Internet Advertisement DataSet Analysis

Snehil Grandhi
Maths and Computing
Indian Institute of Technology, Delhi
mt6180795@iitd.ac.in

Shreyansh Choudhary
Maths and Computing
Indian Institute of Technology, Delhi
mt6180794@iitd.ac.in

***Abstract*—Many popular websites are loaded with ads that hinders one's reading experience. Some of the ads are malicious that can do major harm to one's computer and one's privacy. An ad blocker will help you remove many online ads and reduce the opportunity for malvertising attacks.**

Most of the ad blockers, used now-a-days, block the adverts based on a blocking list (of URL of the Adverts). The machine learning models present remarkable results for the same. This report aims to optimize such machine learning models. Specifically, it investigates various dimension reduction, feature extraction and feature selection methods for the dataset. The report also employs Rough-set and shapley-value based methods for feature selection. The report also mentions some techniques used for boosting the algorithm training time.

We have combined various pre-existing approaches to obtain a better model. Empirical comparison with several other existing models shows that the Random-forest classifier on Shapley values based feature selection along with feature extraction of nlp-score gave the most accurate classification results on our dataset.

***Index Terms*—Exploratory Data Analysis, Machine Learning, Deep Neural Network, Rough set, BERT embeddings, Cosine Similarity, PCA, Random Projection**

I. Introduction and Motivation

Advert detection software/plug-ins are abundantly used for a better internet surfing experience. When your ad blocker stops the analytics code, text and imagery that comes with every online ad, your browser is free to concentrate on loading the actual content of the site you want to visit – say, the article you want to read or the video you want to stream – without distractions.

The main benefit of using and ad blocker is we get cleaner websites, no annoying pop-ups, no sudden and unwelcome sound effects... Replacing all the ads with pleasant, reader-friendly white space can make the time you spend online feel like a radically different experience.

Tree models have shown remarkable results for the dataset.

In this report we use PCA algorithm and Random Projection algorithm for reducing the dimensionality of our features space.

We have used Rough-set based algorithm to generate a pseudo reduct. Various Tree based classifiers gave very promising results.

We have also applied Shapley value based algorithm for feature selection.

We have also used a novel feature extraction algorithm for generating the **nlp-score**.

Since, our data is loosely coupled so we have used attribute based Neural Experts to boost the training speed of our neural net models. Shapley selection on dataset with extracted features gives promising results.

II. Past Work

Random projections[6] have recently emerged as a powerful method for dimensionality reduction. Theoretical results indicate that the method preserves distances quite nicely; however, empirical results are sparse. Fradkin et al.[5] performed a number of experiments to evaluate random projections in the context of inductive learning.

Alvarez et al[12] proposed an effective way to reduce the time complexity of back propagation in presence of multiple loosely coupled data sources. They proposed a new topological representation for the ANN called mixture of attribute experts. They provided a proof of concept of their approach in context of mining over loosely coupled data sources by applying a mixture of attribute experts ANN to the problem of detecting ad/nonad, using our dataset.

Cohen et al[15] proposed the Contribution-Selection algorithm (CSA), a novel algorithm for feature selection. The algorithm is based on the Multiperturbation Shapley Analysis, a framework which relies on game theory to estimate usefulness. The algorithm iteratively estimates the usefulness of features and selects them accordingly, using either forward selection or backward elimination.

We adopt all these approaches and perform further data analysis to obtain better results.

III. About the Dataset

The Internet Advertisement dataset from UCI Repository[1] has been sampled from a set of webpages. Only those images are considered which are enclosed in anchor tag, i.e. they are a hyperlink for a destination web page. Roughly 14% of these images contain advertisements and the rest do not. There are missing values in approximately 28% of these instances. There are total 1557 attributes related to image dimensions, phrases in the URL of the image, text occurring in the image's anchor tag and its caption. The first 3 attributes encode the image geometry. The binary attribute, local, indicates whether the image URL points to a server in the same internet domain as the document URL. The remaining features are based on phrases in the different parts of the HTML document i.e. Original URL, Anchor URL, the alt text in the anchor tag and

the caption of the image. As most of the features are binary and majority of the entries are 0, Our data is sparse.

A. Other Uses Of The Dataset

Some other uses which one may deprive of this dataset are:

1) If we could get more samples, we can create a training set on the support phrases which are likely to be there in an internet advertisement's caption/alt-text. On this training set, we can train a model with attention mechanism (transformers like BERT, roberta) to generate more support phrases which are likely to be in the advertisement's caption/alt-text.
2) If we could get more samples, we can create a training set with another target called type of ad. We can try to detect what type of advertisement it is, trial/introductory/Shared Value Ads/differentiating, especially based on the caption/alt-text of the image advertisement. We can train Models on this and can further detect the type of the advertisement.
3) Another type of ad classification we can employ is to detect whether an ad is commercial or malicious. This can be achieved by considering url features also.

IV. Data Cleaning and Pre-processing

1) Missing Values:

- Height: 903 out of 3279 samples (about 28%)
- width: 901 out of 3279 samples (about 28%)
- Aspect Ratio: 910 out of 3279 samples (about 28%)
- local: 15 out of 3279 samples (about 0.5%)
- Other Binary Values: Don't Have Missing Values.

We used KNN imputation[2] to replace the missing values. KNN is an algorithm that is useful for matching a point with its closest k-neighbors in a multi-dimensional space. It can be used for data that are continuous, discrete and categorical which makes it particularly useful for dealing with all kind of missing data.

The assumption behind using KNN for missing values is that a point value can be approximated by the values of the points that are closest to it, based on other variables. It is more accurate than the mean, median or most frequent imputation methods.

We have scaled the data using MinMaxScaler so that the values of attributes, being in different ranges, don't put a bias on the decision by KNN imputer. Now, to choose a particular value of k, we used repeated Stratified K fold to train the imputed dataset on Random Forest Classifier. Here we present the mean and std. deviation of the scores generated. We used accuracy to compare the results.

Scores of the K-Fold Cross Validation Vs K							
k	1	3	5	7	9	15	18
mean	0.974	0.978	0.979	0.979	0.979	0.979	0.979
std. deviation	0.008	0.006	0.006	0.006	0.007	0.007	0.006

We choose k=5 because it has the highest mean score and the lowest deviation for smallest such k.

2) Outlier Removal: The ad vs nonad ratio in the sample is is nearly 1:6. Hence, we plan use to SMOTE[16] to balance the Samples. SMOTE algorithms are sensitive to outliers and hence, it is mandatory to remove them to achieve a better distribution.

Isolation Forest is an anomaly detection algorithm. Isolation forest explicitly isolates, anonymous records (Outliers) by taking advantage of inherent properties of anomalies(they have unusual values for the set of covariates). Other methods are constrained to low dimensional data, and small data-size due to computational expense. Isolation Forest has a linear time complexity.

Here, we prefer Isolation Forest over the standard Inter-Quartile Range method(IQR). IQR assumes that the distribution of the attribute is normal. We have plotted box-plots fig- 2 for height and width attribute.

The outliers marked by the IQR method are densely populated in the area, we can infer from the plot that the continuous attributes don't follow Gaussian distribution. Isolation Forest method works well in such scenarios.

After removing outliers, the samples size reduces to 3115 (ad:377, nonad: 2738).

3) Sample generation: As we mentioned earlier, The ratio of ad vs nonad is 1:7, also after removing outliers, the ratio changed to 1:7.5. In this scenario, the classifier won't perform equally good for both the classes. Hence, we use SMOTE algorithm to balance the data for both the classes.

We have over sampled it to increase the ratio of ad to nonad to 1:2 to reduce the bias in the data which may be caused by the assumed distribution in the SMOTE algorithm. We will use SVM-SMOTE because it works well for sparse datasets. Then, we scale the data so that the training models don't associate a bias for any attribute.

V. Approaches

Now we present different training models on the data. For these different approaches, we use different modifications on the data. For some of the experiments we have done, we compare the results in skewed data and synthetic data generated by SMOTE. For such cases, we use imputed data. Below is the table that summarizes what all modifications to the data is there.

Data	Modification
Complete data	Applied KNN imputation. Used as skewed data and for finding reducts.
Synthetic Data	Replaced missing values and Applied SMOTE to balance the classes Used this for nearly all the purposes.

