Mobile Price Classification

Team 1
Antonio Moral, Bo Li, Manushi Patel
Scott McCoy, Yixuan Wang, Yulong Gong

About Us

Antonio Moral

Manushi Patel

Bo Li

Scott McCoy

Yixuan Wang

Yulong Gong

Business Problem

- Utilize classification methods to predict the price range of a mobile phone based on the technical features.
- Solving the business problem is a journey of exploration.
  - Multiple models are build.

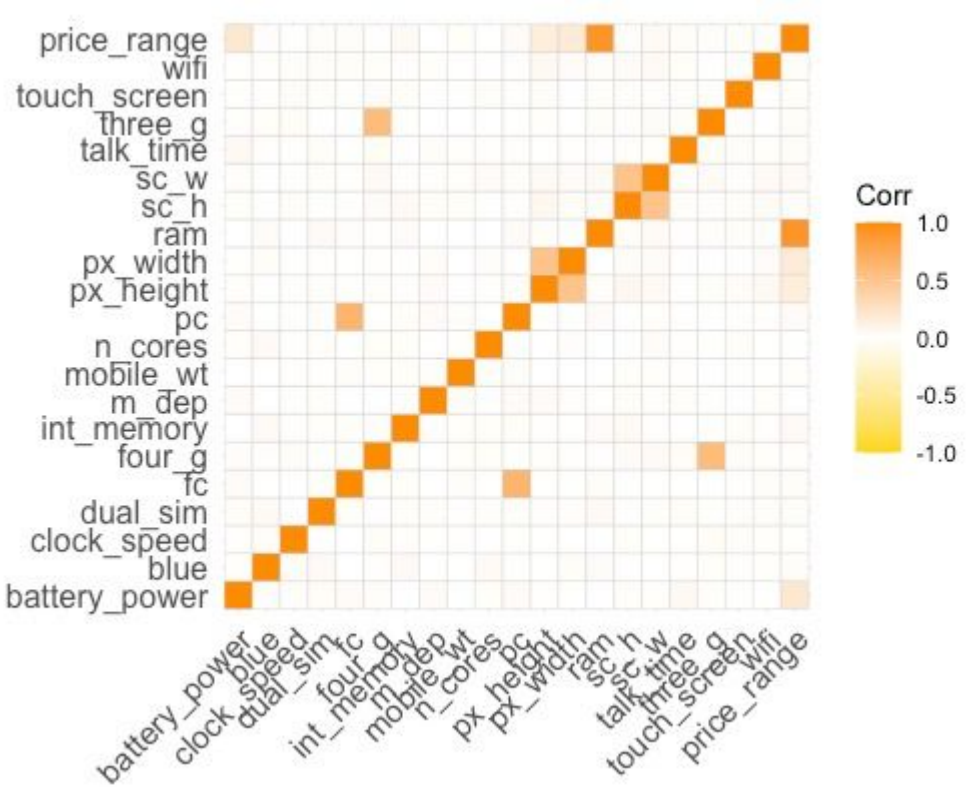
- Random Forest
- Decision Tree
- Layered BL Regressions
- Multiple BL
- Binary Logistic(BL)
- Linear neural network model

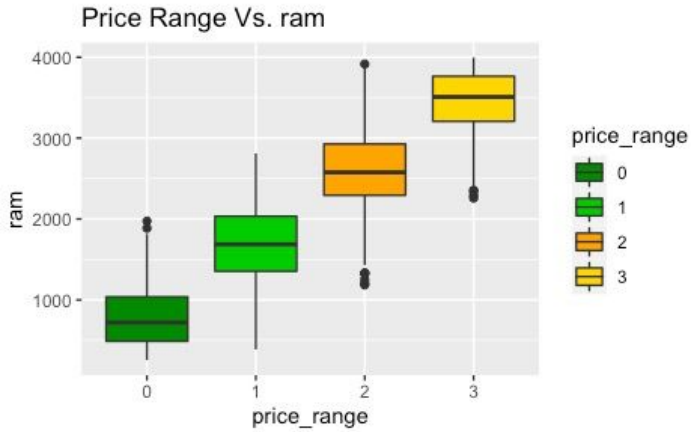
Target Variable	Interpretation
0	Low Cost
1	Medium Cost
2	High Cost
3	Very High Cost

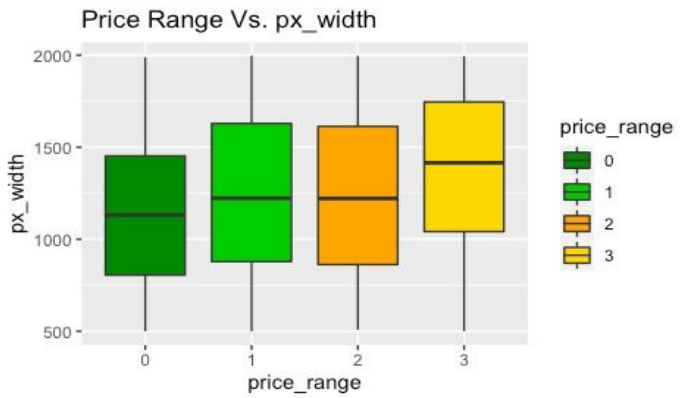
Our Dataset (EDA)

