



THE EFFECT OF PROMINENCE IN VISUAL OBJECTS TO PREDICT CLICK-THROUGH RATE IN CREATIVES

DATA-BASED DECISION-MAKING IN
ADVERTISING DESIGN

LAURENS MATTHEW JOHN HENDRIKX

THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF SCIENCE IN DATA SCIENCE & SOCIETY
AT THE SCHOOL OF HUMANITIES AND DIGITAL SCIENCES
OF TILBURG UNIVERSITY

STUDENT NUMBER

2084481

COMMITTEE

dr. Mirella De Sisto
dr. Nevena Rankovic

LOCATION

Tilburg University
School of Humanities and Digital Sciences
Department of Cognitive Science &
Artificial Intelligence
Tilburg, The Netherlands

DATE

December 15, 2022

WORD COUNT

7106

ACKNOWLEDGMENTS

I would like to express my gratitude to my primary supervisor, Mirella de Sisto, who guided me throughout this project. I would also like to give a special thanks to my company supervisor, Davy van der Horst, for his knowledge and the time he took for guiding me throughout the thesis. This thesis is especially dedicated to my mother, Kimberly Ballenger, I know she would be proud.

DATA SOURCE / CODE / ETHICS STATEMENT

Work on this thesis did not involve collecting data from human participants or animals. The original owner of the data and code used in this thesis retains ownership of the data and code during and after the completion of this thesis. The author of this thesis acknowledges that they do not have any legal claim to this data. Data supporting this thesis are available from Springbok Agency.

CONTENTS

1	Introduction	3
1.1	Problem Statement	4
1.2	Research Questions and Strategy	5
2	Literature Review	6
2.1	Predictive Modeling in Click-Through Rate	6
2.1.1	Multivariate Statistical Models	7
2.1.2	Factorization Machines	8
2.1.3	Deep Learning Models	8
2.1.4	Decision Tree Models	8
2.1.5	Support Vector Machine	9
2.2	Predictive Modeling in Advertising Design	9
3	Methodology and Experimental Setup	10
3.1	Data Description	10
3.2	Object Extraction	10
3.3	Data Pre-processing	10
3.3.1	Variable Construction and Transformations	11
3.3.2	Categorical Predictors Transformation	12
3.3.3	Polynomial Predictors Transformation	13
3.3.4	Continuous Variables Normalization	13
3.4	Algorithms	14
3.4.1	Ordinary Least Squares (OLS)	14
3.4.2	Extreme Gradient Boosting (XGBoost)	15
3.4.3	Support Vector Regression (SVR)	16
3.4.4	Predictors' Importance	16
3.5	Training, Validation, and Testing Procedure	17
3.6	Performance Metrics	18
4	Results	18
4.1	Hyperparameter Tuning and Cross-Validation Results (Without Prominence)	18
4.1.1	Ordinary Least Squares with Regularization	18
4.1.2	XGBoost	19
4.1.3	Support Vector Regressor	19
4.1.4	Baseline Ordinary Least Squares	20
4.2	Performance on Test Data (Without Prominence)	20
4.3	Hyperparameter Tuning and Cross-Validation Results (With Prominence)	21
4.3.1	Ordinary Least Squares with Regularization	21
4.3.2	XGBoost	22
4.3.3	Support Vector Regressor	22
4.3.4	Baseline Ordinary Least Squares	23

4.4	Performance on Test Data (With Prominence)	23
4.5	Predictors' Performance	24
5	Discussion	26
5.1	Results Discussion	26
5.2	Limitations	28
5.3	Future Research	28
6	Conclusion	28

THE EFFECT OF PROMINENCE IN VISUAL OBJECTS TO PREDICT CLICK-THROUGH RATE IN CREATIVES

DATA-BASED DECISION-MAKING IN ADVERTISING DESIGN

LAURENS MATTHEW JOHN HENDRIKX

Abstract

Advertising creatives generate a significant source of revenue within the marketing ecosystem. It has been proven that visually appealing ads generate more desirable responses like clicks and conversions. A team of content designers make decisions on what visually appealing features to include in these creatives. These decisions are usually contingent on intuition, experience and a good eye for design. As advertising design is a highly subjective and creative industry, there is a lot of room for human error due to taste and preference. Within the domain of advertising design, data-driven approaches are an under-researched topic. Prominence is an important visual design feature which describes the relative size of an object within a creative. It is used to determine the visual weight of an element. The higher the prominence the higher the visual weight. Although proven to be a significant feature in advertising design and having contribution to a visually appealing creative, this has never been tested in relation with predicting click-through rate. This thesis therefore aims to contribute to the knowledge base by researching the visual feature prominence and its effect on predicting click-through rate. The research uses a 5-fold cross validation to compare three regression models with the baseline. The results demonstrate that adding prominence as a variable leads to an improved predictive performance of 26% compared to that of the baseline. The study is however limited to the size of the dataset and should be extended in the domain of other industries.

1 INTRODUCTION

Online advertising has been increasingly important over the last decades. With digital banner advertisements (from now on: ‘creatives’) spending from €146,12 billion in 2022 to a projected spend of €234,53 billion in

2027 (Statista, 2022), it is important for advertisers to allocate their budget effectively. Although advertisers spend a lot of this marketing budget on targeting and attribution, 75% of a creative's impact is related to its visual quality (Minnium, 2021). Regardless of these statistics, marketers still spend most of their time and budget on when and where these creatives will run instead of its content. The reason for this is not because marketers have the wrong focus, but rather how difficult it is to quantify a creative and give evaluations on a scalable level.

