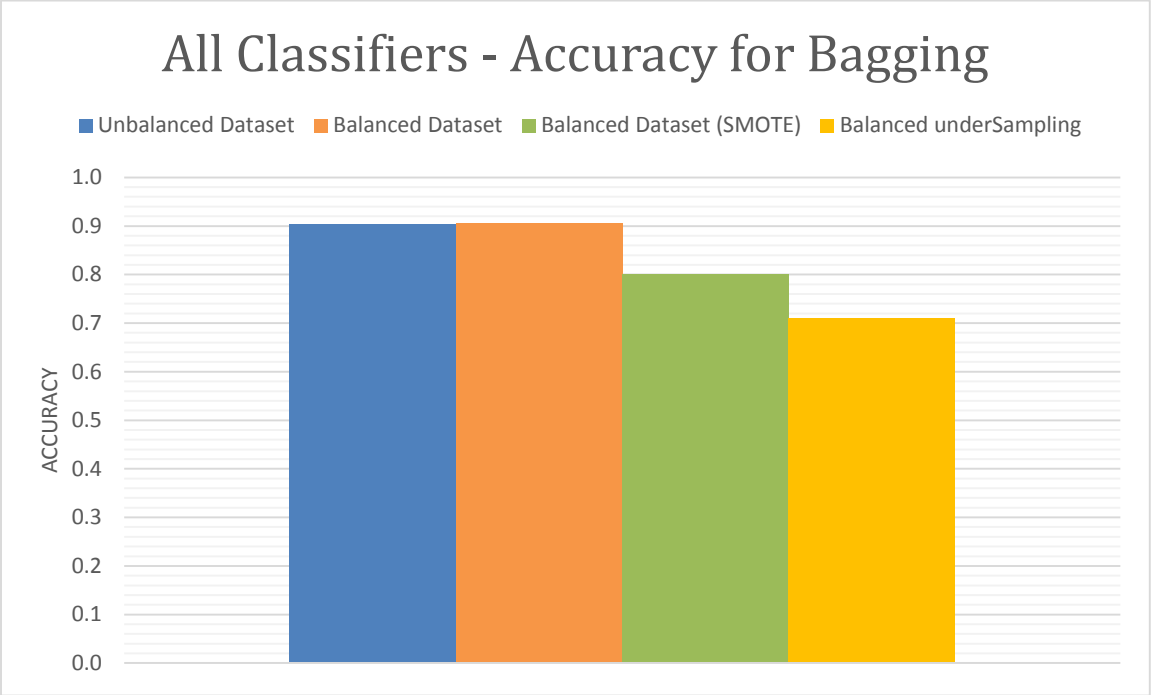


Figure 20

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. Results can be seen in Figure 21, 22 and 23.