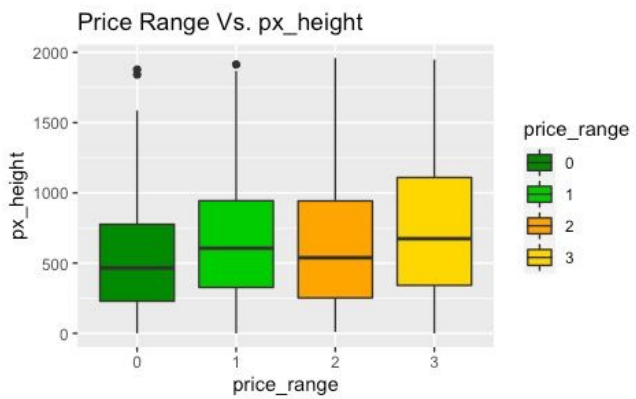
Mobile Price Classification Dataset

- https://www.kaggle.com/iabhishekofficial/mobile-price-classification
- 2000 rows, 21 variables
- Data type: all variables are numeric
- No missing value









Logistic Regression Classification Methods

Multinomial Classification Neural Net:

```
> model_multi
Call:
multinom(formula = price_range ~ ., data = mo_trn)
```

```
> test_accuracy
[1] 0.9716667
> cm
          predicted
observed   0   1   2   3
       0 147   1   0   0
       1   2 147   2   0
       2   0   5 147   6
       3   0   0   1 142
> scores
  precision    recall        f1
0 0.9865772 0.9932432 0.9898990
1 0.9607843 0.9735099 0.9671053
2 0.9800000 0.9303797 0.9545455
3 0.9594595 0.9930070 0.9759450
```

Binary Logistic Regression:

```
log_model_binary <- glm(price_binary ~ .,
family=binomial(link='logit'), data=trainSet)
```

```
> test_accuracy
[1] 0.9883333
> cm
          predicted
observed   0   1
       0 296   4
       1   3 297
> scores
  precision    recall        f1
0 0.9899666 0.9866667 0.9883139
1 0.9867110 0.9900000 0.9883527
```

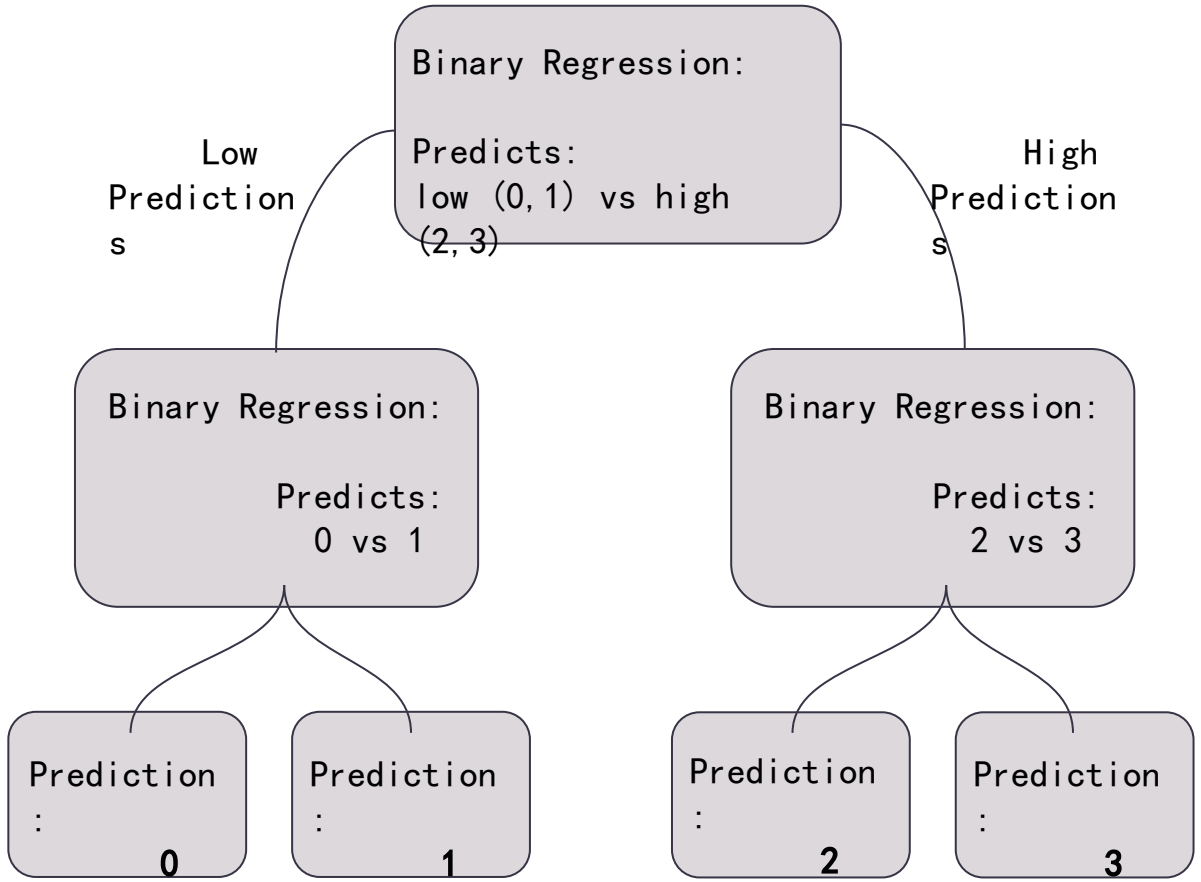
Multiple Binary Logistic Regressions:

```
glm.fit0 <- glm(p0 ~ ., data = X_train0, family = binomial)
glm.fit1 <- glm(p1 ~ ., data = X_train1, family = binomial)
glm.fit2 <- glm(p2 ~ ., data = X_train2, family = binomial)
glm.fit3 <- glm(p3 ~ ., data = X_train3, family = binomial)
```

```
> test_accuracy
[1] 0.8433333
> cm
          predicted
observed   0   1   2   3
       0 148   0   0   0
       1   2 113  36   0
       2   0  45 104   9
       3   0   0   2 141
> scores
  precision    recall        f1
0 0.9866667 1.0000000 0.9932886
1 0.7151899 0.7483444 0.7313916
2 0.7323944 0.6582278 0.6933333
3 0.9400000 0.9860140 0.9624573
```