1.1 Problem Statement

A creative is the advertisement served to users in a digital environment. By showing these creatives, advertisers aim to either promote brand awareness and/or receive desirable responses like click-through rate (CTR) or conversion (Azimi et al., 2012). The most used metric for evaluating the success of a creative is CTR (Yang & Zhai, 2022). It describes the number of clicks your ad receives divided by the number of times your ad is shown. Generally, a high CTR is a good indication that users find your ads attractive. Average click-through rates differ widely between media channels and industries. Display ads hold an average of 0.57% across industries, so there is a lot of room for improvement (CXL, n.d.).

Visual quality relates to the content of a creative. These include visual features like objects, typography, colors, shapes, logos, and rhetoric that work together in order to construct a visually appealing creative (Pieters, Wedel, & Zhang, 2007). A creative team is responsible for decisions based on the design of a creative. Decisions on this design are mostly led by intuition, experience, creativity, and a good eye for design. Advertising design is a very subjective and creative industry where anyone can form its opinion on design decisions. These decisions are subject to human error as there is no data-based approach to this. There is therefore a need to quantify creatives and make decisions based on data rather than on intuition.

Several studies, although limited, have researched the domain of advertising design in relation to that of marketing analytics and all proved significant in their own ways. These studies have researched interaction effects of visual elements (Li, Wu, Cui, & Zhang, 2021), (Yu et al., 2020a), (Wu et al., 2020), visual rhetoric (Ye et al., 2019), dominant colors and filters (Azimi et al., 2012). However, although proven a highly important feature in advertising design (Landa, 2022), (Bradley, 2015), the prominence of an object within a creative has never been tested in relation to CTR.

Prominence is a visual feature that describes the object's likelihood of being seen. It is a ratio of the object's size relative to that of the creative.

This feature is used in advertising design to make an object stand out and creates a sense of visual weight and composition. Visual weight refers to the force where the eye is most attracted to. This can be achieved by using contrast in colors and size composition. A large red element will create more visual weight than a small gray one. The sum of these characteristics is what determines partly what is visually appealing to the viewer of the creative, which can result in a click.

Research in prominence in relation with marketing analytics has mainly focused on the visibility of a creative relative to that of the search page. A creative that is positioned at the top of a page is more prominent and will generate on average more clicks than the ones positioned lower (Xu, Manavoglu, & Cantu-Paz, 2010), (Yang, Li, Zeng, & Jansen, 2018). However, prominence has never been researched in the context of the relative size of a visual object in a creative to its marketing performance.

This study therefore attempts to contribute to the knowledge base by being the first that adapts a machine learning approach to predicting CTR using prominence. By capturing the influence of a variable that has never been tested before it can help suggest improvements to already existing CTR prediction models. Also, the effect of prominence can be extended to other domains besides that of profitable companies. Another benefit is that it provides useful recommendations to designers in guiding their decisions in what features to consider. Instead of using intuition, decisions will be based on data, which makes it easier to explain to their higher-ups. Lastly, the prominence and the regression methods developed here can be used in addition to investigate other features in advertising design such as colors, position and text.

1.2 Research Questions and Strategy

The broad research question of the study is:

How accurately can we predict click-through rate by using the prominence and the objects detected in a creative with machine learning algorithms?

This research question can be expanded in the following sub-questions:

- RQ1 *To what extent can we predict click-through rate based on the objects detected within a creative?*
- RQ2 *Does adding prominence lead to more predictive power?*
- RQ3 *Do the machine learning algorithms outperform the baseline model in predicting click-through rate?*

The approach consists of three main steps. First, features will be extracted from the creatives. These features consist of objects that are shown within the creative. From these objects, a prominence score relative to the full size of the creative will be calculated per element. Then, three regression approaches will be used to predict CTR based on the extracted features. A 5-fold cross validation will be used to compare the models, one with prominence and one without. These models will be hyperparameter tuned to make them perform better. Third, the models will be evaluated and compared to the baseline model and conclusions will be made on what the effect of prominence is on predicting CTR. Also, the predictors' importance score will be used to show what variables have the most effect on predicting CTR.

The chosen models for this study are:

- Baseline The baseline model will use the Ordinary Least Squared regression that excludes feature interactions, regularization, and hyperparameter tuning.
- Model A Model A uses the Ordinary Least Squares regression that includes feature interactions, regularization, and hyperparameter tuning.
- Model B Model B uses XGBoost regression that includes feature interactions and hyperparameter tuning.
- Model C Model C uses Support Vector Regression that includes feature interactions and hyperparameter tuning.

2 LITERATURE REVIEW

The literature review is divided in two subsections. The first one, *Predictive Modeling in Click-Through Rate*, is devoted to reviewing studies in predicting CTR while the second, *Predictive Modeling in Advertising Design*, specifically covers theoretical frameworks and empirical studies in the field of advertising design.