We present 3 different approaches for this problem:

- Dimension reduction: PCA and Random Projections
- Feature Selection: Geometry, Shapley Values and Rough Sets
- Complete Data

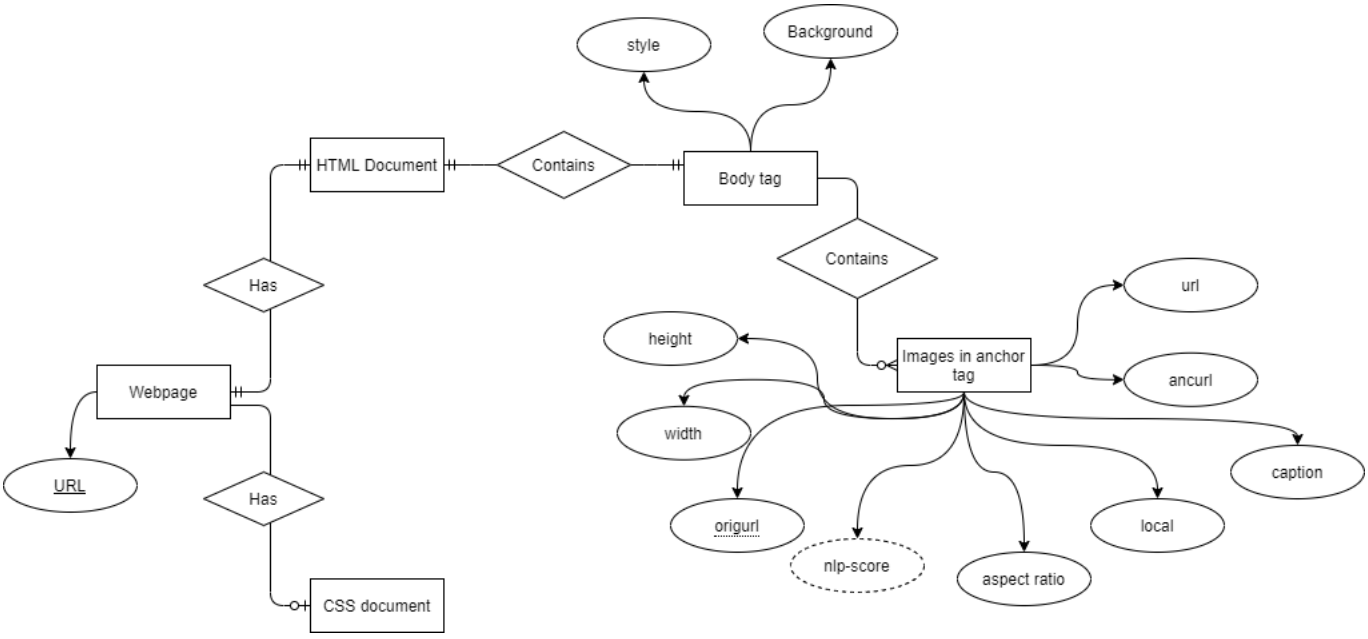


Fig. 1: ER Diagram

A. Dimension Reduction

1) Principal Component Analysis: Principal component analysis (PCA)[4] is an unsupervised algorithm that creates linear combinations of the original features. The new features are orthogonal, which means that they are uncorrelated. Furthermore, they are ranked in order of their explained variance.

There are 1500+ attributes in our dataset with around 4000 samples. Hence, we reduce the dimension of the data to 15 using PCA. With Heat map fig- 3, we verified the correlation between the generated features. The correlation between all the generated features is nearly 0.

Also, we present the distribution plots fig- 4 of the all the features extracted by PCA. From the figure, we can see that majority of the features here follow normal distribution or near-normal distribution, with very small variance which is explained by the binary nature of the majority attributes in the initial dataset.

Now, for the data we have, we trained few models. Those are: Support Vector Classifier, KNN classifier, Random Forest Classifier, XGB Classifier,Logistic Regression, Decision Tree Classifier(this classifier uses optimised CART algorithm which is better than C4.5 rules). For each of them we mention the accuracy and F1-Score.

Model Performance		
Model	Accuracy	F1
Support Vector Classifier	0.89	0.87
KNN classifier	0.96	0.95
Random Forest Classifier	0.97	0.97
XGB classifier	0.95	0.94
Logisitc Regression	0.94	0.92
Decision Tree Classifier	0.97	0.97

For further experiments, we trained Neural Networks on the data. We summarize the results for the same in the Table.

NN Model Performance				
S.No.	No. of Layers	Total Parameters	Train Accuracy	Test Accuracy
1	4	1788	0.9465	0.9501
2	5	2726	0.9681	0.9525
3	**4**	**1821**	**0.9686**	**0.9645**
Note: The AUC for all of these models were near 0.97-0.99				

Since, the AUC score for each model was very high, we can infer that the SMOTE generalised the data well.

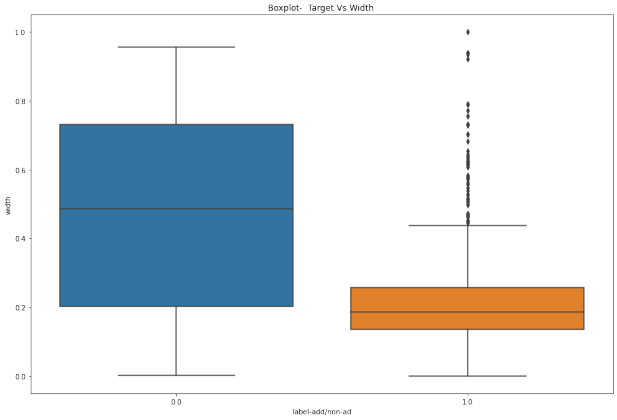
2) Random Projection: In random projection[7,8], the original d-dimensional data is projected to a k-dimensional subspace through the origin, using a random k × d matrix R whose columns have unit lengths. In matrix notation, the conversion is:

$$X^{RP}_{k\times N} = R_{k\times d} X_{d\times N}$$

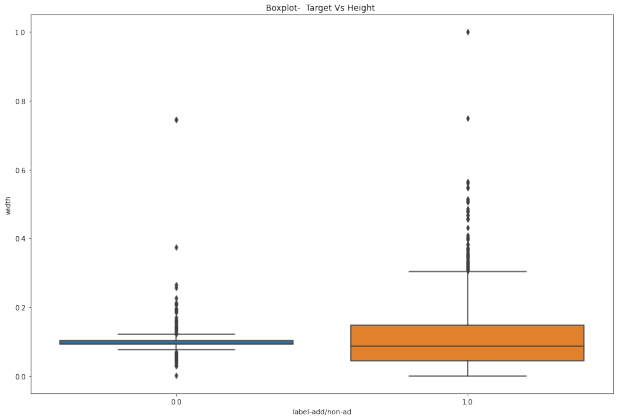
The key idea of random mapping arises from the Johnson-Lindenstrauss lemma i.e. a small set of points in a high-dimensional space can be embedded into a space of much lower dimension in such a way that distances between the points are nearly preserved. The map used for the embedding is at least Lipschitz, and can even be taken to be an orthogonal projection.

We reduced the dimension of the dataset to 198 using random projection. We plotted heat map to remove the highly correlated features. For heat map, we randomly group 50 features so that we can observe the results (fig - 5.

In the Heat Map (figure- 5, we can see all the features are not highly correlated. The highest correlation among any pair of feature is less than 0.5. Hence, we can be assured that the features are not redundant.



(a) Width



(b) Height

Fig. 2: Box Plots of Width and Height, These boxplot shows that IQR method to remove the outliers is not appropriate as they dont follow normal distribution.

For this data, we again trained the same set of models to judge the performance of Random Projections.

Model Performance		
Model	Accuracy	F1-Score
Support Vector Classifier	0.96	0.96
KNN classifier	0.95	0.94
Random Forest Classifier	0.96	0.96
XGB classifier	0.96	0.96
Logistic Regression	0.94	0.93
Decision Tree Classifier	0.90	0.89

For further experiments, we trained Neural Networks on this data. We summarize the results for the same in the Table.

