PROJECT REPORT ON

<PROJECT-NAME>

<>

Submitted towards partial fulfilment of the criteria

For award of PGPDSE by Great Lakes Institute of Management

SUBMITTED BY

Group No. 7 [Batch: Dec 2018]

GROUP MEMBERS

1. **<>**
2. **<>**
3. **<>**
4. **<>**

RESEARCH SUPERVISOR

Mr. Srikar Muppidi

Great Lakes Institute of Management

ABSTRACT

The project "<> classifies images from an internet web page into two classes namely 'Advertisement' (ad) and 'Non-Advertisement' (nonad) using the data from the HTML version of that webpage. Logistic Regression and Supervised Machine Learning Models have been used to identify the best learner and achieve an accuracy of more than 95% in classifying the images data correctly. The code has been written in python language, the most widely used open source is programming language used in the field of Data Analytics and Data Science. It is observed that amongst a number of images on the webpage there are a few unwanted advertisement image banners also which can be eliminated using the learner module implemented in this project.

- Techniques:
  - Predictive Modelling
  - Supervised Machine Learning Models
  - Ensemble Techniques
  - SMOTE
- Tools:
  - Python
  - IDE - Jupyter Notebook
- Domain:
  - Data Analytics
  - Web Analytics

ACKNOWLEDGEMENT

We hereby certify that the work done by us for the implementation and completion of this project is absolutely original and to the best of our knowledge. It is a team effort and each of the member has equally contributed in the project.

Date: <>

Place: <>

CERTIFICATE OF COMPLETION

This is to certify that the project titled **"<proj-name> – <>"** for case resolution was undertaken and completed under the supervision of Mr. Srikar Muppidi for Post Graduate Program in Data Science and Engineering (PGP – DSE)

Mentor: Mr. Srikar Muppidi

TABLE OF CONTENTS

ABBREVIATIONS

S. No.	Full Form	Abbreviation
1.	Advertisement	Ad / ad
2.	Non Advertisement	NonAd / nonad
3.	Recursive feature Engineering	RFE
4.	Variation Inflation Factor	VIF
5.	Principal Component Analysis	PCA
6.	Logistic Regression	LR
7.	K - Nearest Neighbours	KNN

EXECUTIVE SUMMARY

"Ad-Filter" is an image classification learner that classifies images from an internet web page into ad. or nonad. using Unsupervised Machine Learning Models.

The collected dataset represents a set of possible advertisements on Internet pages. The features encode the geometry of the image (if available) as well as phrases occurring in the URL of the webpage, the image's URL and alt text, the anchor text, and words occurring near the anchor text. The size of the advert image has also been taken into consideration in the form of height, width and aspect ratio.

An internet or web page primarily consists of a lot of text and images. Some of these images are advertisements - wanted or unwanted. The need is to differentiate these advertisements from the regular text and images which are organically supposed to be a part of the webpage. We have tried to use the raw HTML information of the internet image and page information so as to cast it as a standard machine learning problem. We have implemented a supervised learner for the classification of the image as an advertisement or not with more than 97% accuracy.

There are a total of 1559 variables in the dataset formed by combining 3 continuous variables, 457 binary features derived from the URL terms of the webpage, 495 binary features from original image URL terms, 472 binary features from ANCURL terms, 111 binary features from alt terms of the image, 19 binary features extracted from caption of the image and 1 binary target variable

There are 3278 instances each corresponding to a candidate advertisement in the HTML file of a webpage. Given training instances that are pre-classified as being advertisement or not, an attempt has been made to implement different algorithms, bagging and gradient boosting techniques to finally learn a classifier that maps to either 'Ad' or 'Non-Ad'.