2.1 *Predictive Modeling in Click-Through Rate*

CTR prediction is a data science problem that many advertisers are facing. The meta-analytical study of [Yang and Zhai \(2022\)](#) has summarized a total of 175 different articles related to CTR prediction. Kaggle has also published multiple CTR-prediction challenges that have resulted in the configuration of new machine learning algorithms. The task is defined as a supervised learning problem in which a continuous variable, CTR,

will be predicted based on a certain set of features X . The state-of-the-art CTR prediction models reported in the literature can be subdivided into five main categories: Multivariate Statistical Models, Factorization Machines (FMs), Deep Learning Models, Tree Models, and Support Vector Machines (Yang & Zhai, 2022). In the following section, we will decompose the models in terms of advantages and disadvantages and review its performances on CTR prediction.

2.1.1 Multivariate Statistical Models

Ordinary Least Squares (OLS) is the simplest model for predicting CTR. It generates a best-fit line to the independent variables based on a continuous target that minimizes the sum of squared errors (Zdaniuk, 2014). The most common method that utilizes OLS is linear regression. OLS with multiple independent variables takes the form of:

$$y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \dots + \beta_p X_p \quad (1)$$

where β_p describes the coefficients (weights) of variables X_p . It is commonly used in CTR prediction as the baseline model in performance evaluation (Azimi et al., 2012), (Guo, Tang, Ye, Li, & He, 2017), (Z. Wang, Yu, Shen, & Hu, 2016). The advantages of this model lay in its interpretability and simplicity. If we are interested in statistical inference, this model provides a solution as it is easy to interpret. However, linear regression usually does not perform well as it does not take into account the possibility of interaction between variables. An interaction effect exists when the effect of an independent variable on a dependent variable changes, depending on the value of another independent variable (Molnar, 2022). Interactions have been proven important in CTR prediction. The majority of the predictive models use a two-order or higher-order feature interaction. A two-order interaction means that there are two independent features interacting with each other. A higher-order feature interaction, is when there are more than two features interacting, however, these are more difficult to interpret.

To model a two-order interaction, the Degree-2 Polynomial (Poly2) model has been introduced. It differentiates with the OLS model in that it adds an interaction term for each feature pair. Poly2 takes the form of:

$$y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_1 X_2 \quad (2)$$

By not only considering the main effect of independent variables, polynomial interactions can capture relationships between features which can improve model performance. Compared to linear models, Poly2 performs better in CTR prediction in terms of MSE (Chang, Hsieh, Chang, Ringgaard, & Lin, 2010). However, feature interactions in Poly2 have a $O(n^2)$ model

complexity. When many features are tested within a research, we now require a lot of time and memory for training the model. In addition, when a categorical variable has high cardinality, it might have issues in capturing the feature interaction due to sparsity issues (Ren, Zhang, & Liang, 2020).

2.1.2 Factorization Machines

In order to reduce the problem that sparsity introduces, researchers have used Factorization Machines (FMs). FMs have won multiple CTR prediction challenges on Kaggle (Juan, Zhuang, Chin, & Lin, 2016). It is an extension of the Poly2 model where it is designed to capture interactions between features in high dimensional sparse datasets. It is formulated as a combination between a linear model and matrix factorization. When compared to Poly2, the weights of each interaction in FM is estimated using the inner product of the two latent vectors. Compared to OLS and Poly2, FMs perform better where the data is very sparse and you are using categorical variables with high cardinality.

FMs take the form of:

$$y = \beta_0 + \sum \beta_i X_i + \langle v_i, v_j \rangle x_i x_j \quad (3)$$

2.1.3 Deep Learning Models

The modeling frameworks previously presented involve one or two-order feature interactions. Although low-order interaction models are easy to implement and interpret, they suffer from under-performance when feature interactions are complex (Yan, Chen, Wan, & Wang, 2021). Deep learning models can capture these high-order feature interactions and have been gaining ever increasing performance in predicting CTR (Cheng et al., 2012), (Ouyang et al., 2019), (Jiang et al., 2018). However, these deep neural networks suffer from drawbacks in terms of interpretability and reliability (Molnar, 2020). Decision-making process in terms of what features contribute to a higher CTR are hard to be reasonably explained since the weights and activations of the neural networks are usually deemed unexplainable (Yu et al., 2020b). It also requires very large amount of data in order to perform better than other techniques.

2.1.4 Decision Tree Models

Tree models have been proven successful in predicting CTR, especially Extreme Gradient Boosting (XGBoost) (T. Chen & Guestrin, 2016), (Ke et al., 2017), (Livne et al., 2020). Gradient boosting machines use an ensemble method of combining a large number of relatively weak decision trees to obtain a stronger ensemble prediction (Natekin & Knoll, 2013). In XGBoost,

feature selection and interactions are automatically performed to generate new discrete feature vectors. The depth of the decision tree determines the order of the feature interactions (An & Ren, 2020). XGBoost is sparsity-aware but under-performs in cases with very high feature dimensions and big data (Yang & Zhai, 2022).

2.1.5 Support Vector Machine

Support Vector Machines (SVMs) are a set of supervised learning methods used for classification, regression, and outlier detection. Support Vector Regression (SVR) has been used in multiple CTR prediction papers (F. Wang, Suphamitmongkol, & Wang, 2013), (Azimi et al., 2012) where it outperformed other baseline models. It configures a hyperplane that can either be linear or non-linear. The model proves to perform well in high dimensional spaces. It is effective where the number of dimensions is greater than then the number of samples (Pedregosa et al., 2011). However, it is not suitable for large data sets.