Logistic Regression Classification Methods

Layered Binary Logistic Regressions:




```
> binary1_accuracy
[1] 0.9883333
```

```
> test_accuracy_low
[1] 0.98
```

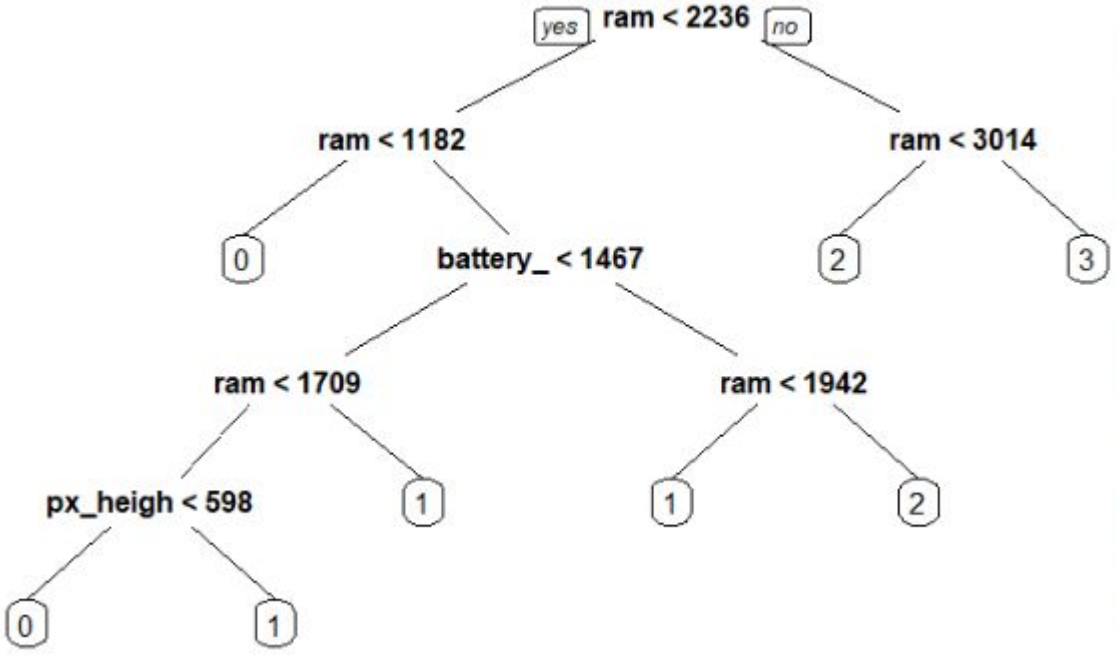
```
> test_accuracy_high
[1] 0.9533333
```

```
> test_accuracy
[1] 0.9666667
```

```
> cm
          predicted
observed   0   1   2   3
       0 148   0   0   0
       1   2 146   3   0
       2   0   4 145   9
       3   0   0   2 141
> scores
  precision    recall        f1
0 0.9866667 1.0000000 0.9932886
1 0.9733333 0.9668874 0.9700997
2 0.9666667 0.9177215 0.9415584
3 0.9400000 0.9860140 0.9624573
```