CHAPTER 1

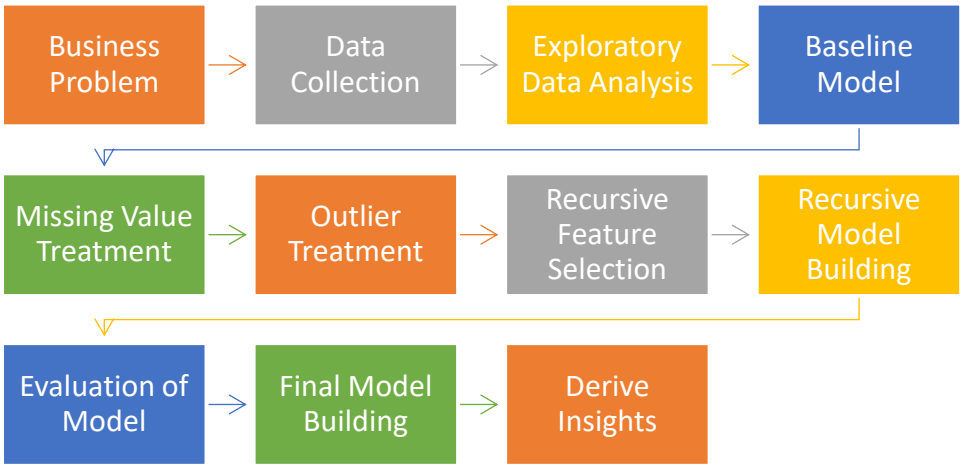
1. DATA SOURCES

The dataset used represents a set of possible advertisements on Internet pages. The features encode the geometry of the image (if available) as well as phrases occurring in the URL, the image's URL and alt text, the anchor text, and words occurring near the anchor text. There are a total of 1559 features - 3 continuous, one or more of the three continuous features are missing in 28% of the instances; missing values are interpreted as "unknown", 1556 binary class variables and a target variable with 3279 instances (2821 nonads, 458 ads). (This is the "STANDARD encoding" mentioned in the [Kushmerick, 99])

Citation: @misc. {Dua: 2017, author = "Dheeru, Dua and Karra Taniskidou, Efi", year = "2017", title = "{UCI} Machine Learning Repository", URL = "http://archive.ics.uci.edu/ml", institution = "University of California, Irvine, School of Information and Computer Sciences”}

Data source URL: https://archive.ics.uci.edu/ml/machine-learning-databases/internet_ads/

2. PROJECT FLOW



The process we chose to work on is an iterative process.

- We define the business problem.
- We collect data from different relevant sources.
- In Exploratory Data Analysis, we try to understand the structure of the data, the nature of the variables. The information present in the data. Statistical summary of the variables. A special attention is paid to the target variable distribution

- We build baseline models to get the idea of the minimum amount of performance that we can get from each of the models we have decided to use in the future.
- The first step of data preparation is Missing Value Treatment which means that If we have missing values in our data, then according to the percentage of the missing values in a row or column, we decide whether to drop a row or a column or replace those values with one of the central tendency measures as studied in statistics. Any method that will cause minimum impact on the data so we can get the actual pattern within the data for our models to learn.
- The other step of data preparation is Outlier Treatment. It is important to treat the outliers as they can affect the mean of the data in question and our evaluation of the data could be way off than the actual values.
- Recursive Feature Selection is a method to select features which have the maximum impact on the metrics of the models. The features which don't influence the performance of the model much are dropped.
- Recursive model building is one of the key highlights of our methodology as we are going to build models, evaluate their performance, find out the features or hyper parameters behind the performance of the model and then fine tune the possible causes and make our model better up to a certain threshold which in real world will be decided by the actual business requirements.
- Final Model building will be done when we are equipped with the information as to what drives the performance of the models and what we want from our model, such as, interpretability, explain ability, performance etc.
- We can also derive insights from the selected features, like, which features to focus on the most in the future.

3. PROCESS OVERVIEW

A stepwise process has been followed in implementation of the supervised learner.

1. The project starts with collation of data from different tags present in the raw HTML.
2. Once the data is ready, it has been imported as a pandas data frame and the structure of the data is noted.
3. A recursive approach has been followed while checking the data for missing values, variable data types and outliers.
4. Exploratory data analysis has been performed including the univariate and bivariate analysis for the continuous variables and for the categorical variables as far as possible.
5. Data has been visualized using different charts including distance plots, histograms, bar charts and box plots. Five-point summary statistics and correlation of data is also noted.