2.2 Predictive Modeling in Advertising Design

Quantifying creatives and predicting its performance based on its visual design is a highly under-researched problem. Contributions have been made by Azimi et al. (2012) where it introduces 43 visual features and evaluates its effect on CTR based on three regression models. Among others, the study found that creatives with high contrast between visual elements perform better. Also, it found that a highly cluttered number of objects and text within a creative are undesirable. The Support Vector Regression model proved to have the best results in this study. A different approach was taken in a study that aimed to quantify visual rhetoric¹ of a creative in relation to marketing effectiveness (Ye et al., 2019), (Chakrabarti, Agarwal, & Josifovski, 2008). They show that improving the creative's content's relevancy to the webpage is more important than considering the content of the creative by themselves. Many studies have also researched interaction effects between visual elements and CTR prediction. In the study of J. Chen, Sun, Li, Lu, and Hua (2016), they found, for instance, that having one person in an advertisement will have better performance than having multiple persons. The study of Steen (n.d.) has also made significant contributions to the relation of advertising design and marketing performances. Among others, it found that large ads perform better than

¹ Visual rhetoric refers to how we are persuaded by the things we see. It includes visual elements and the context of these elements to persuade a reader to action (Newbold, 2017)

small ads. Also, it has proven that including a logo has a positive effect on the CTR.

3 METHODOLOGY AND EXPERIMENTAL SETUP

3.1 *Data Description*

The data for this study contains two groups of information. First, it contains the creatives where the objects and prominence will be extracted from. Second, it contains the marketing performances of these creatives to see how well they have performed. The raw data set is collected through Google Cloud BigQuery and contains real company data from a supermarket. Google BigQuery is a tool that allows you to query data from your cloud warehouse storage. The initial query retrieved 69,362 rows of data. Data of creatives has only been retrieved where there was at least 1 click and more than 50 impressions. After aggregating the data per creative, and not querying rows with missing values, data of 416 unique creatives were retrieved with no missing values.

Within the data, each creative has a unique URL that leads to the visual creative. These URLs have been scraped and the output has been joined with the queried marketing performances. The next step in the data mining pipeline is to extract the objects from the creative.

3.2 *Object Extraction*

The Google Cloud Vision API will be used to extract visual objects from the creatives. Developed by Google, it is a very powerful AI cloud service that uses pre-trained models to extract visual features from an image. Within this model, more than a million different categories are stored that can recognize objects. Some use cases of the API are automating the tracking of explicit content, identifying endangered species, detect human emotion, text recognition and many more. The features it can extract include object labelling, face and landmark detection, text recognition and retrieving dominant colors. The tool will be used to extract the objects from a creative which will output it into .CSV file. Figure 1 shows how Cloud Vision can be used to detect the fruits within a creative.

3.3 *Data Pre-processing*

This section describes the data cleaning and pre-processing pipeline. The output from the Cloud Vision is joined with its corresponding marketing

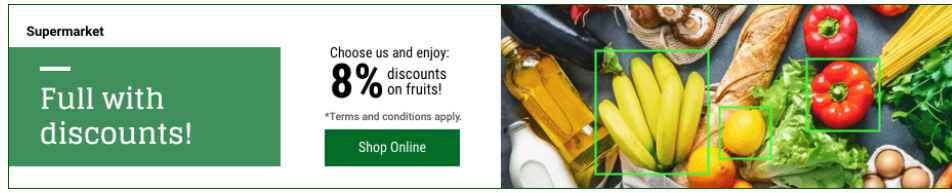


Figure 1: Objects detected by Google Cloud Vision: Banana, Citrus, Paprika.

performances using the `file_name` variable. The raw data set contains 1205 rows, which has increased since that a unique creative can have multiple objects. Variable selection has been done in the querying phase where they have been selected based on analytical performance. It has filtered out variables that are unrelated to the domains of advertising design or marketing analytics. The limitations of of this approach are discussed in section 5.2. The final raw dataset comprises of 6 variables: 1 ID variable, 2 variables used to construct the target variable, 3 predictor variables, and 1 target variable. The data doesn't contain any missing values.



Figure 2: (X,Y) Coordinates of Detected Object

3.3.1 Variable Construction and Transformations

Constructed variables:

- The "prominence" variable has been constructed by using the "vertex" variable. Vertex contains four tuples of x and y coordinates. These vertices show the bounding box around an object that the Cloud Vision tool has extracted relative to the overall advertisement size, see Figure 1. For the example in Figure 2, the formula of length * width will be made: $(0.80 - 0.30) * (0.70 - 0.60) = 0.05$. To create a percentage of this variable we multiply it by 100 to get a 5% prominence relative to the whole surface of the creative. In the dataset, prominence is referred to as area_ratio.
- The target variable "ctr" has been constructed from the "impressions" and "clicks" variable. The formula for the CTR is clicks / impressions * 100.

See Table 1 for the full descriptions of the dataset after the variable construction.

Table 1: Feature Description of Dataset

Feature	Description	Data Type	Example
file_name	Unique file name	object	creative_1
number_objects	Number of objects detected per creative	int64	3
object	Object detected	object	Person
area_ratio	Bounding box for location of object	float64	0.10
impressions	Number of impressions	int64	10.000
clicks	Number of clicks	int64	50
ctr	Click-through rate	float64	0.05

3.3.2 Categorical Predictors Transformation

The categorical variables that will be treated belong to the variable "object". This feature describes which object Cloud Vision has recognized within the creative. The variable has a cardinality of 62, which means it contains 62 different objects. Encoding this feature as it is would make the data set very sparse. Sparsity is a problem in which a data set consists of many zero's and a single one labeling the category. To solve this issue, infrequent categories of each nominal variable that constitutes less than 5 counts were combined into one artificial category called "other". This affected 47 variables nominal predictors out of 62 present in the data.