Tree Based Methods

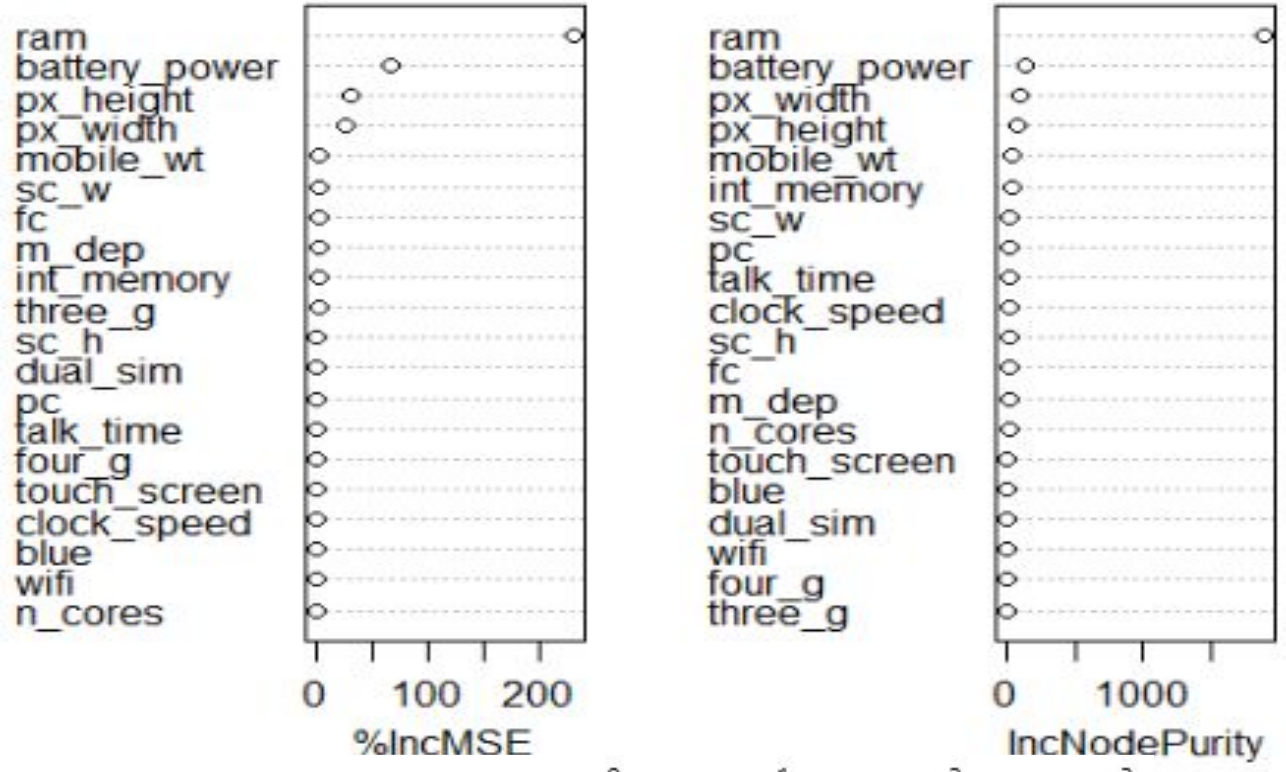
- Decision Tree predict accuracy: 77.5%



observed \ predicted	0	1	2	3
0	139	11	0	0
1	33	99	18	0
2	0	24	94	32
3	0	0	17	133

Accuracy
0.775

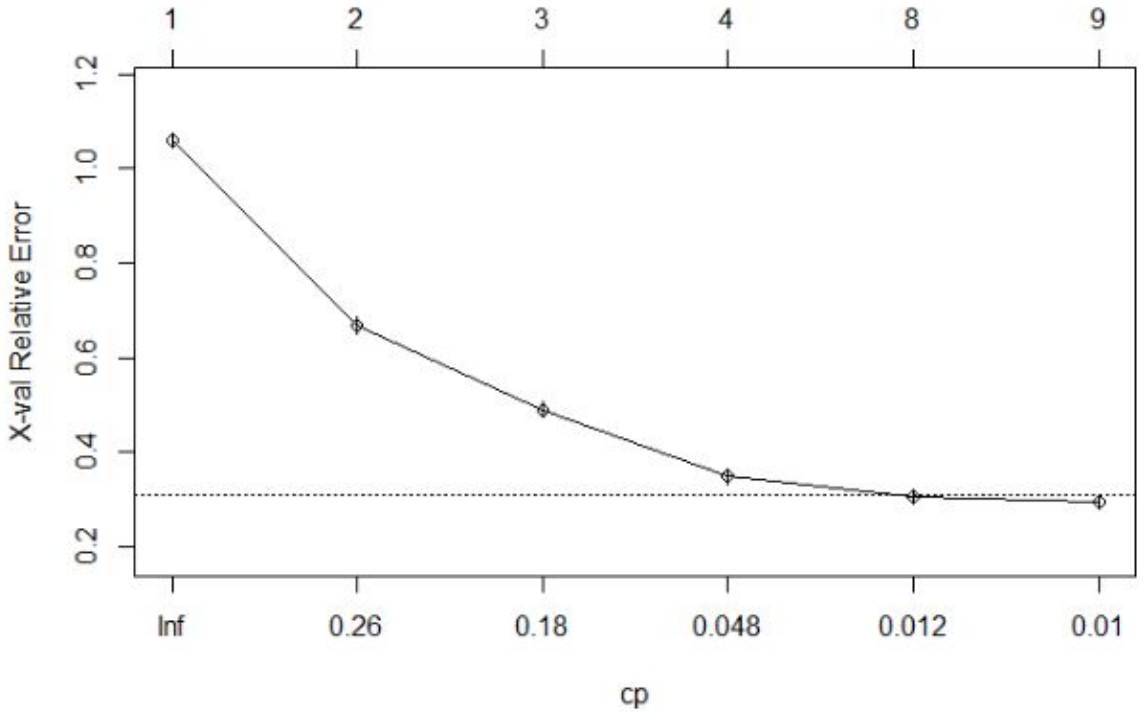
Feature Importance ScatterPlot

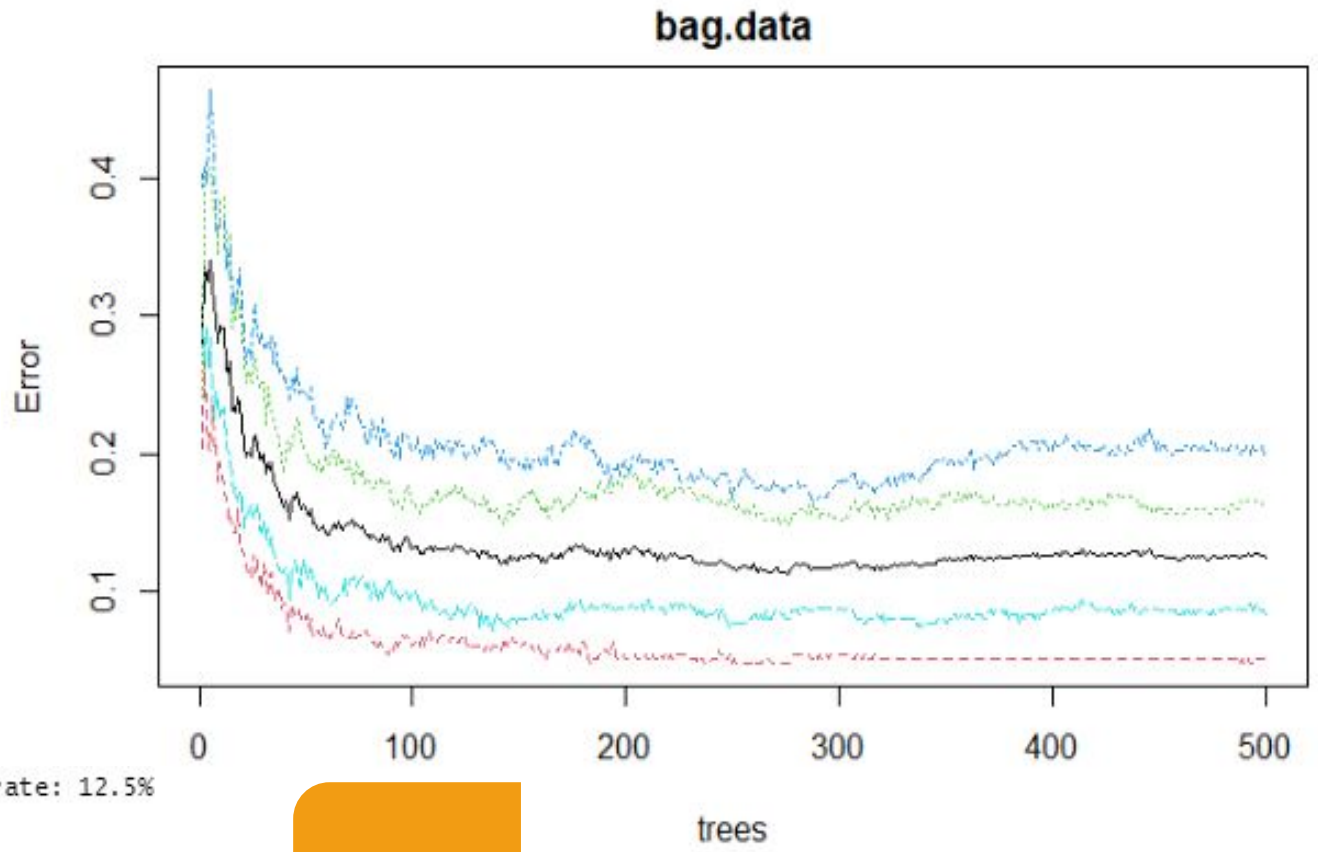


	0	1	2	3	MeanDecreaseAccuracy	MeanDecreaseGini
battery_power	20.315411783	27.2379966	26.4401697	21.5151653	39.7358388	80.225757