NN Model Structure		
Layer Type	Shape	Parameter count
Dense	100	19900
Dense	30	3030
Dense	20	620
Dense	1	21
Note: Total trainable parameters = 23,571		

Results of the best the neural net trained :-

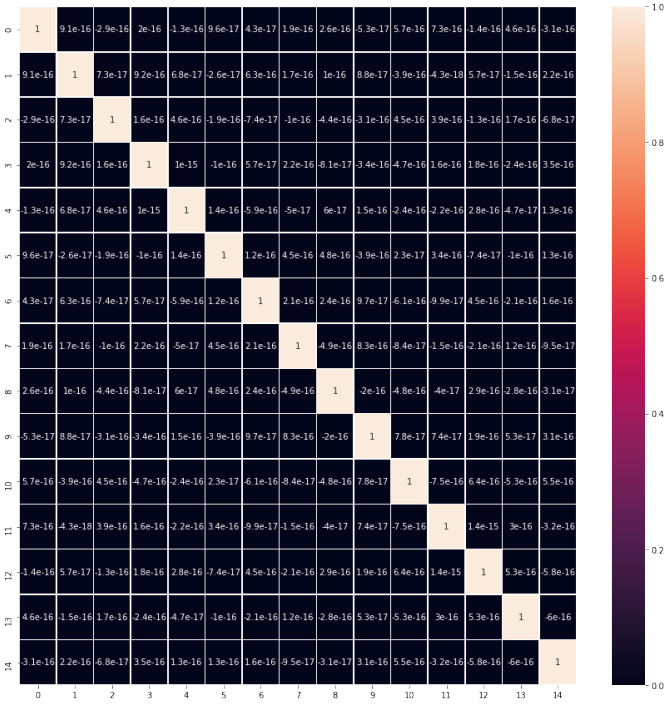


Fig. 3: HeatMap of features generated by PCA

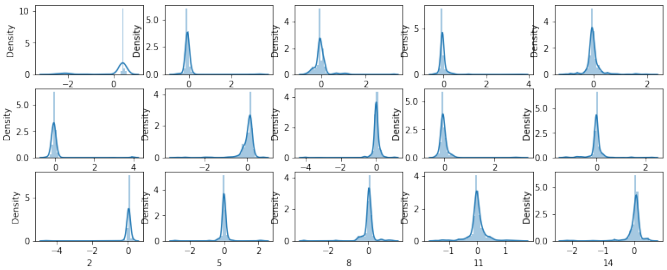


Fig. 4: Distributions of features generated by PCA

NN Model Result		
Train accuracy	Test Accuracy	validation Accuracy
0.9928	0.9708	0.9975
Train AUC	Test AUC	validation AUC
0.9991	0.9894	1.000
Note: Total trainable parameters = 23,571, Layers count = 4		

B. Feature Selection

1) Geometry: On plotting distribution plots (fig- 6) of the continuous features for ad vs nonad, we observe that their variance can sufficiently explain decision boundary between the ad and the nonad class. We assume that features with a higher variance may contain more useful information. Below we plot the distribution plots of height, width, aratio 6. Also, we select the local feature from the dataset.

In the heat map(fig: 7), we see that the correlation among width and aspect ratio is quite high. It is because aspect ratio is width/height and height varies in a smaller region, hence width and aratio have high correlation. Observing from the distribution plot (fig- 6) and the violin plot(fig- 8), the distribution of ad vs non ad for width and aratio are quite

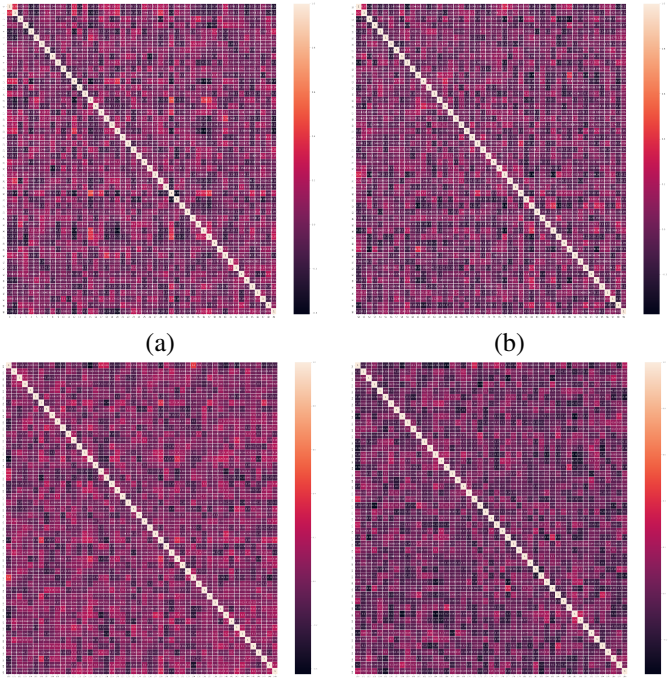


Fig. 5: Heat Maps of Randomly grouped features generated from Random Projection.

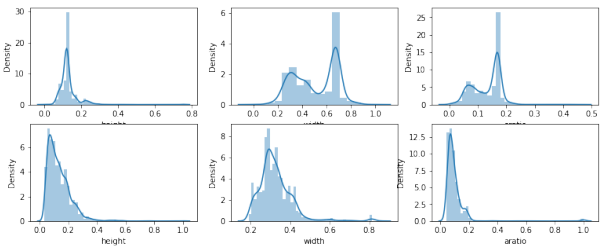


Fig. 6: Distribution Plot of Height, Width and Aspect Ratio. above images correspond to ad and lower ones are for nonad

different, but for height it may be same. Hence, we train the following models considering both the possibilities to obtain the best results.

For training we choose the same set of models. The results are presented in the table below.

Model Performance taking height, width, aratio, local.

Model Performance with Height		
Model	Accuracy	F1-Score
Support Vector Classifier	0.84	0.79
KNN classifier	0.91	0.89
Random Forest Classifier	0.93	0.92
XGB classifier	0.90	0.88
Logistic Regression	0.85	0.80
Decision Tree Classifier	0.90	0.89

Model Performance taking only width, aratio, local.

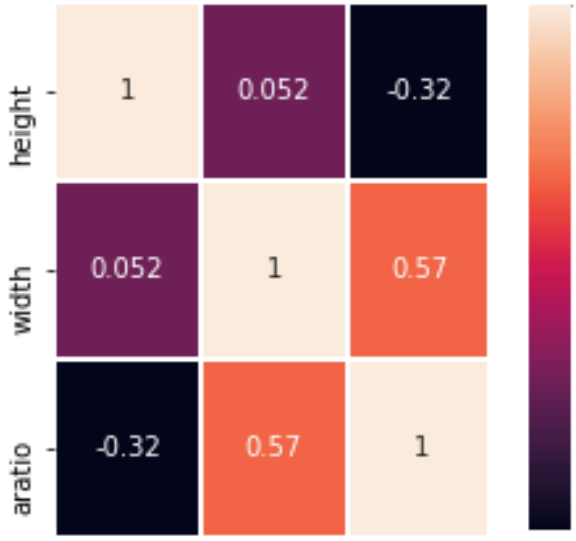


Fig. 7: Heat Map of Geometry features

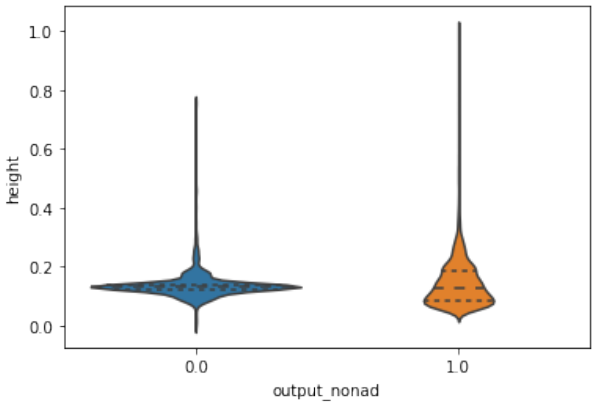


Fig. 8: Violin plot of Height

Model Performance without Height		
Model	Accuracy	F1-Score
Support Vector Classifier	0.84	0.79
KNN classifier	0.90	0.88
Random Forest Classifier	0.92	0.91
XGB classifier	0.89	0.87
Logistic Regression	0.84	0.79
Decision Tree Classifier	0.91	0.90

As we can see, in Random Forest Classifier, we obtained better results when we select height. Hence, we can be assured that height is not redundant feature. The lesser accuracies of the above models show that Geometry is not sufficient to classify images into ad vs nonad categories. Hence, we need to try something more. For further experiments, we tried Neural Networks on the selected features. For subsequent experiments, we keep height in the dataset as it was proved non-redundant.

NN Model Performance				
S.No.	No. of Layers	Total Params	Train Acc	Test Acc
1	6	1186	0.8547	0.8527
2	6	3216	0.8611	0.8818
3	**5**	**709**	**0.8931**	**0.8861**

2) Rough Sets (Novelty): We use K-means Discretization Transform[9] technique for discretizing the continuous variables (i.e. height, width and aspect ratio).

A K-means discretization transform will attempt to fit k-clusters for each input variable and then assign each observation to a cluster.
The number of clusters is likely to be small, such as 3-to-5.

We use rough set based technique to prune the data without compromising much on the classificatory power and to generate a partial reduct.

