Project

# Data

For the data selection problem, we tried to choose with sufficient number of features, observations and including NA values. So I used *Internet Advertisements Data Set* which is published on Ucl Machine Learning Repository. The task is to predict whether an image is an advertisement ("ad") or not ("nonad"). Therefore this is a binary classification problem.

Data consists of 3279 number of points with 1558 features and a prediction class. First 3 columns are continuous and the rest is binary.

## Pipeline

Pipeline will be as followed:

* Data preparation
* Model Selection
* Try Models with:
  + All columns
  + Top 10 of the best features (extra)
  + Top x of the best features (extra)
  + Select From Model (SFM) method
  + RFE feature selection

## Data Preparation

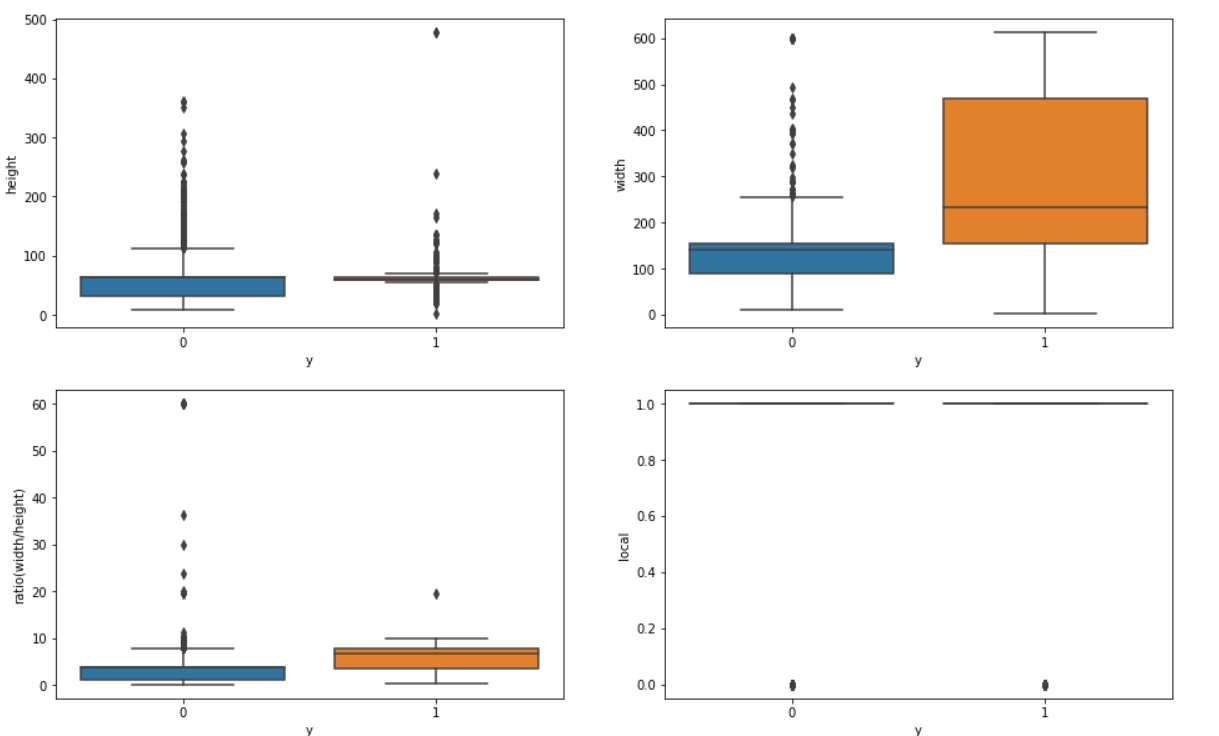
Data includes unknown values for the first 4 columns. As can be seen below, around 30-40% of respective columns are NA. In order not to lose any data, for continuous columns, we replaced NA values with the respective mean of that column, for binary, we replaced with mode of the column.

If we remove the binary column ‘local’, then datasize will be down to 2033.

|  |  |
| --- | --- |
| **Column** | **NA %** |
| width | 27.48% |
| height | 27.54% |
| ratio(width/height) | 27.75% |
| local | 38.00% |

Table 1: Shows the NA percentages

Below chart shows the first 4 columns and their box plot with respect to *y* value. We can clearly see that ‘width’ has an affect on *y* value more than the others.

Finally, we replaced prediction column, which consists of two categorical value suc as “ad” and “nonad”, with 1 for “ad”, 0 for “nonad”.