Then, all categorical predictors were transformed into a set of dummy variables (in machine learning terminology this is known as one-hot encoding). Transforming these categorical variables into a numeric value is necessary for running the algorithms used in this research.

3.3.3 Polynomial Predictors Transformation

From the *sklearn* package, the module *PolynomialFeatures*, has been used to engineer new features that have an interaction effect with each other. The feature pairs of *area_ratio* in conjunction with each individual object has been created. These engineered features will only be used in Model A, where the OLS model is used with regularization and interaction terms. Manually engineering these features will raise the number of input variables that can help better expose the important relationships between the input and target variables. A total of 136 extra columns have been added due to the interaction effects with each variable. The Baseline model will not use these newly engineered variables. Model B, which uses the XGBoost model, already has an interaction term incorporated in it that is dependent on the depth of the tree. Model C, which uses the Support Vector Regressor, also has includes a hyperparameter for including interactions terms. Table 2 and Table 3 shows how this interaction term works.

Table 2: Features Before Polynomial Transformation

area_ratio	packaged_goods	person	flower
0.10	1	0	0
0.20	0	1	0
0.15	0	0	1

Table 3: Features After Polynomial Transformation

area_ratio	area_ratio:packaged_goods	area_ratio:person	area_ratio:flower
0.10	0.1	0	0
0.20	0	0.2	0
0.15	0	0	0.15

3.3.4 Continuous Variables Normalization

All continuous predictors and target variable, including newly created one-hot encoded variables, have been min-max normalized. The goal of min-max normalization is to make every data point have the same scale so each feature is equally important. For every feature, the minimum value of that feature gets transformed into a 0 and the maximum value into a 1, and every other value gets transformed into a decimal between 0 and 1

(Akanbi, Amiri, & Fazeldehkordi, 2015). Tree-based models, like XGBoost, are usually not dependent on scaling, but non-tree models like OLS and SVM are. The formula for this min-max normalization is:

$$x_{scaled} = \frac{x - x_{min}}{x_{max} - x_{min}} \quad (4)$$

The normalization process has been done after splitting the data between training and test set, using only the data from the training set. This is done to avoid any data leakage.

3.4 Algorithms

To answer the main research question, that is, to what extent does a visual object's prominence have an effect on predicting a creative's click-through rate, we will employ the following machine learning algorithms: *Ordinary Least Squares*, *Poly2*, *XGBoost* and *Support Vector Regression*. OLS has been chosen as the baseline model. The baseline model does not include any feature interactions or regularization techniques. The choice of OLS with polynomial feature interactions is driven by the fact that due to the dimensional reduction of categorical variables in section 3.3.2, the complexity of the model is not to a scale that it cannot train well. XGBoost have been chosen as it has shown superior performance in other CTR prediction studies (Livne et al., 2020), (Brownlee, 2021). In addition, SVR has been chosen as it has shown superior performance in a similar study that predicts CTR based on visual features (Azimi et al., 2012).

3.4.1 Ordinary Least Squares (OLS)

OLS is a machine learning algorithm that is based on linear regression. There are multiple implementations of the OLS algorithm. This study will use from the *statsmodels* package in Python, the *LinearRegression* module (Seabold & Perktold, 2010). In more detail, OLS estimates the unknown parameters in a model. The method minimizes the sum of squared errors between the actual and predicted values which results in a linear best-fit line.

OLS has the disadvantage that it performs poorly when too many predictive features are added. Regularization techniques can be applied in order to improve generalizability. The two most common methods are L1 regularization (Lasso) and L2 regularization (Ridge). Regularized OLS puts constraints on the magnitude of the coefficients and will progressively shrink them towards zero. This helps to reduce the magnitude of fluctuations of the coefficients and will reduce the variance of the model (Boehm, 2022). In the case of Lasso, the penalization eliminates coefficients from the

regression equation, it does so based on the sum of absolute coefficients. In the case of Ridge, the penalization imposes a higher penalty for larger feature coefficients. There is one hyperparameter that will be tuned in OLS Regularization: *penalty*. The penalty term λ controls the strength of the penalty. When $\lambda = 0$ there is no penalty and it equals OLS. The higher the λ , the more the coefficients are shrunk to zero.

Both the regularization techniques will be used and compared with each other using from the *sklearn* package, the *Lasso* and *Ridge* module (Pedregosa et al., 2011).

3.4.2 Extreme Gradient Boosting (XGBoost)

XGBoost is a machine learning algorithm that is based on the gradient boosting of decision trees. Gradient boosting is a technique used to sequentially train decision trees. At each particular iteration, a new weak, decision tree is trained with respect to the error of the whole ensemble learnt so far (Natekin & Knoll, 2013).

There are several variations of gradient boosting algorithms. These include *GBM*, *LightGBM*, and *CatBoost*. However, in this study, we are going to use XGBoost as it has shown good performance in similar studies (Livne et al., 2020), (Brownlee, 2021). It performs well when there are more observations than features and when the data has a mixture of numerical and categorical features. This study will use from the *xgboost* package in Python, the *xgb* module (T. Chen & Guestrin, 2016).