This technique is efficient for algorithm design for identification of hidden patterns in the data. It evaluates the significance of each attribute in the form of dependence value γ.

$\gamma_P(Q) = \frac{\|POS_P(Q)\|}{\|U\|}$,
where U is the Universal Set. $\|POS_P(Q)\|$ is the P-Positive region of Q .

Discernibility matrices(Naive method) are easy to implement but require very high computational power and space complexity(NP Hard). So, we use approximate algorithm for the same.

Here, we use modified quick reduct algorithm to generate pseudo-reducts.
Stopping condition discernibility of the the pseudo-reduct is $\gamma_R < 0.96$.
Time Complexity = O(N^2)

$Modified\ quick\ reduct : -$
$C :\ set\ of\ all\ conditional\ attributes,\ here,\ 1528$
$D : decision\ attribute,\ here,\ ad/nonad$
$Output : R \leftarrow the\ attribute\ reduct\ R \subset C$
$R \leftarrow \{\}$
do
  $T \leftarrow R$
  $var_val \leftarrow 0$
  $var \leftarrow NULL$
  $for\ each\ \alpha \in (C - R)$
    $if(\gamma_{T \cup \alpha}(D) - \gamma_T(D) > var_val)$
      $var_val \leftarrow \gamma_{T \cup \alpha}(D) - \gamma_T(D)$
      $var \leftarrow \alpha$
  $R \leftarrow R \cup var$
$until\ \gamma_R < 0.96$
$return\ R$

We found that there were no duplicate rows in the new dataset composed of the selected features .

Some important observations from the Heat Map(fig 9) are mentioned as follows:-
Firstly, The correlation between origurl*home+html and origurl*home is quite high(0.84), implying they occur together in most of the samples. Hence, we keep origurl*home as it is more important than origurl*home+html.
Similarly, we remove url*keith+dumble, url*images+geoguideii,ancurl*familyid because of its high correlation value with ancurl*members+keith, url*geocities.com and ancurl*runid namely 0.65, 0.62, and 0.54 respectively.
Secondly, we observe that a single feature ancurl*ng has significant correlation with ancurl*click+runid, ancurl*familyid,ancurl*runid. Hence we remove ancurl*ng.

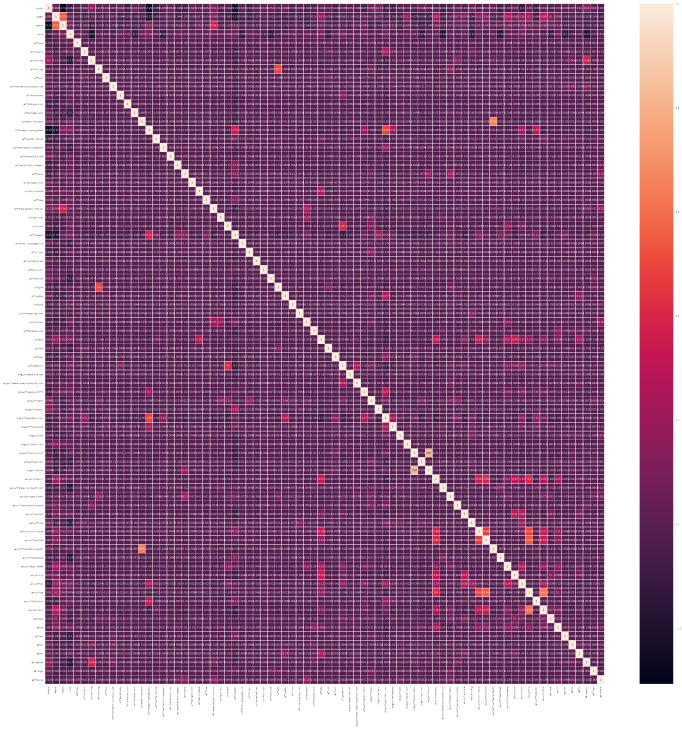


Fig. 9: Heat Map for Features Selected from Reducts

pseudo-reduct size after removing dependent attributes = 78 - 5 = 73

Model Performance		
Model	Accuracy	F1-Score
Support Vector Classifier	0.95	0.94
KNN classifier	0.96	0.95
Random Forest Classifier	0.98	0.97
XGB classifier	0.95	0.94
Logistic Regression	0.94	0.93
Decision Tree Classifier	0.95	0.94

Summary of various Neural network models defined on this dataset:-

NN Model Performance				
S.No.	No. of Layers	Total Params	Train Acc	Test Acc
1	4	3686	0.9808	0.9645
2	4	3321	0.9898	0.9708
3	**4**	**709**	**0.9895**	**0.9734**

3) Shapley Values: Shapley values[15,17] correspond to the contribution of each feature towards pushing the prediction away from the expected value. In other words, the Shapley value of a feature value is the average change in the prediction

that the coalition already in the room receives when the feature value joins them. We can't compute the exact shapley value of a feature due to high computational complexity, so we use the approximation with Monte-Carlo sampling.

$$\hat{\phi}_j = \frac{1}{M} \sum_{m=1}^{M} \left(\hat{f}(x_{+j}^m) - \hat{f}(x_{-j}^m) \right) \tag{1}$$

where $\hat{f}(x_{-j}^m)$ is the prediction for x, but using a random number of feature values except feature j and $\hat{f}(x_{+j}^m)$ is the prediction for x, but using the same random feature values used for $\hat{f}(x_{-j}^m)$ and the feature j. M is the number of different samples taken.

Algorithm 1 : *CSA Algorithm*
CSA algorithm(F, Δ, d, s) :
1 $selected \leftarrow \emptyset$
2. *for each* $f \in F/selected$
2.1 $C_f \leftarrow contribution(f, selected; d)$
3. *if* $max_f\, C_f > \Delta$
3.1 $selected \leftarrow selected \cup selection(\{C_f\}; s, \Delta)$
3.2 *goto* 2
else
3.3 *return selected*

The Contribution-Selection algorithm (CSA), described in detail above, is iterative in nature, and can either adopt a forward selection or backward elimination approach. Using the subroutine contribution, it ranks each feature according to its contribution value($\hat{\phi}_j$), and then selects s features with the highest contribution values with forward selection (using the sub-routine selection) or eliminates e features with the lowest contribution values with backward elimination (using elimination). It repeats the phases of calculating the contribution values of the remaining features given those already selected (or eliminated), and selecting (or eliminating) new features, until the contribution values of all candidate feature exceed a contribution threshold Δ with forward selection (or fall below a contribution threshold with backward elimination).

We plotted a bar diagram(fig- 10) and ranked the features based on their shapley values. We selected top 15 features among them.

From the heat map(fig- 11) of the selected 15 values, we want to infer some points. Firstly, width and aratio have correlation value equal to 0.57, which as explained above, is because of definition of aspect ratio. Secondly, 'ancurl*click' and 'ancurl*com' have significant correlation, this implies that among the collected samples, in the ancurl of most of the images, click and com appear together. One can say similar thing for the phrases 'click' and 'here to' in the alternate text based on their notable correlation.

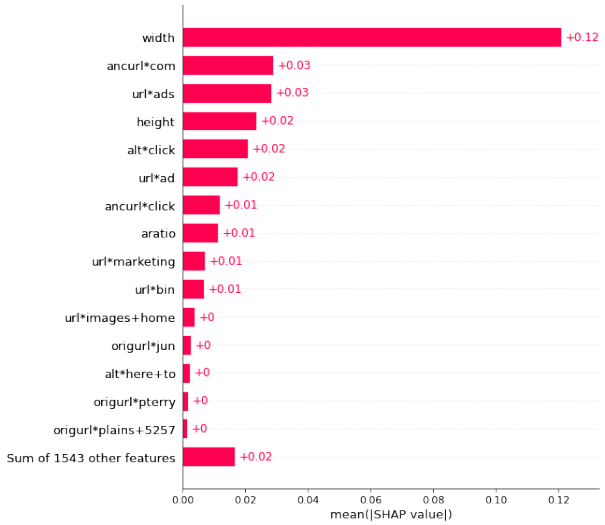


Fig. 10: Bar Plot Shapley Values

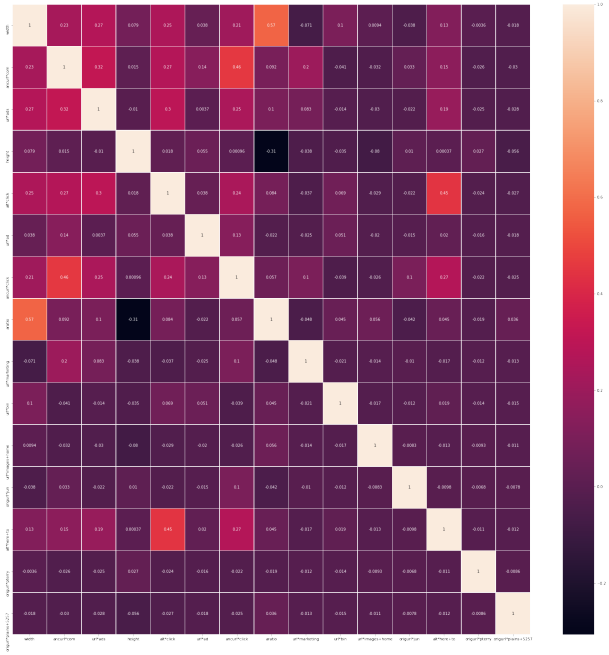


Fig. 11: Heat Map for features selected from shapley values

Tree based Model Performance		
Model	Accuracy	F1-Score
Support Vector Classifier	0.97	0.96
KNN classifier	0.95	0.94
Random Forest Classifier	**0.98**	**0.98**
XGB classifier	0.96	0.95
Logistic Regression	0.97	0.97
Decision Tree Classifier	0.98	0.97