ram	102.087642597	66.7879289	66.1242760	88.9441201	97.3703194	497.722544

Tree Based Methods

- Decision Tree - performance improvement part 1





	CP	nsplit	rel error	xerror	xstd
1	0.333333	0	1.00000	1.06095	0.014367
2	0.198095	1	0.66667	0.66762	0.017817
3	0.160952	2	0.46857	0.48952	0.017177
4	0.014286	3	0.30762	0.34762	0.015645
5	0.010476	7	0.25048	0.30476	0.014964
6	0.010000	8	0.24000	0.29238	0.014744

OOB estimate of error rate: 12.5%

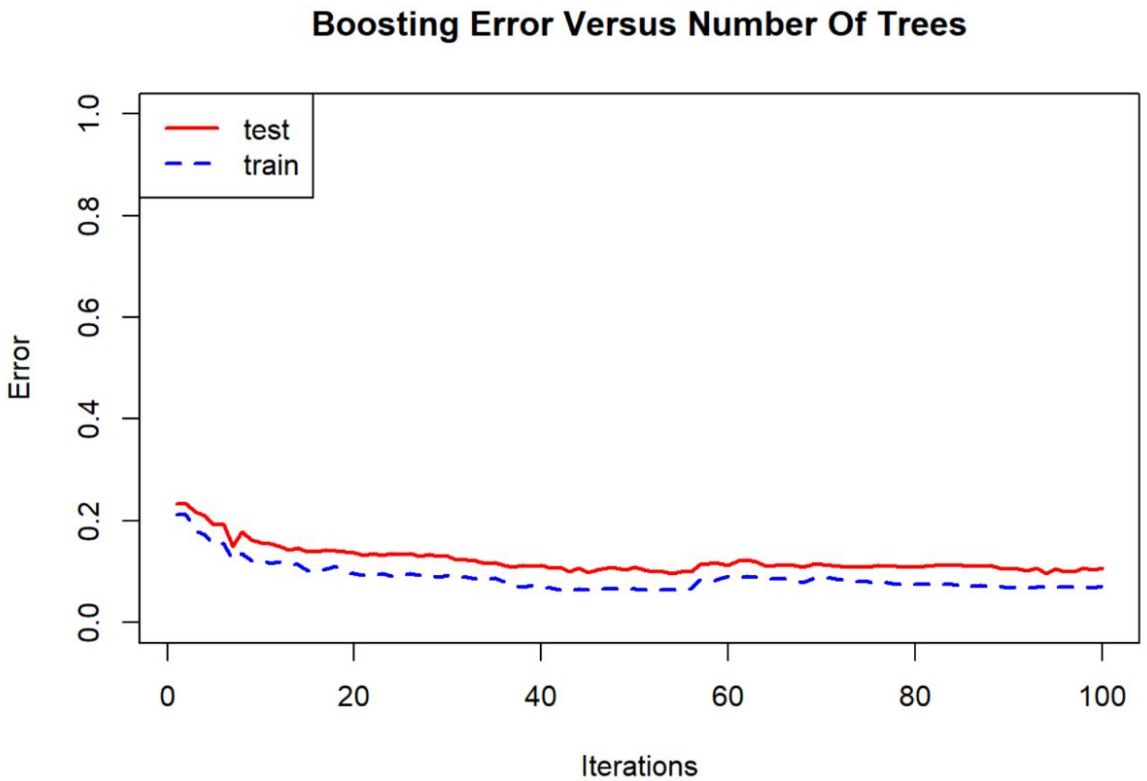
Confusion matrix:

	0	1	2	3	class.error
0	332	18	0	0	0.05142857
1	31	292	27	0	0.16571429
2	0	38	280	32	0.20000000
3	0	0	29	321	0.08285714

Accuracy
0.877

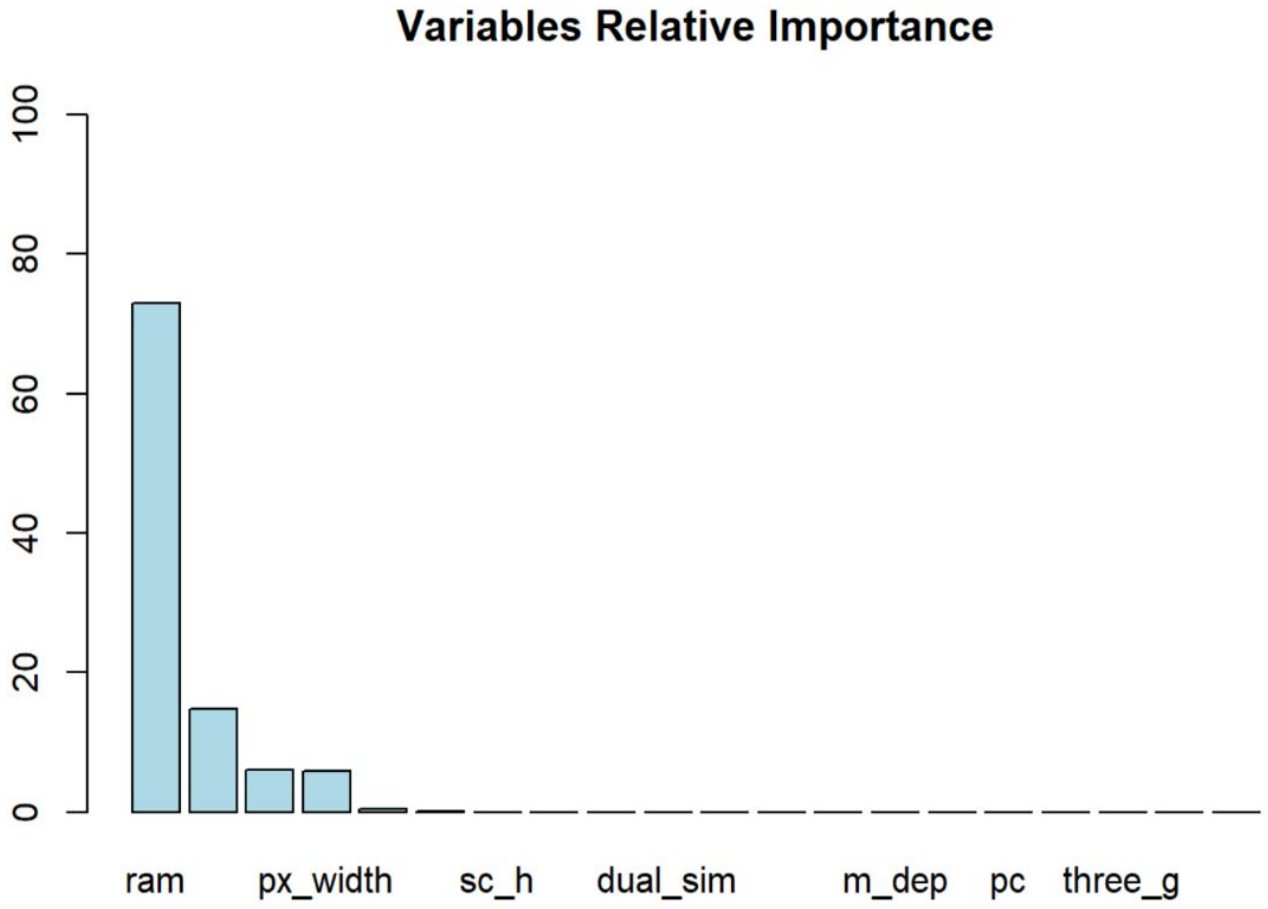
Tree Based Methods

- Decision Tree - performance improvement part 2



	Observed Class			
Predicted Class	0	1	2	3
0	142	3	0	0
1	8	143	9	0
2	0	4	137	9
3	0	0	4	141