XGBoost provides a large range of hyperparameters. We employed a grid-search approach to create a list of hyperparameters that the model tests. Table 4 describes the hyperparameters that will be tested during the grid search.

Table 4: Hyperparameters for XGBoost

Hyperparameters	Range
learning_rate	[0.001, 0.01, 0.1]
max_depth	[1, 3, 5, 7, 10]
min_child_weight	[1, 3, 5]
subsample	[0.5, 0.7]
colsample_bytree	[0.5, 0.7]
n_estimators	[100, 200, 500]

3.4.3 Support Vector Regression (SVR)

SVR is a machine learning algorithm based on minimizing the errors between the actual and predicted values within a threshold value, also called the epsilon-intensive tube. This method differs from OLS in that it captures errors based on the points that fall outside of this tube. It shows how tolerant the model is for errors.

The way SVR draws a decision boundary is the most critical part of the algorithm. Setting the right hyperparameters will help improving this task. SVR has four hyperparameters that will be tested using a grid-search approach. Table 5 describes the list of hyperparameters.

Table 5: Hyperparameters for Support Vector Regressor

Hyperparameters	Range
kernel	['linear', 'rbf', 'poly']
c	[0.1, 1, 10, 1000]
gamma	[1, 0.1, 0.01, 0.001, 0.0001]
epsilon	[0.1, 0.2, 0.3, 0.4, 0.5]

3.4.4 Predictors' Importance

We will employ a permutation feature importance to see which predictors are deemed to be the most important. The permutation feature importance randomly shuffles the predictor variables in a model and outputs the decrease in a model score for each feature. When the model score drops, this gives a good indication of how the dependent the model is on the feature. This study will use from the *sklearn* package in Python, the *permutation_importance* module (Buitinck et al., 2013).

The predictors' importance is calculated after a model is fitted. The algorithm randomly shuffles the values of a single column from the validation data, leaving the target and all other columns in place. This is called permutation. Re-ordering the values within a column while keeping the other intact should show less accurate predictions. It breaks the relationship between the feature and the true outcome. A feature is "unimportant" if shuffling its values leaves the model error unchanged. One of the main advantages of this measurement is that it is comparable across different algorithms (Molnar, 2020). The permutations will be held on the baseline model and the best performing machine learning model.

The algorithm outputs, per variable, the change in predictability if the variable was randomly shuffled. In our case, we use the mean squared error.