Summary of various Neural network models defined on this dataset

NN Model Result				
S.No.	No. of layers	Total params	Train acc	Test Acc
1	4	30,283	0.9864	0.9720
2	4	32,451	0.9871	0.9708
3	3	47,411	0.9871	0.9732
4	**3**	**42,367**	**0.9874**	**0.9781**
The AUC for all of these models were almost 0.99				

C. Feature Extraction(Novelty)

As most of our features represent whether a particular phrase is there in the HTML document of the advertisement, we can lend some help from natural language processing to extract some new features. We only select features which mark words appearing in caption and alternate text.

We can generate scores based on their contextual similarity[11] with our context i.e. Internet Advertisements. But, a word can have different meanings based on the context. And to define the context, we need to create a dictionary of words which are related to the context.

Generating the Dictionary:-
We calculated the odd's ratio for caption and alt-text attributes. We selected words appearing in the features having high odd's ratio.

An odds ratio (OR) is a measure of association between an exposure and an outcome. The OR represents the odds that an outcome will occur given a particular exposure, compared to the odds of the outcome occurring in the absence of that exposure Formula to find odd's ratio for a particular feature.

Feature	ad	nonad
0	a	b
1	c	d

$$Odd's\ Ratio\ =\ (a/b)/(c/d)$$

We also added some common words/phrases related to Internet adverts in the dictionary.
Generating **nlp-score**:- We use Sentence transformer to convert the phrases/words into tensor embeddings(BERT embedding[10,11]) and then use cosine similarity. The BERT allow for obtaining word vectors that morph knowing their place and surroundings. BERT embeddings are contextual. Hence, they are more powerful than most of the techniques.

We used pretrained BERT-models[13] to encode the words into vectors and then used cosine similarity to find the similarity of each caption feature and alt-text feature with the above generated Dictionary. We store the mean of sum of the scores generated for each value as a new feature namely **nlp-score**.

Now with this newly created feature, we do a series of experiments. Let's observe its distribution plot (fig- 12) for ad and nonad.

From the distribution plot we can see, the density of samples with **nlp-score** near 0 are lesser for ad than nonad. Also, for ad samples, there are more samples in higher ranges as compared to those in nonad class. First among them are the Tree-based and other models. We only select features using their Shapley Values to measure the importance of the newly extracted feature.

One important observation is that **nlp-score** have significant contribution in decision making, its shapley value is 0.01!

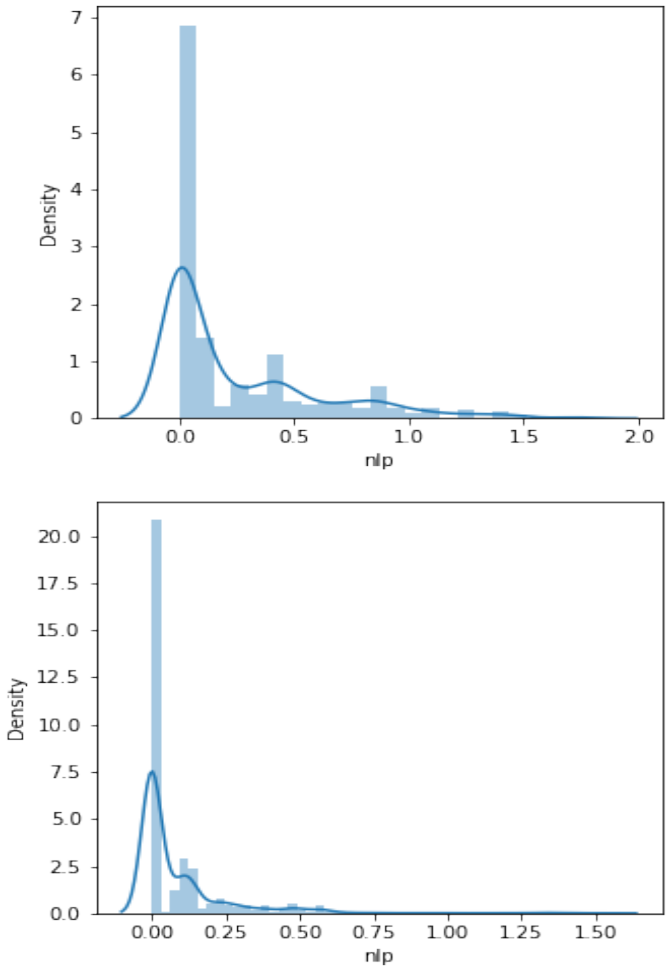


Fig. 12: Nlp-score distribution, Above: ad and below: nonad

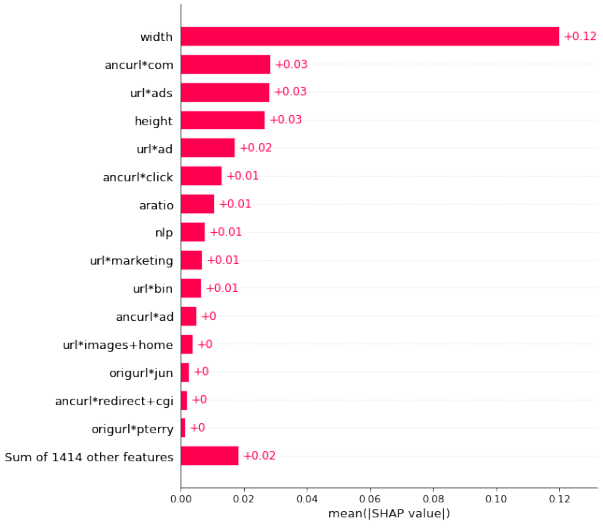


Fig. 13: Shapley Values with nlp-score

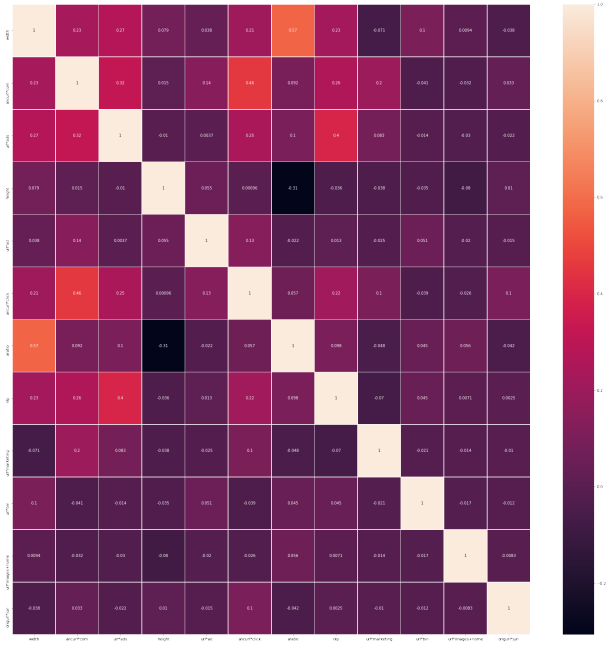


Fig. 14: Heat Map for Shapley with nlp-score

As all the correlation value in the heat-map (fig 14) are less than 0.5 except the width, aratio pair, we are not removing any features as they are not highly dependent. Now we present the results of the tree-based and and other models trained on these selected features.