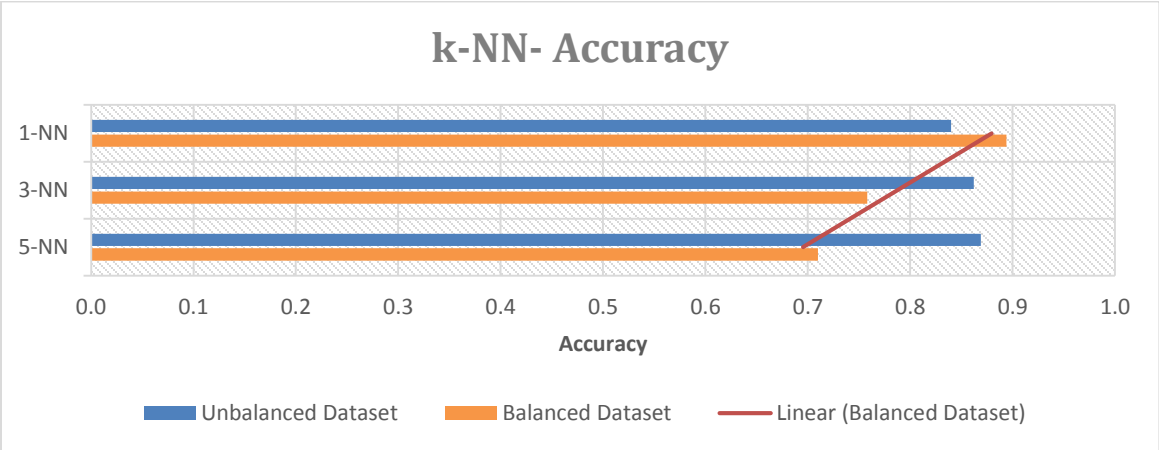


Figure 21

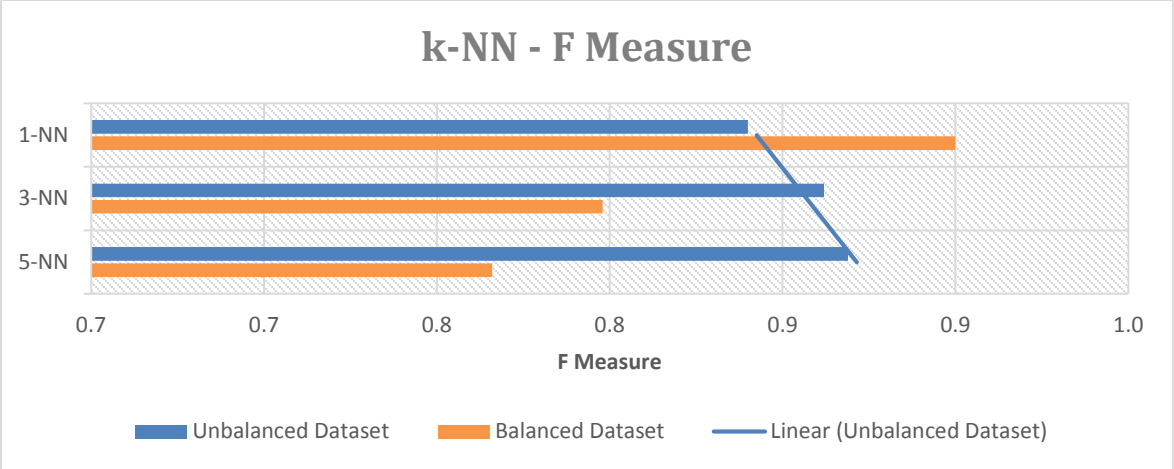


Figure 22

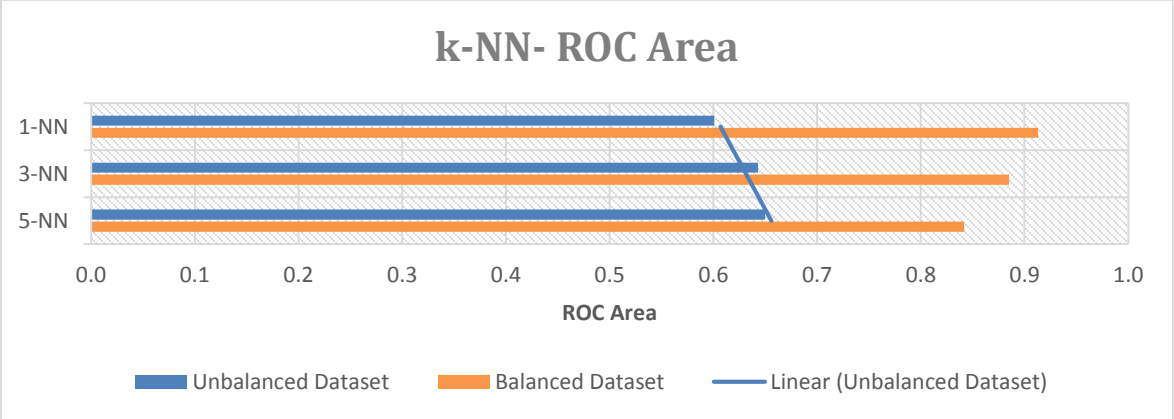


Figure 23

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