6. The target variable has been completely analysed and SMOTE technique has been applied to overcome the class imbalance in the data.
7. The missing values in data denoted by the character '?' have been replaced with NaNs.
8. Baseline models have been designed using 2 approaches – first by completely dropping the missing values which is a biased approach and second, by replacing the missing values with 0's which is the more logical technique. An accuracy of 98% is observed.
9. Based on conclusions on baseline model, different Missing Value Imputation and Outlier Treatment techniques are used to prepare data for application of models.
10. After processing the data, z-test has been performed and important features have been identified using Recursive Feature Engineering, feature importance, p-values, variation inflation factor.
11. Principal Components have also been developed using PCA technique so as to deal with the curse of dimensionality.
12. Multiple classification models including Logistic Regression, Decision Tress, Random Forest, K-Neighbours, and AdaBoost Classifier etc. are implemented recursively in the process to compare the results of each.
13. Finally models have been cross validated using K-fold Cross Validation so as to learn a highly sensitive classifier which maps or classifies the image data to 'Ad' or 'Non-Ad' with high accuracy.

4. LIMITATIONS

Although the classifier gives a highly accurate result, there are certain limitations to the project:

- In data science, prediction models can never be 100% accurate. For example in our dataset it could be because of the unidentified features which we fail to gather or observe may also contribute to the classification of the image. Our model uses data such as html script to identify ads. We can also look into the landing page of that embedded image and compare it to a list of websites already flagged as ads. This could make our model even more robust. Searching and identifying such features might improve our model performance.
- The data has been manually extracted while in real time we need to extract it from the raw HTML version of the webpage dynamically and thus a completely automated data extraction algorithm or code is required to derive data in the required format.
- In spite of being highly relevant data, since most of our data is a set of binary class variables, thus we see less scope of visualization and bivariate analysis in terms of categorical variables.

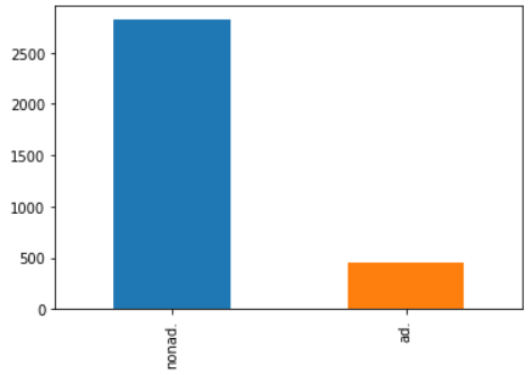
- Statistical measures such as Pearson's correlation coefficient matrix to find out multi-collinearity among variables was not applicable as it can only be used for continuous variables and not categorical.

It is not possible to apply VIF on all 1558 features to know importance of features. We can't apply it on categorical variables, as it measures R-square value, which is also only applicable for continuous variables which limited our feature selection process.

CHAPTER 2

1. VISUALIZATION OF THE TARGET

The following code and visualization shows the imbalance in the dataset.




```
nonad.    2820
ad.        458
Name: Target, dtype: int64

nonad.    0.860281
ad.       0.139719
Name: Target, dtype: float64
```

2. BASELINE MODEL

For building the base line model we have used the logistic regression using all the variables. Logistic regression uses log of maximum likelihood estimation to give the best fitting curve for classification. The accuracy of the baseline model is greater than 97% which is either due to shortage of samples in an unbalanced data or due to over-fitting of the models.

3. FEATURE SELECTION PROCESS

"MLE can be defined as a method for estimating population parameters (such as the mean and variance for Normal, rate (lambda) for Poisson, etc.) from sample data such that the probability (likelihood) of obtaining the observed data is maximize"

After building the base line model, & doing further analysis on the model summary, we removed variables one by one by taking into consideration their respective p-values & build the model again & again with the remaining variables.

We then used Recursive Feature Elimination (RFE) to find the top performing features affecting the target variables and then applying logistic regression on these variables to predict the target variables. RFE as its title suggests recursively removes features, builds a model using the remaining attributes and calculates model accuracy.