Model Performance		
Model	Accuracy	F1-Score
Support Vector Classifier	0.97	0.96
KNN classifier	0.97	0.97
Random Forest Classifier	**0.99**	**0.99**
XGB classifier	0.96	0.95
Logistic Regression	0.97	0.97
Decision Tree Classifier	0.98	0.97

The results improved when we took **nlp-score** into consideration. As we can see, the Random Forest Classifier's accuracy is 0.99 along with equally good F1-Score.

Neural networks also gives improved results. We present the results below:

NN Model Result				
S.No.	No. of layers	Total params	Train acc	Test Acc
1	3	43,541	0.9895	0.9744
2	**3**	**43,477**	**0.9948**	**0.9875**
The AUC for all of these models were almost 0.99				

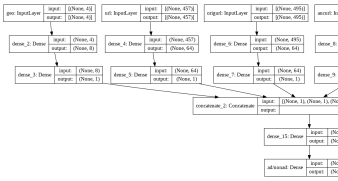
D. All Features

The fully connected neural net requires a lot of training time, so to boost the speed of the training we employ mixture of attribute based Neural experts models. They are as follows:-

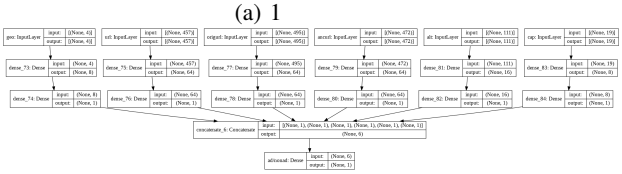
1) Neural Experts: When we observe our feature set, we may notice that most of the features describe only one property of the ad in the HTML document. For example, all the features that encode what is in the URL, basically describe the URL. Hence, we can divide the feature set into further groups, and further employ multiple neural networks to train them(called neural experts) and merge their results in another neural network. We Divide the Feature Set on 2 basis, described below:

- Grouping in 6 Major classes
  We have divided the feature set into 6 categories namely, geometry(includes local feature), image url, original url, anchor url, alternate text, caption text based on where the features were taken from the document.

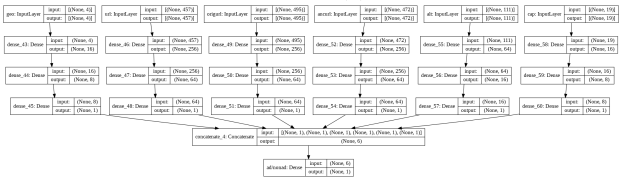
Neural Experts Model Performance			
S.No.	Figure number	Train Accuracy	Test Accuracy
1	15a	0.9753	0.9622
2	15b	0.9759	0.9562
3	15c	**0.9777**	**0.9647**



(a) 1



(b) 2



(c) 3

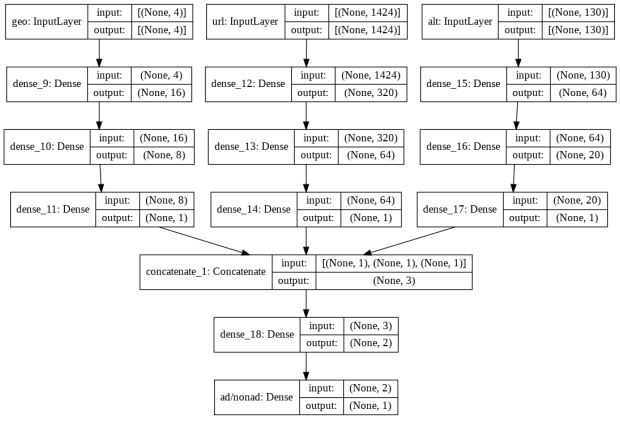
Fig. 15: Neural Experts with 6 groups

- Grouping in 3 Major Classes Here, we group the Features in terms that those what all they describe on a larger scale i.e. dividing them into url(includes URL, origurl, ancurl), Geometry and text(includes alt-text and captions).
  We present below the results of our Neural Experts:

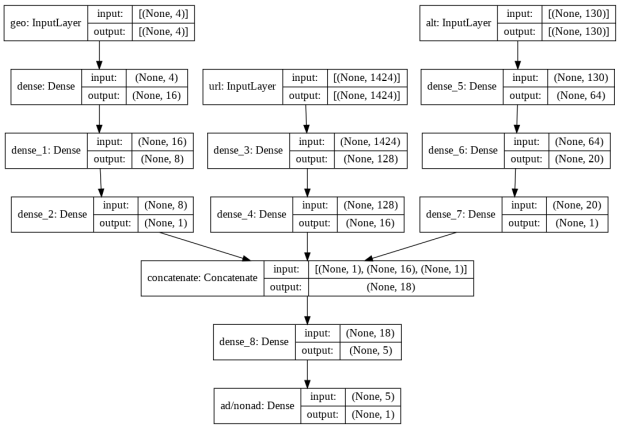
Neural Experts Model Performance			
S.No.	Figure name	Train Accuracy	Test Accuracy
1	16a	0.9805	0.9586
2	16b	0.9812 1	0.9635
3	16c	**0.9833**	**0.9672**

2) Neural Experts with NLP-scores(Novelty): In our subsequent analysis, we introduce nlp-scores in the feature set and remove all the caption and alt-text features. And again we group them into further classes

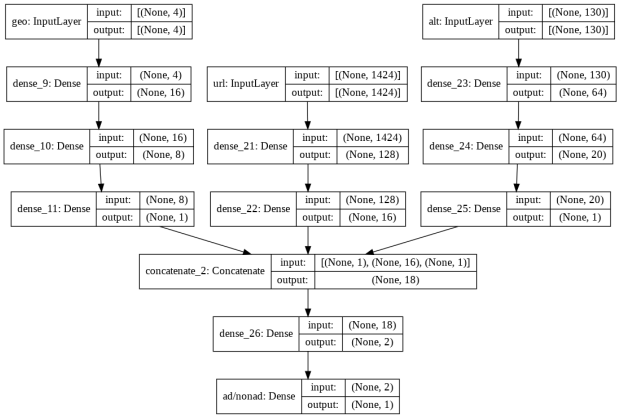
- Grouping in 5 Major classes



(a) 1



(b) 2

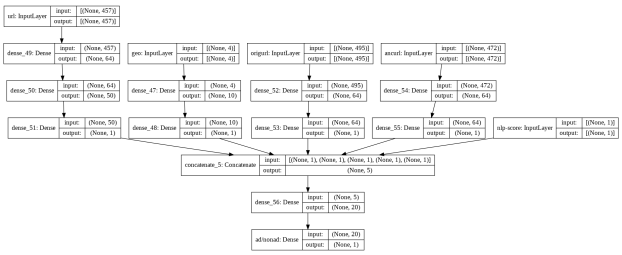


(c) 3

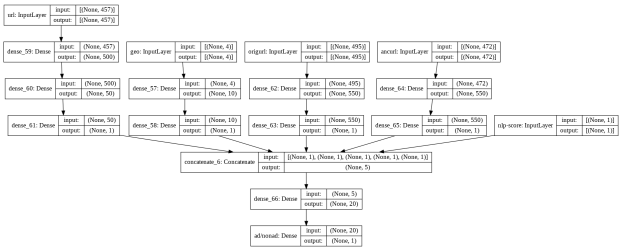
Fig. 16: Neural Experts with 3 groups

We have divided the feature set into 5 categories namely, geometry(includes local feature), image url, original url, anchor url, nlp-score.