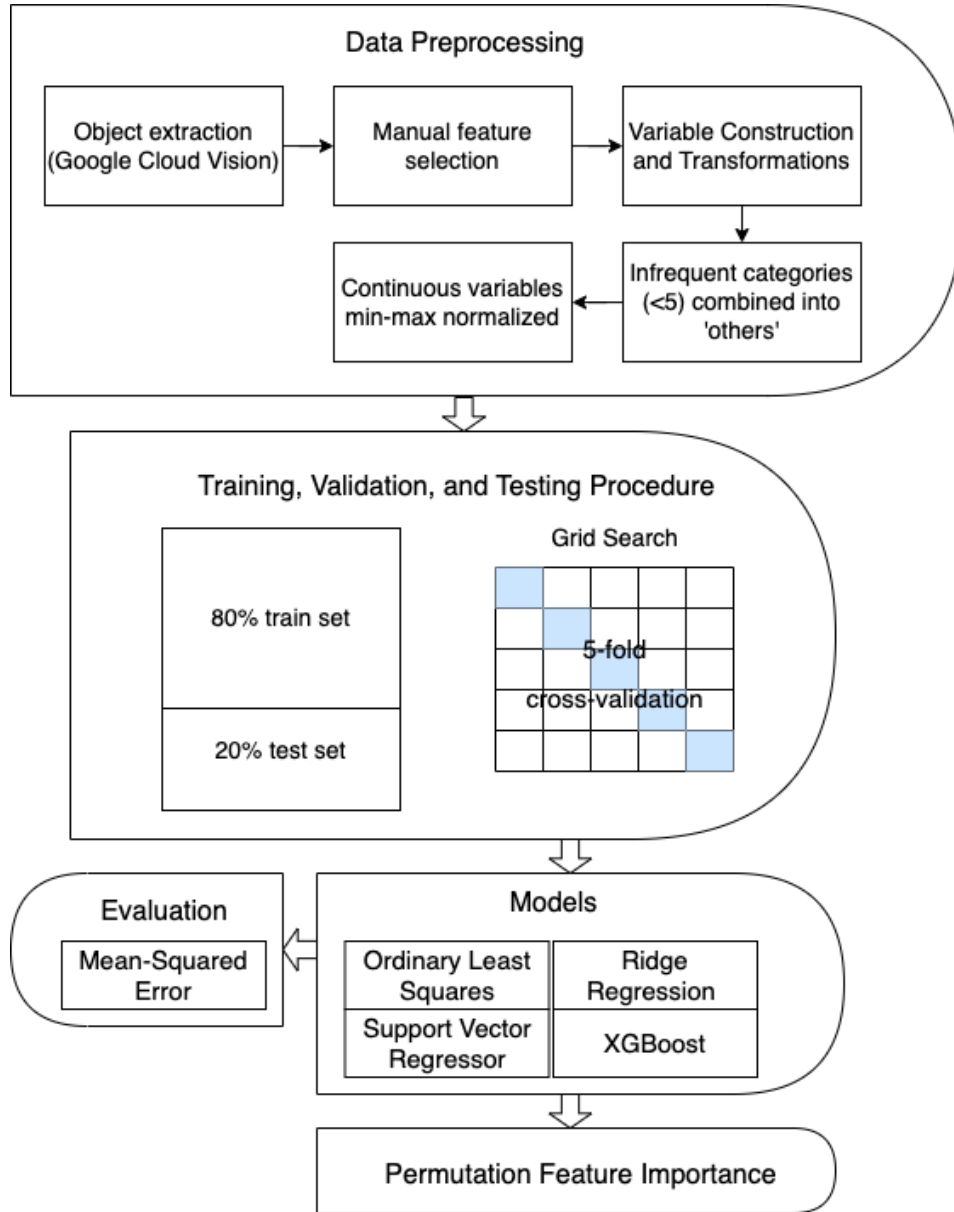


Figure 3: Pipeline Methodology

3.5 Training, Validation, and Testing Procedure

The split for training, development, and validation of the models was set at 80% while the remaining 20% was used to evaluate the out-of-sample predictive performance of the models. The research used a 5-fold cross validation to tune the hyperparameters of the machine learning algorithms. Cross-validation is the preferred method when assessing the outcomes of

predictive models. When employing 5 folds, the models outputs 5 different results on 5 different hold-out sets.

3.6 Performance Metrics

Mean Squared Error (MSE) is used to measure the prediction accuracy for each algorithm. It quantifies the extent to which the predicted response value for a given observation is close to the true response value for that observation. The MSE will be calculated for each fold, so a total of 5 times. From these 5 scores, the average will be taken. Then, the MSE will be calculated to evaluate the final performance of the models. The lower the MSE the better the model can predict the true responses.

$$MSE = \frac{1}{n} \sum_{i=1}^n (Y_i - \hat{Y})^2 \quad (5)$$

For a visual summary of the full project pipeline, see figure 3

4 RESULTS

In this section, we present and discuss the results of the parameter tuning with grid search and cross-validation assessment on the predictive performance of the models. The first two subsections, 4.1 and 4.2, present the results of the tuned models and predictive performances *without prominence*. Then, we repeat the task of tuning and predicting *with prominence* which will be presented in section 4.3 and 4.4.

4.1 Hyperparameter Tuning and Cross-Validation Results (Without Prominence)

Table 6 shows a comparison between the tuned models and the baseline without using the variable prominence. From the results, we can see that the XGBoost model performed best using a 5-fold cross validation. Other than that, only the regularized ridge OLS performer better than the baseline.

4.1.1 Ordinary Least Squares with Regularization

We have tested both the *Lasso* and *Ridge* regularization method with a 5-fold cross validation. The ridge regularized model performed better with a 5-fold average MSE of 0.019066. The penalty λ for this model was set at 0.79424. The lasso regularized model performed with a 5-fold average MSE of 0.024275. The penalty λ for this model was set at 9.62355. Both alphas have been found using a grid-search with a range between 0 and 10.

Table 6: MSE Comparison 5-fold Cross Validation Without Prominence

Models	MSE
Lasso OLS	0.024275
Ridge OLS	0.019066
XGBoost	0.018988
SVR	0.022462
Baseline	0.019297

When $\lambda = 0$, the penalty term has no effect, the regularized regression becomes an OLS model. The relatively low penalty λ of the ridge regularization therefore indicates that it is more directed towards an OLS. However, it is not 0, which means that the impact of the shrinkage of the coefficients has some impact. The high penalty λ in lasso regularization indicates that it has shrunk some of the regression coefficients with a minor contribution towards zero.

Since the lower MSE of the ridge regression performs better than lasso, it indicates that the coefficients of the predictors roughly have an equal size (Gareth, Daniela, Trevor, & Robert, 2013).

Table 7: OLS Regularization Hyperparameter Without Prominence

Hyperparameters	Lasso	Ridge
penalty	9.62355	0.79424

4.1.2 XGBoost

The best performing combination of hyperparameters of XGBoost (see Table 8) achieved an average MSE of 0.018988 on a 5-fold cross-validation.

We can see that the results of a grid_search led to an optimal tree depth of 2. A depth of a node is the number of edges from the node to the tree's root node. The model made an interaction of the order 2 (An & Ren, 2020). This indicates that there are predictor's within the model that are dependent on each other in predicting CTR.

4.1.3 Support Vector Regressor

The best performing combination of hyperparameters for SVR (see Table 9) achieved an average MSE of 0.022462 on a 5-fold cross-validation. We can see that the best performing kernel is polynomial. With a polynomial

Table 8: Hyperparameters XGBoost Without Prominence

Hyperparameters	Best values
learning_rate	0.1
max_depth	2
min_child_weight	1
subsample	0.7
colsample_bytree	0.5
n_estimators	200

kernel the model can learn non-linear models where there are features that interaction with each other.

Table 9: Hyperparameters SVR Without Prominence

Hyperparameters	Range
kernel	['poly']
c	[1]
gamma	[0.1]
epsilon	[0.1]

4.1.4 Baseline Ordinary Least Squares

The baseline OLS does not have any hyperparameters to tune. It has been trained on a 5-fold cross validation. On these five folds, it scored an average MSE of 0.019297.