Three benefits of performing feature selection before modelling your data are:

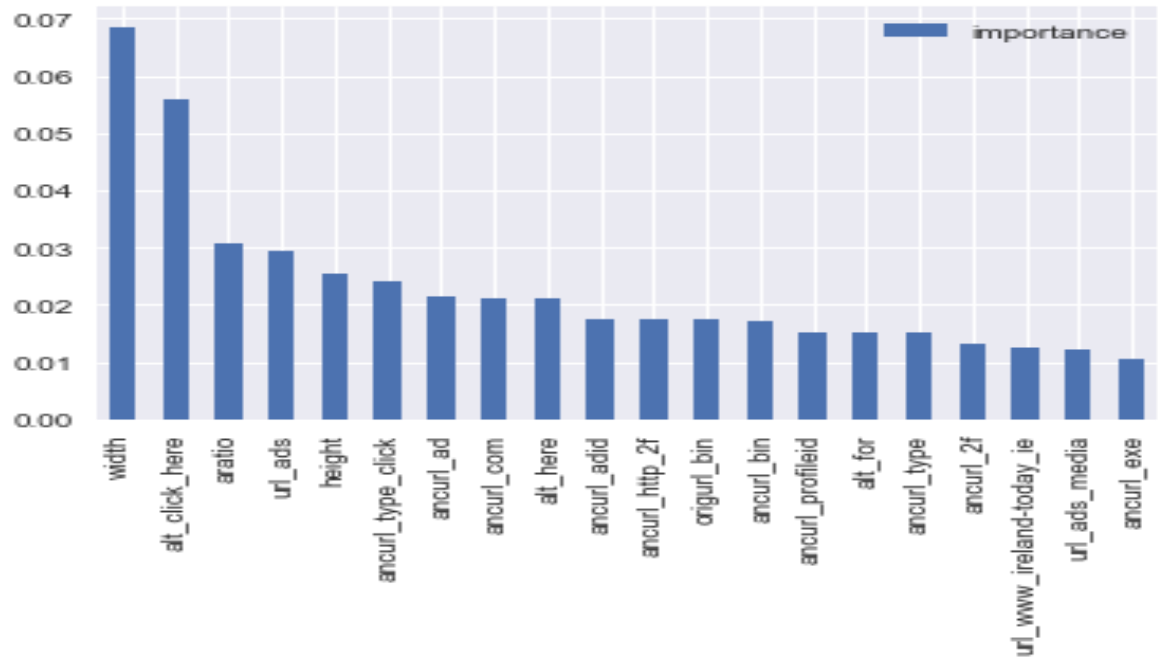
- **Reduces Overfitting**: Less redundant data means less opportunity to make decisions based on noise.
- **Improves Accuracy**: Less misleading data means modelling accuracy improves.
- **Reduces Training Time**: Less data means that algorithms train faster

We have an ensemble technique which provides us with the relative importance of the variables i.e. Random Forest.

"Feature importance is used commonly & tells us that which variable is contributing the most to a model & is critical to interpreting the results."

The technique resulted in providing some visualization & insights from which we were able to recognize the important variables & tried removing the insignificant variables, taking into consideration the relative importance values.

Now we will build our base model using these 20 features. The result of fitting the base model is shown below:



Next we use the GLM model,

Generalized Linear Model Regression Results

Dep. Variable:	Target	No. Observations:	1657
Model:	GLM	Df Residuals:	1639
Model Family:	Binomial	Df Model:	17
Link Function:	logit	Scale:	1.0000
Method:	IRLS	Log-Likelihood:	nan
Date:	Sat, 16 Feb 2019	Deviance:	nan
Time:	22:38:26	Pearson chi2:	1.49e+09
No. Iterations:	100	Covariance Type:	nonrobust