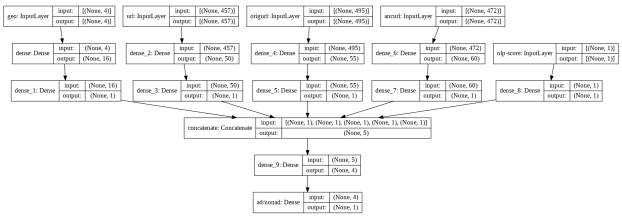
Neural Experts Model Performance			
S.No.	Total Parameters	Train Accuracy	Test Accuracy
1	17a	0.9753	0.9622
2	17c	0.9759	0.9562
3	**17b**	**0.9789**	**0.9659**



(a) 1



(b) 2

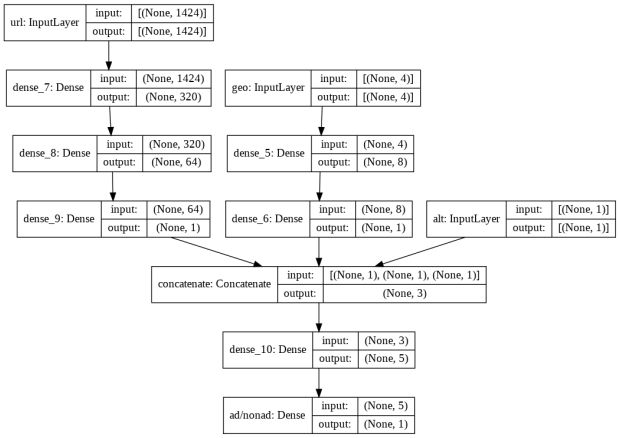


(c) 2

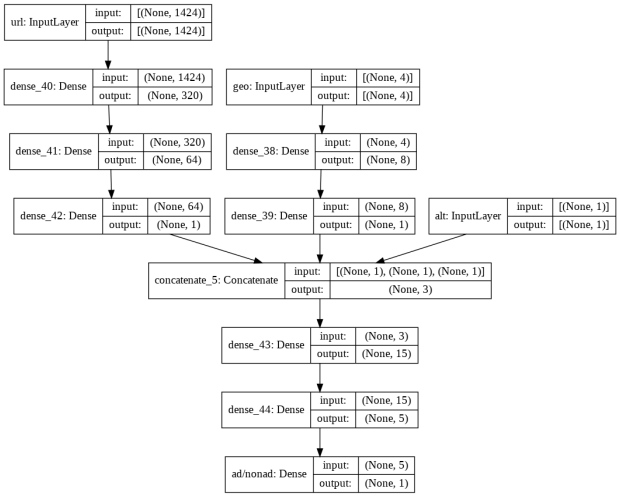
Fig. 17: neural experts 5 groups(nlp-score included)

- Grouping in 3 Major Classes Here, we group the Features in terms that those what all they describe on a larger scale i.e. dividing them into url(includes url, origurl, ancurl), Geometry and nlp-score. We present below the results of our Neural Experts:

Neural Experts Model Performance			
S.No.	Figure name	Train Accuracy	Test Accuracy
1	18a	0.9893	0.9682
2	18b	**0.9847**	**0.9730**



(a) 1



(b) 2

Fig. 18: Neural Experts Architecture with 3 groups and nlp-score included

VI. Models on Skewed Data

Due to time complexity bounds, we extracted the top 15 features using Shapley Values (fig- 19).

From the heat-map 20 we note the following :-
1. ancurl*click and ancurl*com are highly co-related so we remove ancurl*com
2. url*ad and url*doubleclick.net are highly co-related so we remove url*doubleclick.net
3. alt*click and alt*here+for are highly co-related so we remove alt*click

We have selected 12 (15 - 3) features out of 1558 features. Now we train different models on this dataset :-

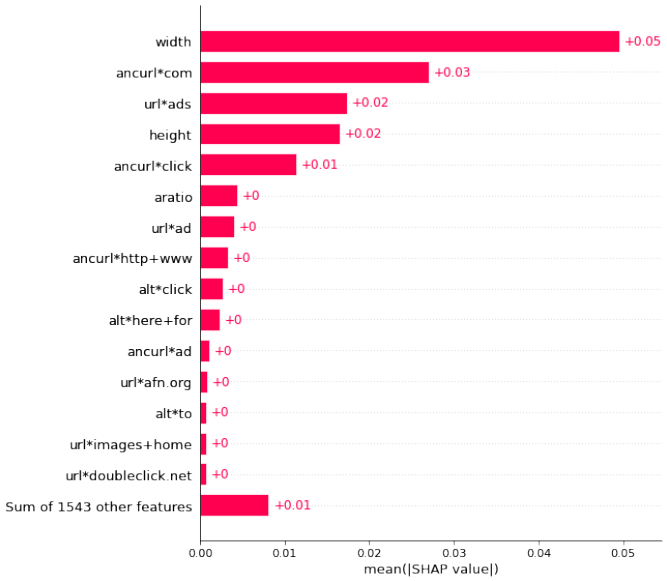


Fig. 19: Shapley Values for Skewed data

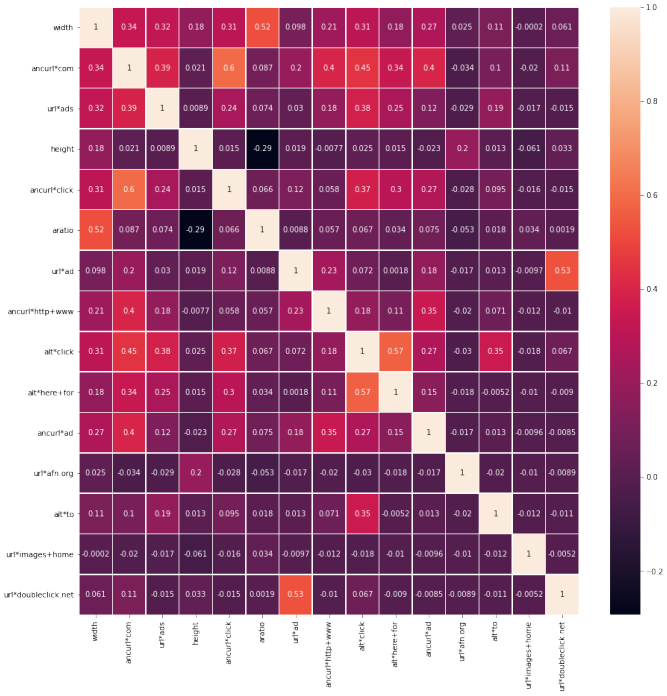


Fig. 20: Heat map of attributes selected using shapley values for skewed data

Model Performance		
Model	Accuracy	F1-Score
Support Vector Classifier	0.97	0.93
KNN classifier	0.97	0.94
Random Forest Classifier	**0.97**	**0.94**
XGB classifier	0.97	0.94
Logistic Regression	0.95	0.89
Decision Tree Classifier	0.97	0.94

The results show that we get good accuracy scores but the f1-scores are not very high. As we can see, the Random Forest Classifier (*accuracy is 0.97 and F1-Score 0.94*) performs the best.

NN Model Structure(Shapley Values)		
Layer Type	Shape	Parameter count
Dense	25	325
Dense	20	520
Dense	5	105
Dense	1	6
Note: Total trainable parameters = 956		

Neural Network Model Summary (feature selection technique : Shapley Values) with uniform class weights.

NN Model Result with uniform class-weights		
Train accuracy	Test Accuracy	validation Accuracy
0.9748	0.9634	0.9756
Train AUC	Test AUC	validation AUC
0.9539	0.9316	0.9583
Note: Layers count = 4		

Next we used cost effective learning to penalize the error on ad class more than non-ad class to have better AUC score. We chose the class weights to be
$class_w = \{0 : 8, 1 : 1\}$ We use the same model architecture as above for this model as well.

NN Model Result with different class-weights		
Train accuracy	Test Accuracy	validation Accuracy
0.9734	0.9634	0.9787
Train AUC	Test AUC	validation AUC
0.9736	0.9403	0.9689
Note: Layers count = 4		

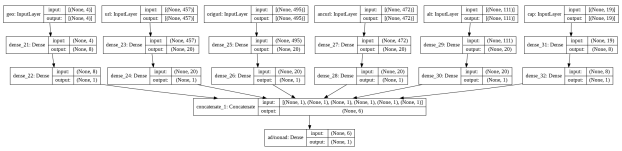
We observe that the test AUC increased from 0.9316 to 0.9403 by using different class weights.

Neural experts model Results				
No of groups	Figure-name	Train acc	Test Acc	validation Acc
6	21a	0.9858	0.9710	0.9634
3	21b	0.9922	0.9649	0.9604

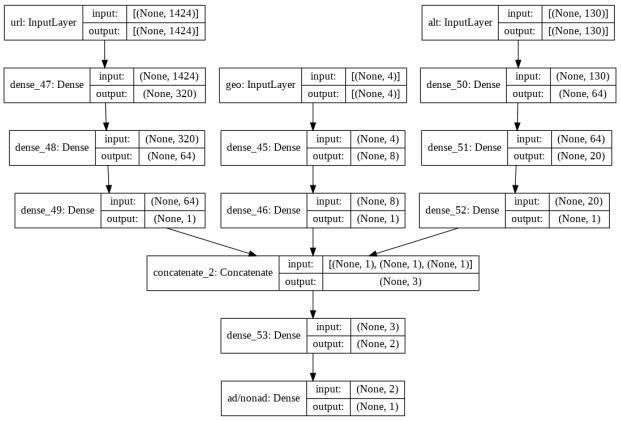
VII. Novelty Aspect of our Scheme

We implemented Modified quick reduct to select 78 features with the combined dependence value ¿0.96. The model outperformed the previously existing models trained on this dataset by other people and some other models we trained on the same. Most of the features in the dataset are binary, which motivated us to use the rough set theory presented in the class for feature selection and we got promising results.