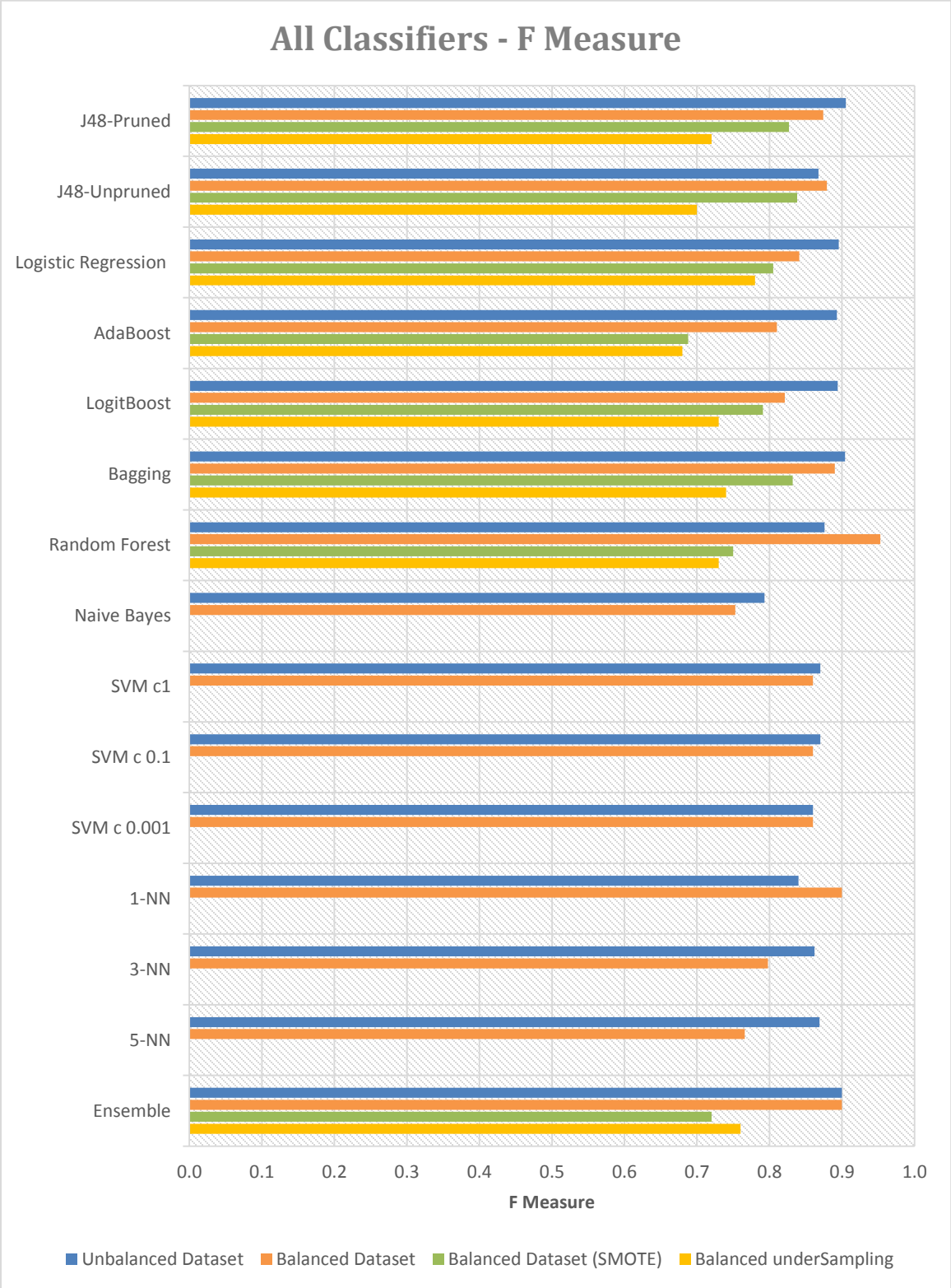


Figure 24

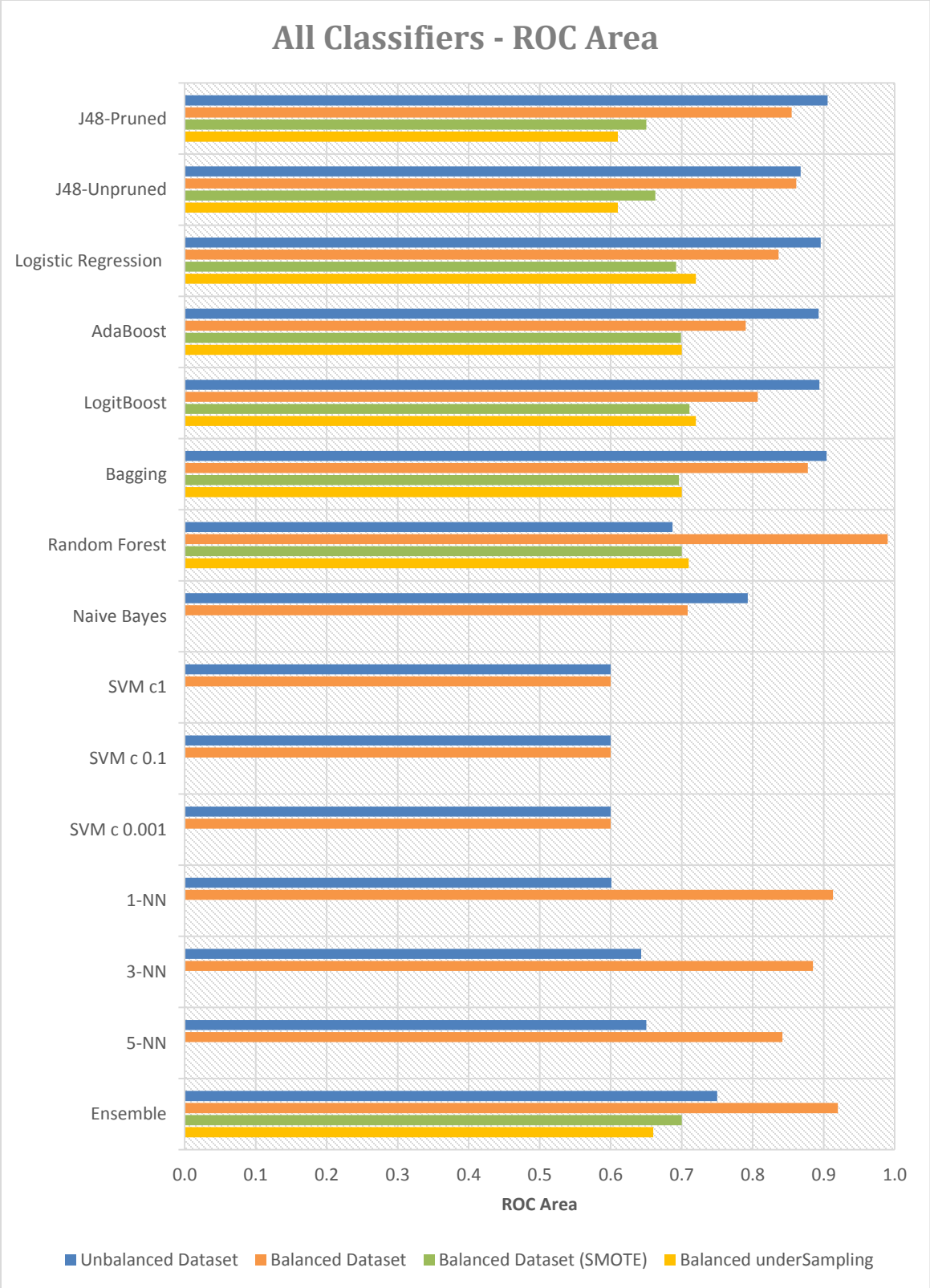

All Classifiers - ROC Area
J48-Pruned
J48-Unpruned
Logistic Regression
AdaBoost
LogitBoost
Bagging
Random Forest
Naive Bayes
SVM c1
SVM c 0.1
SVM c 0.001
1-NN
3-NN
5-NN
Ensemble
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0
ROC Area
Unbalanced Dataset
Balanced Dataset
Balanced Dataset (SMOTE)
Balanced underSampling