	coef	std err	z	P>|z|	[0.025	0.975]
width	0.0270	0.002	15.093	0.000	0.023	0.030
alt_click_here	36.2159	3.43e+07	1.05e-06	1.000	-6.73e+07	6.73e+07
aratio	-0.9760	0.074	-13.211	0.000	-1.121	-0.831
url_ads	2.8989	0.786	3.687	0.000	1.358	4.440
height	-0.0624	0.003	-18.001	0.000	-0.069	-0.056
ancurl_type_click	-25.2405	2.69e+07	-9.38e-07	1.000	-5.28e+07	5.28e+07
ancurl_ad	-0.0140	1.425	-0.010	0.992	-2.808	2.780
ancurl_com	5.4154	0.786	6.887	0.000	3.874	6.957
alt_here	-34.2060	3.43e+07	-9.96e-07	1.000	-6.73e+07	6.73e+07
ancurl_adid	37.6180	3.02e+07	1.25e-06	1.000	-5.92e+07	5.92e+07
ancurl_http_2f	-25.2405	2.69e+07	-9.38e-07	1.000	-5.28e+07	5.28e+07
origurl_bin	35.9869	1.59e+07	2.26e-06	1.000	-3.12e+07	3.12e+07
ancurl_bin	0.8670	0.353	2.459	0.014	0.176	1.558
ancurl_profileid	-25.2405	2.69e+07	-9.38e-07	1.000	-5.28e+07	5.28e+07
alt_for	0.0869	0.849	0.102	0.918	-1.577	1.750
ancurl_type	-0.7879	3.17e+07	-2.49e-08	1.000	-6.21e+07	6.21e+07
ancurl_2f	40.5215	6.71e+07	6.04e-07	1.000	-1.32e+08	1.32e+08
url_www_ireland-today_ie	41.1815	4.75e+07	8.68e-07	1.000	-9.3e+07	9.3e+07
url_ads_media	69.8262	1.46e+07	4.77e-06	1.000	-2.87e+07	2.87e+07
ancurl_exe	2.0423	1.048	1.949	0.051	-0.011	4.096

We can eliminate the features that are having p-value greater than 0.05.

After that we have used the logistic regression and built the model using 20 variables.

The result for which is shown below.

```
lr = LogisticRegression()
model_lr = lr.fit(x_f,y_train)
pred_lr = model_lr.predict(x_t)
metrics.accuracy_score(y_test,pred_lr)
```

```
0.9592123769338959
```

```
print(metrics.classification_report(y_test,pred_lr),"\n\n")
```

	precision	recall	f1-score	support
0	0.97	0.99	0.98	611
1	0.91	0.79	0.84	100
avg / total	0.96	0.96	0.96	711

The sensitivity of the model is 0.80 which means 80% of the time we are able to predict an Advertisement image as 'ad' which is decent enough using only 20 features in comparison to the 1558 features provided originally.

4. ENSEMBLE TECHNIQUES

Ensemble methods is a machine learning technique that combines several base models in order to produce one optimal predictive model.

Random Forest: *To say it in simple words, random forest builds multiple decision trees and merges them together to get a more accurate and stable prediction*

One big advantage of random forest is, that it can be used for both classification and regression problems, which form the majority of current machine learning systems and it helps in reducing the variance in model.

Using the 20 variable given by random forest feature importance function we have built the model. The results of which is shown below.

```
rf = RandomForestClassifier()
model_rf = rf.fit(x_f,y_train)
pred_rf = model_rf.predict(x_t)
metrics.accuracy_score(y_test, pred_rf)
```

```
0.9845288326300985
```

```
print(metrics.classification_report(y_test,pred_rf),"\n\n")
```

```
             precision    recall  f1-score   support

          0       0.98      1.00      0.99       611
          1       0.99      0.90      0.94       100

avg / total       0.98      0.98      0.98       711
```

Bagging using Decision Tree and Random forest:

Decision Trees over fit the model and increases the variance. This makes the model vulnerable. Bagged Trees averages many models to reduce the variance. Although, Bagging is often applied to Decision Trees but it can be used with any type of method. In addition to reducing variance, it also helps avoid overfitting.