Next, we created a new feature using BERT embeddings and cosine similarity. For This we selected those phrases occurring in alt-text/caption which have odds ratio ¿1 and some other custom words/phrases were added. This design of dictionary was important because of different context a word can be used



(a) 1



(b) 2

Fig. 21: Neural Experts Architecture on skewed data

in. using this dictionary, we created a new attribute called **nlp-score** and observed that its distribution for ad and non ad can explain the decision boundary very well, even better than others. The results with this feature were improved when we used top‘15 features ranked on the basis of their Shapley Values.

We replaced the caption and alt-text features with nlp-score attribute and trained Neural Experts models of two types:-
1. Number of groups = 5(geometry(includes local feature), image url, original url, anchor url, nlp-score)
2. Number of groups = 3(url(includes url, origurl, ancurl), Geometry and nlp-score)

Both these models outperformed the Neural Expert model using the initial features(without the nlp-score feature).

VIII. Results and conclusion

Now we compare all the various techniques of dimension reduction/feature selection/feature extraction with the performance of the best among the various models we trained so far.

Summary of Tree Based Models			
Techniques Used	Best Classifier	Accuracy	F1-score
Geometry	Random Forest Classifier	0.93	0.92
Skewed+Shapley Values	Random Forest Classifier	0.97	0.94
Random Projection	Random Forest Classifier	0.96	0.96
PCA	Random Forest Classifier	0.97	0.97
RoughSet Reducts	Random Forest Classifier	0.98	0.97
Shapley Values	Random Forest Classifier	0.98	0.98
NLP+Shapley	**Random Forest Classifier**	**0.99**	**0.99**
Note: Skewed data = missing values are imputed but has class imbalance.			

We conclude the following from the above table:

1) We can observe that in all the above data handling techniques, Random Forest outshines other models. It is well expected because the trees in Random Forest Classifier are not pruned and hence they present a better decision boundary.
2) Geometry features, in spite of having maximum variance, don't contain sufficient information to explain the decision boundary in ad vs nonad. Hence, we need to include some more features to have better classificatory power.
3) If we only select More important features for classification using shapley values, even for the skewed data set we get higher accuracy with a significantly better F1-Score.
4) Random Projection is much faster than PCA for same value of reduced dimension, but both the methods naively reduce the dimensions leading to loss of interpretability in the results.
5) Feature Selection using Rough Set, namely pseudo-reduct, give much better results than the pre-existing models. The dependence value chosen for the reduct was very high (0.96), the discern-ability using the selected features is very high and would be sufficient to explain the decision boundary.
6) As evident from the above table, most of the models gives exceedingly high accuracy and F1 score when trained on features selected by Shapley Values. Shapley Values ranks the features according to their importance and hence presents excellent scores.
7) The models trained on extracted feature, **nlp-score**, along with feature selection using Shapley Values gives the best results. We replaced all the caption and alt-text features with the **nlp-score** attribute. We can conclude from this that many features among these text-attributes were redundant.

Next we present the Neural Networks model performance with these techniques. For each technique, the neural networks are the ones which are tuned perfectly and best results.

Neural Network Performance Summary		
Techniques Used	Train acc	Test acc
Skewed+shapley(uniform class weights)	0.9748	0.9634
Skewed+shapley(non-uniform class weights 1:8 = nonad:ad)	0.9734	0.9634
PCA	0.9686	0.9645
Random Projection	0.9928	0.9708
Geometry	0.8931	0.8861
Rough Sets and Reducts	0.9895	0.9734
Shapley Values	0.9874	0.9781
NLP+Shapley Values	0.9948	0.9875
Skewed+Neural Expert 6 groups	0.9858	0.9710
Skewed+Neural Expert 3 groups	0.9922	0.9649
Neural expert 6 groups	0.9777	0.9647
Neural expert 3 groups	0.9833	0.9672
Neural expert with nlp score with 5 groups	0.9789	0.9659
Neural expert with nlp score 3 groups	0.9847	0.9730

We conclude the following from the above table:

1) The attribute based neural experts gave significantly good results with lesser time even when trained for 1000 epochs.
2) With the help of Shapley values, we were got promising results for neural networks.
3) We observe that by using nlp-score in place of caption features and alternate text give better results(On the basis of accuracy).
4) We observe that neural expert models do not compromise on test accuracy so we can be assured that the Neural expert is able to learn the pattern in the data.
5) The best Neural network model trained so far was obtained by nlp-score and features selected using Shapley values.
6) The class imbalance is very well handled by cost sensitive learning.

Based on all the models we trained, we conclude the following about the best model for this dataset:-
1. Classifier = Random Forest Classifier
2. Feature extraction = nlp-score from the caption text and the alternate text.
3. Feature selection = Shapley values (Top 15 features)
Best Model performance :-
accuracy = 0.99
f1-score = 0.99

Correlation of various attributes

From the Heat Map(fig 9) we can infer the following:-
Firstly, The correlation between origurl*home+html and origurl*home is quite high(0.84), implying they occur together in most of the samples. Hence, we keep origurl*home as it is more important than origurl*home+html.
Similarly, we remove url*keith+dumble, url*images+geoguideii,ancurl*familyid because of its

high correlation value with ancurl*members+keith, url*geocities.com and ancurl*runid namely 0.65, 0.62, and 0.54 respectively.
Secondly, we observe that a single feature ancurl*ng has significant correlation with ancurl*click+runid, ancurl*familyid,ancurl*runid. Hence we remove ancurl*ng.

From the heat map(fig- 11) of the selected 15 values, we can infer that:-
Firstly, width and aratio have correlation value equal to 0.57, which as explained above, is because of definition of aspect ratio.
Secondly, 'ancurl*click' and 'ancurl*com' have significant correlation, this implies that among the collected samples, in the ancurl of most of the images, click and com appear together. One can say similar thing for the phrases 'click' and 'here to' in the alternate text based on their notable correlation.

From the heat map (14)nlp and url*ad have a correlation coefficient of 0.4 , so they are moderately correlated. The feature nlp's correlation with other features is very low.

IX. Codes

The codes for all the python notebooks and all the saved neural networks can be found here.

X. References

[1]N. Kushmerick (1999). Internet Advertisement Datasets, UCI Machine Learning Repository
[2] Olga Troyanskaya, Michael Cantor, Gavin Sherlock, Pat Brown, Trevor Hastie, Robert Tibshirani, David Botstein, Russ B. Altman, Missing value estimation methods for DNA microarrays , Bioinformatics, Volume 17, Issue 6, June 2001, Pages 520–525
[3] Liu, F. T., Ting, K. M., and Zhou, Z. H. (2008, December). Isolation forest. In Data Mining, 2008. ICDM'08. Eighth IEEE International Conference on (pp. 413–422). IEEE
[4] Principal component analysis: a review and recent developments,an T. Jolliffe and Jorge Cadima
[5] Experiments with Random Projections for Machine Learning,Dmitriy Fradkin and David Madigan
[6]A survey of dimensionality reduction techniques based on random projection, Haozhe Xie, Jie Li, Hanqing Xue
[7] Rough Set Theory in Decision Support Systems, Agnieszka Nowak - Brzezinska
[8] S. Dev, F. M. Savoy, Y. H. Lee, S. Winkler, Rough-set Based Visible Color Channel Selection, IEEE Geoscience and Remote Sensing Letters, accepted, 2016
[9]Discrete Transforms Tutorial
[10]BoW to BERT,Ashok Chilakapati
[11]Text Similarities : Estimate the degree of similarity between two texts,Adrien Sieg
[12]Alvarez, S. "Mining over loosely coupled data sources using neural experts." (2003).
[13]Pretrained BERT Models
[14]Cost-Sensitive Learning for Imbalanced Classification, Jason Brownlee
[15]Shay Cohen, Eytan Ruppin, Gideon Dror, Feature Selection Based on the Shapley Values, Conference: IJCAI-05, Proceedings of the Nineteenth International Joint Conference on Artificial Intelligence, Edinburgh, Scotland, UK, July 30-August 5, 2005
[16] Nguyen Hien M.,Cooper Eric W.,Kamei Katsuari Borderline Over-sampling for Imbalanced Data Classification, Proceedings : Fifth International Workshop on Computational Intelligence and Applications, volume2009, issue1,Novem 2009, Pages 24-29
[17]Molnar, Christoph. "Interpretable machine learning. A Guide for Making Black Box Models Explainable", 2019. https://christophm.github.io/interpretable-ml-book/.