Accuracy
0.93

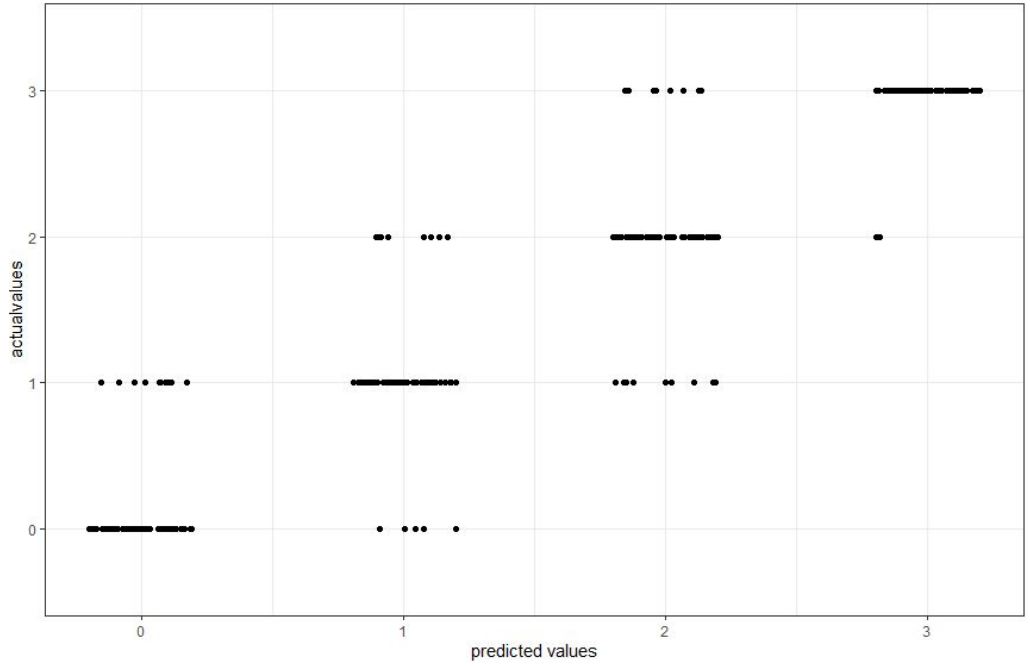


Tree Based Methods

Random Forest Multiple Binary Classifier

Accuracy
0.85

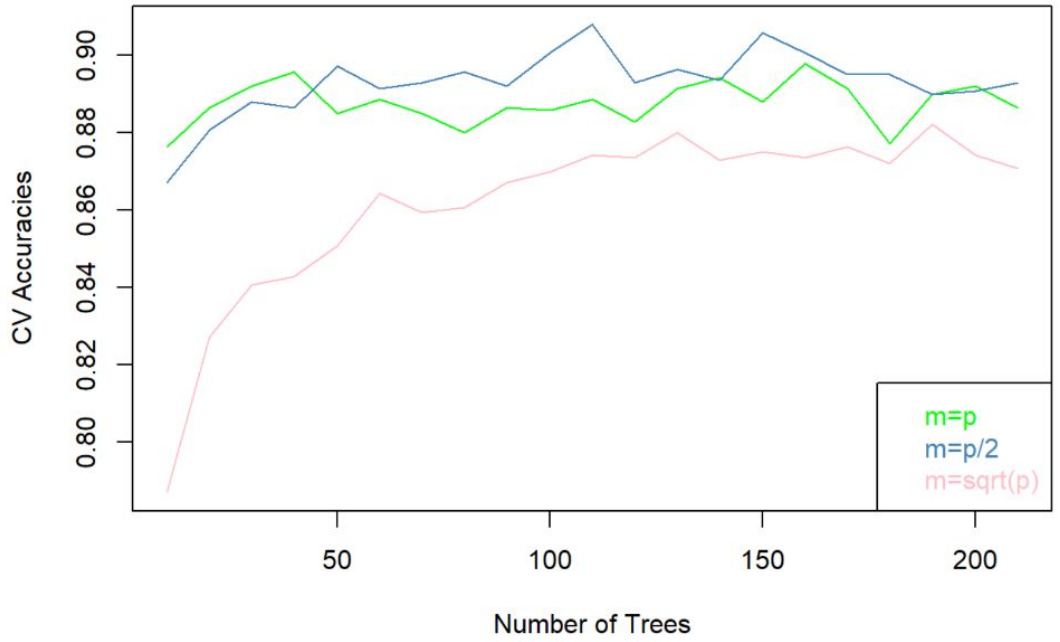
	precision
0	0.8750000
1	0.8219178
2	0.7323944
3	0.9705882