Results with bagging using decision tree as estimator are as follows.

```
bagg_dt = BaggingClassifier(base_estimator=dtree, n_estimators=25, bootstrap=True, oob_score=True, random_state=123)
model_bagg_dt = bagg_dt.fit(x_f,y_train)
pred_bagg_dt = model_bagg_dt.predict(x_t)
metrics.accuracy_score(y_test, pred_bagg_dt)
```

```
0.9746835443037974
```

```
print(metrics.classification_report(y_test,pred_bagg_dt),"\n\n")
```

```
             precision    recall  f1-score   support

          0       0.98      0.99      0.99       611
          1       0.94      0.88      0.91       100

avg / total       0.97      0.97      0.97       711
```

Results of bagging with random forest are as follows

```
6]: bagg_dt = BaggingClassifier(base_estimator=rf, n_estimators=25, bootstrap=True, oob_score=True, random_state=123)
    model_bagg_dt = bagg_dt.fit(x_f,y_train)
    pred_bagg_dt = model_bagg_dt.predict(x_t)
    metrics.accuracy_score(y_test, pred_bagg_dt)
```

```
6]: 0.9789029535864979
```

```
8]: print(metrics.classification_report(y_test,pred_bagg_dt),"\n\n")
```

```
             precision    recall  f1-score   support

          0       0.98      1.00      0.99       611
          1       0.97      0.88      0.92       100

avg / total       0.98      0.98      0.98       711
```

5. MODEL EVALUATION

Training data: The actual dataset that we use to train the model, model sees and learns from this data.

Validation data: the sample of data used to provide an unbiased evaluation of a model fit on the training dataset while tuning model hyper parameters

Test data: The sample of data used to provide an unbiased evaluation of a final model fit on the training dataset.

Data has been split such that 70% of the instances have been used to train the learner and the remaining 30% has been used to validate the model. A 10 fold Cross Validation has been performed on each of the classification models that are implemented so as to achieve unbiased results and avoid over-fitting of the model.

Finally, the learners have been trained well on the data and the models are not over-fit. Hence we see that Random forest is our best performing model when we use it with top 20 performing features.

Sensitivity of a model is its ability to detect the abnormality in the data, which is the most important evaluation metric. Sensitivity or Recall for the models is as below:

S. No.	Model	Sensitivity
1.	Logistic regression	0.79
2.	Decision Tree classifier	0.90
3.	Bagging with Decision Tree	0.88
4.	Bagging with Random Forest	0.88
5.	Random Forest	0.90

We hereby see that both decision tree and random forest are giving the same sensitivity but we will use the random forest classifier since it is more stable.

CHAPTER 3

1. COMPARISON TO BENCHMARK

The internet sites usually make money from third-party advertisements that pop up in the form of banners on any webpage. Anyone who is interested in the content can click on the advertisement and jump to the advertiser's page. However, there are many users who prefer to ignore or eliminate all such advertisements that break their flow or slow up their browsing by increasing the download time.

The point of reference for the project was "The Ad-Eater System" which is a fully implemented browsing assistant that automatically removes advertisement images from Internet pages and takes and inductive approach to automatically generating rules based on training samples. The three modules involved are:

- A collection of training examples in an offline manner.
- Build an algorithm to process these samples and generate rules for discriminating advertisements from non-advertisements.
- An online module to scan internet pages that a user is scrolling and remove the image that are classified as 'advertisement' by the learned rules.

Our benchmark is to implement a new learner which can be used by such systems to be informed of whether any piece of information or image on a webpage is an advertisement or not in an easier way and less downtime. Without making any assumptions about the data, we have trained our classifiers in such a way that they accurately classify the information basis the training data. This can be used by various assistants like Ad-Eater to carry the process of classification of web image into Advertisement or Non-Advertisement and act based on results whether to remove that piece of information or not.

Base Line Model: Logistic Regression model taking all variables into consideration.

```
4]: lr = LogisticRegression()
    model_lr = lr.fit(x_train,y_train)
    pred_lr = model_lr.predict(x_test)
    metrics.accuracy_score(y_test,pred_lr)

4]: 0.9760900140646976

5]: print(metrics.classification_report(y_test,pred_lr),"\n\n")

                precision    recall  f1-score   support

             0       0.98      0.99      0.99       611
             1       0.93      0.90      0.91       100

    avg / total       0.98      0.98      0.98       711
```

Here we are using all the features and our sensitivity is 0.90 but since our number of features are 1558 it is possible that not all of them are having same importance. Hence we have use feature importance function of random forest and now we will see the result by extracting the top 20 features affecting the target variable.

```
lr = LogisticRegression()
model_lr = lr.fit(x_f,y_train)
pred_lr = model_lr.predict(x_t)
metrics.accuracy_score(y_test,pred_lr)
```

```
0.9592123769338959
```

```
print(metrics.classification_report(y_test,pred_lr),"\n\n")
```

	precision	recall	f1-score	support
0	0.97	0.99	0.98	611
1	0.91	0.79	0.84	100
avg / total	0.96	0.96	0.96	711

Although our sensitivity has come down to 0.79 but still we are in decent position since we are using only 20 features to explain the variance.