Figure 25

3.5 Random Forest and Ensemble

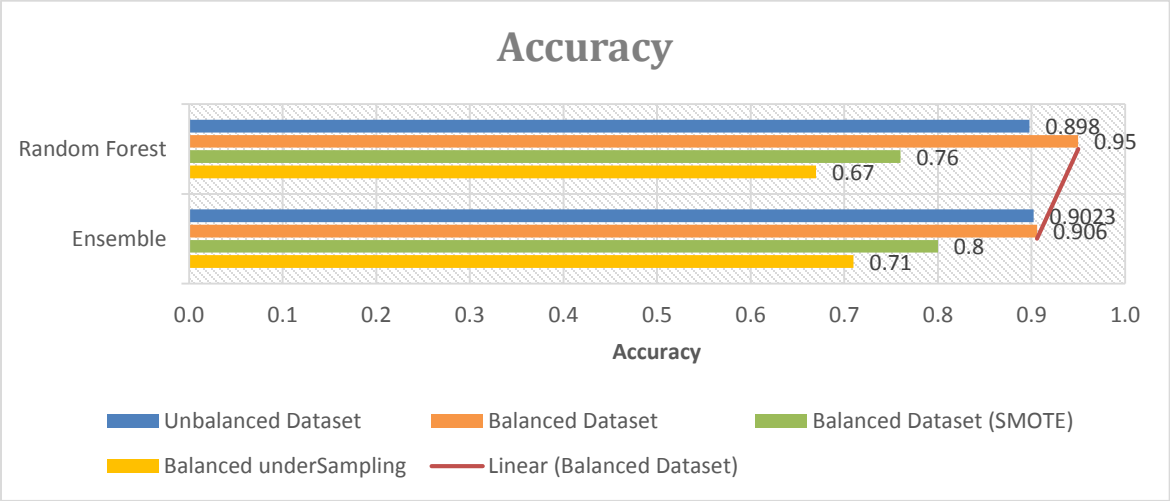


Figure 26

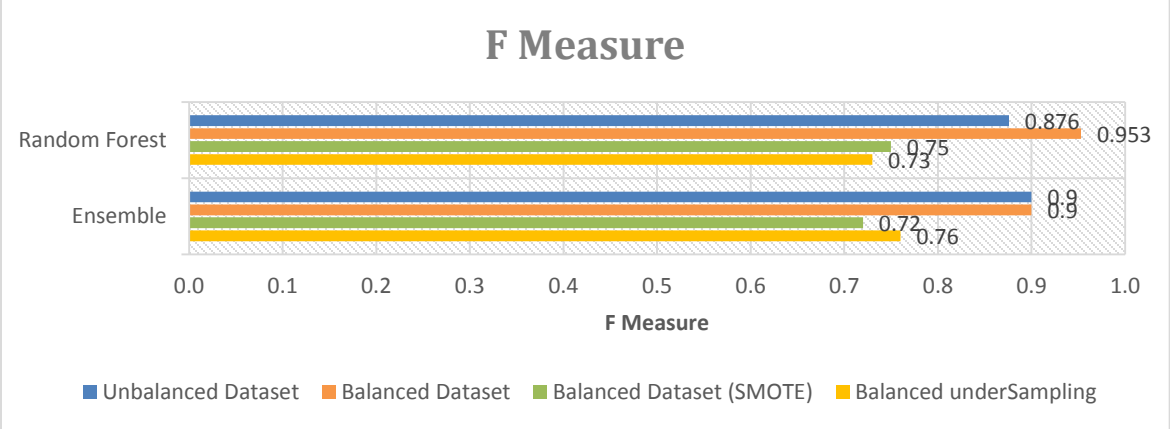


Figure 27

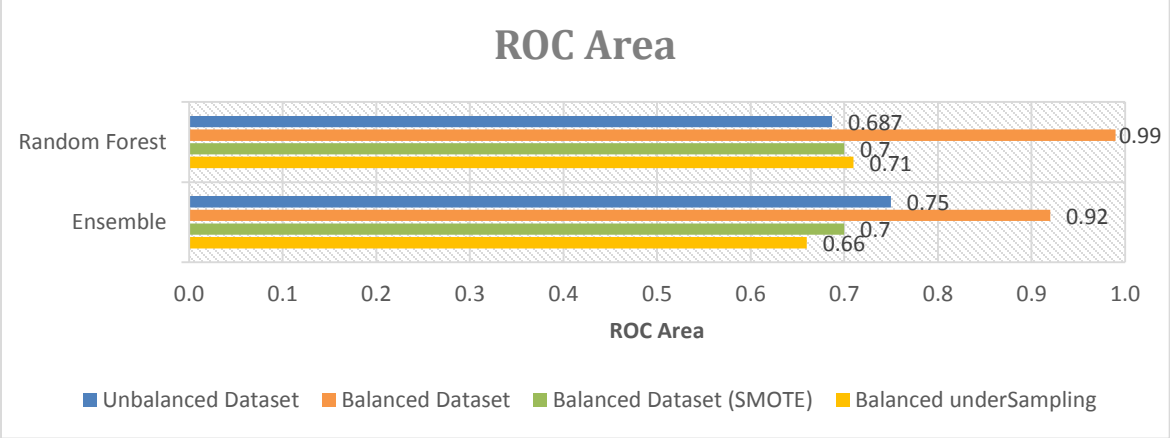


Figure 28

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**. These comparisons can clearly be seen in Figure 24 and Figure 25

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 19, Figure 24 and Figure 25 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 can be clearly seen in Figure 19, Figure 24 and Figure 25.

This shows that SMOTE cannot always give better results.

3.7 SVM

As it can be seen in the Figure 19, Figure 24 and Figure 25, 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 through Figure 19, Figure 24 and Figure 25 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.

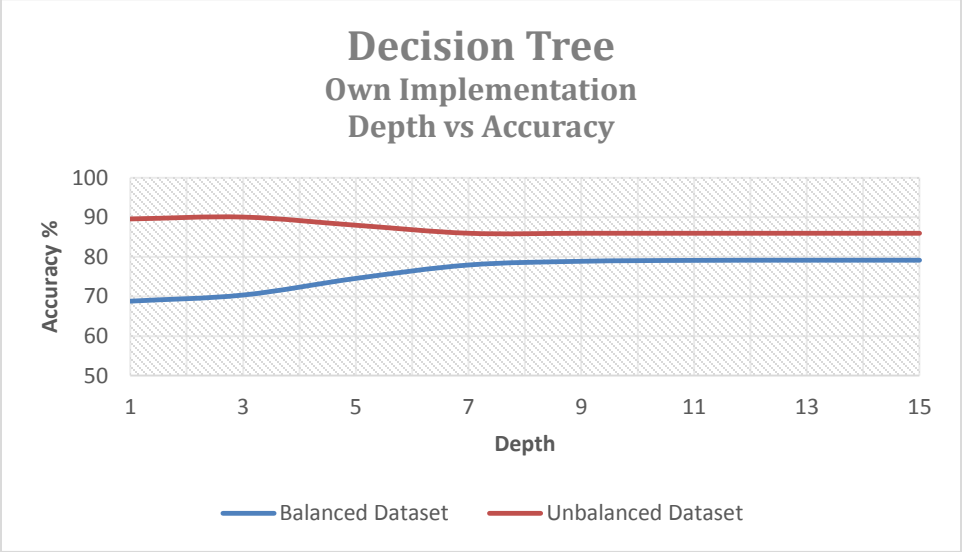
3.9 Logistic Regression

Logistic regressions works very well as it can be seen in Figure 19, Figure 24 and Figure 25. It 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**.

4. OWN IMPLEMENTATIONS

4.1 Decision Trees

Our implementation of decision tree algorithm was run on the balanced and unbalanced datasets on various depths starting from 1. **While the unbalanced dataset provided better results initially, it decreased gradually until its saturation**. On the other hand, the performance of balanced dataset increased and attained a saturation.



4.2 Logistic Regression

Logistic Regression was run on the normalized balanced and unbalanced datasets, and the performance is comparable to weka version where Unbalanced dataset again led to better results compared to the balanced one.

5. REFERENCES

- SMOTE: Synthetic Minority Over-sampling Technique - *Chawla, Bowyer, Hall and Kegelmeyer*
- Class slides – Fall 2015 - *Sriraam Natarajan*
- Dataset: https://www.kaggle.com/c/DontGetKicked