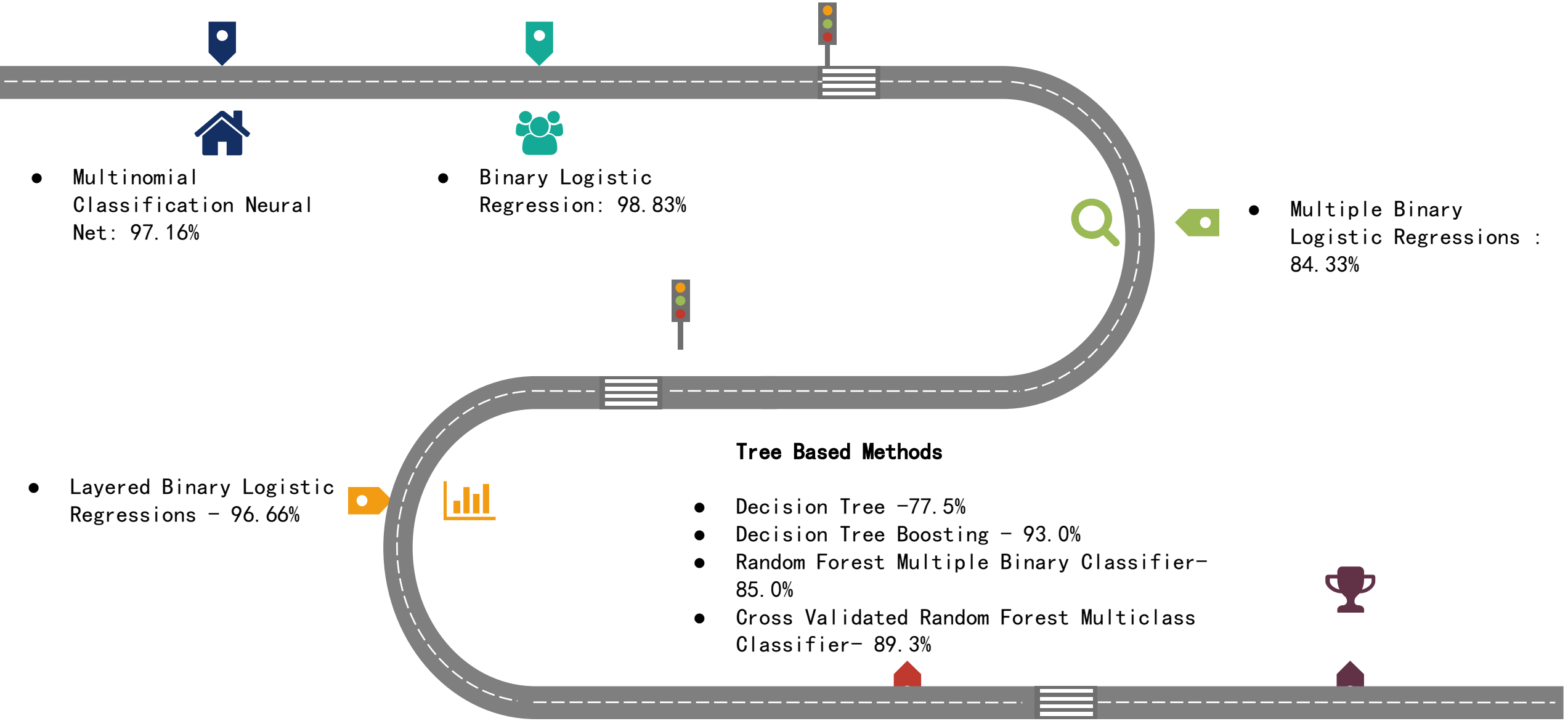
Cross Validated Random Forest Multiclass Classifier

Accuracy
0.893

	precision
0	0.9391892
1	0.8490566
2	0.8943662
3	0.8940397



Conclusion



- Multinomial Classification Neural Net: 97.16%
- Binary Logistic Regression: 98.83%
- Multiple Binary Logistic Regressions : 84.33%
- Layered Binary Logistic Regressions – 96.66%

Tree Based Methods

- Decision Tree –77.5%
- Decision Tree Boosting – 93.0%
- Random Forest Multiple Binary Classifier– 85.0%
- Cross Validated Random Forest Multiclass Classifier– 89.3%

Real World Applications Of Classification SML

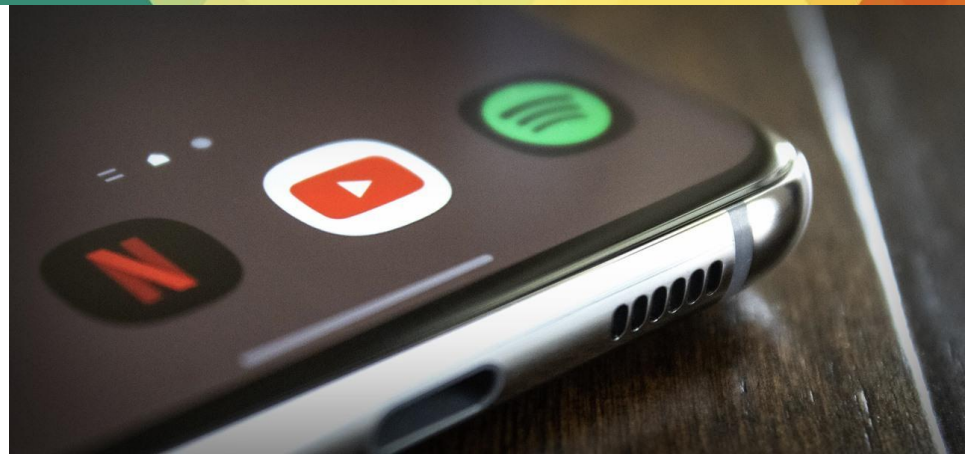
A classification model draws conclusion from observed values. Given inputs a classification model will try to predict the value of outcomes.

Classification predicts categorical class labels or classifies data based on the training set and the values in classifying attributes and uses it in classifying new data.

Real World Applications of Classification in Supervised Machine Learning:

- *classifying news into different categories(sports/entertainment/political)*
- *sentiment analysis*
- *classifying text into either positive negative or neutral*
- *segmenting customers for marketing purposes* etc.

Q&A



THANK YOU!