Now we will compare the base line Random Forest model and result with same Random Forest using the top 20 features.

Base line Random Forest Result is shown below.

```
76]: rf = RandomForestClassifier()
     model_rf = rf.fit(x_train,y_train)
     pred_rf = model_rf.predict(x_test)
     metrics.accuracy_score(y_test, pred_rf)
```

```
76]: 0.9718706047819972
```

```
77]: print(metrics.classification_report(y_test,pred_rf),"\n\n")
```

	precision	recall	f1-score	support
0	0.98	0.99	0.98	611
1	0.93	0.87	0.90	100
avg / total	0.97	0.97	0.97	711

With using only 20 features Random Forest Result is shown below.

```
rf = RandomForestClassifier()
model_rf = rf.fit(x_f,y_train)
pred_rf = model_rf.predict(x_t)
metrics.accuracy_score(y_test, pred_rf)
```

```
0.9845288326300985
```

```
print(metrics.classification_report(y_test,pred_rf),"\n\n")
```

	precision	recall	f1-score	support
0	0.98	1.00	0.99	611
1	0.99	0.90	0.94	100
avg / total	0.98	0.98	0.98	711

We see that after using only 20 features the sensitivity of our Random model has increased from 0.87 to 0.90 as compared to base model of Random Forest.

Hence we come to conclusion that random forest with 20 features is able to explain the variability in a very decent manner as compare to logistic regression's base line model which is using all the variables for prediction.

2. IMPLICATIONS

Our implication in the project is that not all images on a webpage are advertisements but certain images are found which are classified as advertisements and this information can be used directly or indirectly by different Ad-blocker systems if they feed the implemented learner with the required data that can be easily extracted from the raw HTML in no time while the user browses through a webpage.

As far as the business value is concerned, our project shows huge implications on the user-experience of an internet user. He/she doesn't have to deal with the huge amount of clutter that these ads cause when the website content is being viewed. These ads are flashy and have constantly changing images to grab attention or to distract the user from the actual content. They are also found in a form meant to deceive the user into clicking it when it is actually not a relevant link but a link that might take the user to an irrelevant website. Although our model is built to label the images as ad or no ad using its URL but we can also make it usable for identifying malwares, bloatware and suspicious websites using URL. The link could also

download bloatware into our machines or even download viruses which could completely crash our systems requiring a complete reset leading to the loss of confidential and important data or information which could cost the user. We live in a connected world and by accidentally clicking a stray link can install virus that remains dormant in our laptops until it finds the right time to copy sensitive information and also to reach other machines active in the same network. Since our method is different, it could make it harder for these websites to be passed as a relevant website.

3. CLOSING REFLECTIONS

Reflecting our learnings from the project:

- We learned how a basic HTML Script can give us important decision making insights about a website and its content.
- To improve precision and recall in an imbalanced data we must use ensemble techniques.
- Use codes that make our code shorter, cleaner and more beautiful and also print simple outputs that can be understood by anyone.
- SMOTE technique used to deal with unbalanced dataset.

Reflecting our approach for consecutive projects:

- Continuous and Categorical variables need to be dealt separately and their interaction needs to be studied before we can further use them as features in a model.
- Additional knowledge of data extraction from HTML files and tags, data mining techniques.
- Based on the dataset, more stringent feature engineering and feature selection process.
- Evaluating each step not only with a mathematical point of view but also from a business perspective.

BIBLIOGRAPHY

1. Learning to remove Internet advertisements by Nicholas Kushmerick – University College Dublin
2. Introduction to Statistical Learning using R by Hastie and others
3. Course textbooks etc.
4. <Also specify any websites or blogs or articles or whitepapers or libraries or tools being used>