## Model Selection

For this dataset, we tried multiple models such as: ***Decision Tree, Random Forest Classifier, Bagging Classifier, Gaussian Naive Bayes, Support Vector Classifier*** and ***k-Nearest Neighbors*** and ***Gradient Boosting Classifier***.

### Scenario 1:

This scenario is solving models using all available features. So we define the models and predict. Since one of the performance criteria of the classification problems is *ROC-AUC* score, below we can see the roc auc scores of each model, individually. Random Forest and SVM performed pretty good when comparing to others.

 Table 2: Model scores for scenario 1

### Scenario 2: (Extra)

We tried to choose the best features by using a **sklearn.feature\_selection** module. Since there are hundreds of features and some of the features does not affect the outcome. Analyzed the best features by using *chi2* as a scoring function inside of the **SelectKBest** module. As we can see from the Table 3, top 10 features selected into the pipeline. We know that ‘width’ can have an impact on the outcome from the chart above, but the rest is about *url* features.

|  |  |
| --- | --- |
| **Columns** | **Score** |
| width | 69019.1 |
| 1243 | 1027.6 |
| 351 | 850.1 |
| 1399 | 814.7 |
| 1483 | 624.5 |
| 968 | 616.2 |
| 1344 | 604.6 |
| 1455 | 598.5 |
| 1435 | 576.6 |
| 1143 | 524.7 |

Table 3: Top 10 Best Features

Using this features, we tried the pre-specified models. Results can be seen below, which is not satisfying since almost all the models’ scores decreased, except naive bayes.

Table 4: Model scores for scenario 2

This shows us that we need to include more features than top 10.

### Scenario 3 (Extra):

We need to increase the number of features, so by using the method in previous scenario, we can select top *n* features. *n* corresponds the nth feature which the cumulative scores of all features reach the 80% of total, also known as the Pareto Chart line. So the *n* value is 88. Using these 88 features, we can see the roc auc scores of the models.

 Table 5: Model Scores for scenario 3

We are still not performing better than the first scenario (using all available features). However, model score performances improved with respect to previos scenario (using top 10 features), this proves us to include more features, even greater than 88.

### Scenario 4:

This scenario is about using a subset of features based on the feature importance values acquired by the method used. In this scenario, we used the **sklearn.feature\_selection.SelectFromModel** module for three models, which are *Random Forest*, *Gradient Boosting* and *Decision Tree classifier.* The threshold value to use for feature selection is kept default, which is *mean*. This means that features whose importance is greater or equal are kept while the others are discarded.

 Table 6: Number of features by model

We can see from the above that every model requires different number of features, this means that we can not select one number for all classifiers. Also, all three models performed slightly worse than the initial scenario (using all features).

### Scenario 5:

This scenario is about using best subset of features based on the recursive feature elimination process. Since feature elimination process in this scenario is computationally expensive, we select the best model for further analysis, which is Random Forest with roc\_auc score of 98%.

First of all, we need to find the optimal features for the model, but to find them among 1558 features, it is pretty hard. To overcome this problem, we need to eliminate the correlated features so that the process will not require long computational time. Also, taking correlated feautures will not be affect that much to model score. So, we removed features which have greater than 75% of correlation.

After that we used the Recursive Feature Elimination process. It is a process that assigning weights to features and select them recursively by considering smaller and smaller sets of features. We have chosen Random Forest classifier so the external estimator for feature selection is Random Forest Classifier. We used **sklearn.feature\_selection.RFE** module and the optimal number of features for Random Forest classifier is 212.

When we trained the model with these optimal features, our roc\_auc score of random forest was around 97.45% which beats all previous scenarios except the initial scenario. Although the difference is quite small that it can be negligible.

 Table 7: Comparison of all scenarios in terms of roc\_auc scores.

The reason can we cannot beat the initial scenario is that random forest does feature importance inside the algorithm by itself. Also, it chooses random number of subsets for each decision tree building process that it can perform better than eliminating some of the features. Neverthless, the difference is quite small and both of the model scores are pretty good.

## References

<https://archive.ics.uci.edu/ml/datasets/Internet+Advertisements>

<https://towardsdatascience.com/the-5-classification-evaluation-metrics-you-must-know-aa97784ff226>

<https://bookdown.org/max/FES/recursive-feature-elimination.html>

<https://math.usu.edu/~adele/forests/cc_home.htm>

<https://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.RFE.html>

<https://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.SelectFromModel.html#sklearn.feature_selection.SelectFromModel>

<https://www.kaggle.com/uciml/internet-advertisements-data-set>