4.2 Performance on Test Data (Without Prominence)

In this section, we report the performances of the fine-tuned models on the hold-out test data. All the chosen hyperparameters of these models can be found in section 4.1. The models will be re-trained on the training data and predictions have been made on a test-set.

All the models show better predictive performance then during the cross-validation stage. The best performing model on the test data is the Ridge regularized OLS with a MSE of 0.020003. This is the only model that predicts better than the baseline, however the improvement is minor.

Table 10: Predictive Performance Without Prominence

Models	MSE
Lasso OLS	0.022971
Ridge OLS	0.020003
XGBoost	0.020137
SVR	0.023361
Baseline	0.020095

4.3 Hyperparameter Tuning and Cross-Validation Results (With Prominence)

In this paragraph, we present the results of the hyperparameter tuning while including the variable prominence. The set-up is the same as for the one without prominence. First, a 5-fold cross validation will be used to determine the best performing hyperparameters. Then, these tuned models will be used to predict CTR on a test-set. Table 11 shows a comparison between the tuned models and the baseline. From the results, we can see that XGBoost performed best using a 5-fold cross validation. Other than that, none of the other models perform better than the baseline model.

Table 11: MSE Comparison 5-fold Cross Validation With Prominence

Models	MSE
Lasso OLS	0.019079
Ridge OLS	0.016249
XGBoost	0.014412
SVR	0.017788
Baseline	0.016244

4.3.1 Ordinary Least Squares with Regularization

We have tested both the *Lasso* and *Ridge* regularization method with a 5-fold cross validation. The ridge regularization performed better with an average MSE of 0.016249. The penalty λ for this model was set at 0.39861. The lasso model performed with an average MSE of 0.019079. The penalty λ was set at 9.96717. Both alphas have been found using a grid-search with a range between 0 and 10. See 12 for the chosen penalty.

Table 12: OLS Regularization Hyperparameter With Prominence

Hyperparameters	Lasso	Ridge
penalty	9.96717	0.39861

4.3.2 XGBoost

The best performing combination of hyperparameters of XGBoost (see Table 13) achieved an average of 0.014412 on 5-fold cross validation.

We can see that this grid search led to a depth of 3 with prominence, as compared to the depth of 2 without prominence. This indicates that adding the variable prominence has lead to a more complex relationship between independent variables of the order 1.

Table 13: Hyperparameters XGBoost With Prominence

Hyperparameters	Best values
learning_rate	0.01
max_depth	3
min_child_weight	1
subsample	0.7
colsample_bytree	0.7
n_estimators	500

4.3.3 Support Vector Regressor

The best performing combination of hyperparameters for SVR (see table 14) achieved an average MSE of 0.017788. The kernel is polynomial which indicates that it performs better with feature interactions in the model.

Table 14: Hyperparameters SVR With Prominene

Hyperparameters	Range
kernel	['poly']
c	[1000]
gamma	[0.1]
epsilon	[0.1]

4.3.4 Baseline Ordinary Least Squares

The baseline OLS does not have any hyperparameters to tune. It has been trained on a 5-fold cross validation. One these five folds, it scored an average MSE of 0.016243.

4.4 Performance on Test Data (With Prominence)

In this section, we report the performances of the fine-tuned models on the hold-out test data. All the chosen hyperparameters can be found in section 4.3. All the models will be re-trained on the training data and predictions have been made on a test-set.

The models all score lower then on the cross-validated data. The best performing model on the test data is XGBoost with a MSE of 0.015889. The ridge OLS also performs better then the baseline model with a MSE of 0.015923. The improvement from these models to that of the baseline OLS with prominence is however minor. See figure 4 for the actual versus the predicted output of the model.

Table 15: Predictive Performance With Prominence

Models	MSE
Lasso OLS	0.019289
Ridge OLS	0.015923
XGBoost	0.015889
SVR	0.019508
Baseline	0.016101

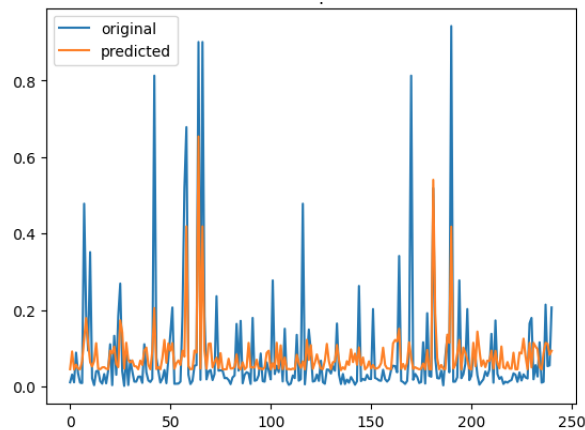


Figure 4: Predicted vs. Actual on Best Performing Model (XGB)

4.5 Predictors' Performance

In this section, we explore the differences in features' importance based on the best performing machine learning model without prominence (Ridge Regression) and the baseline model (OLS). Also, we explore the features' importance of best performing machine learning model with prominence (XGBoost) and the baseline model (OLS). The top five most important predictors without prominence have been plotted in Figure 5 and 6. Figure 7 and 8 shows the most important predictors with prominence. The method used for this task is the permutation feature importance, more details of this algorithm can be found in section 3.4.4.

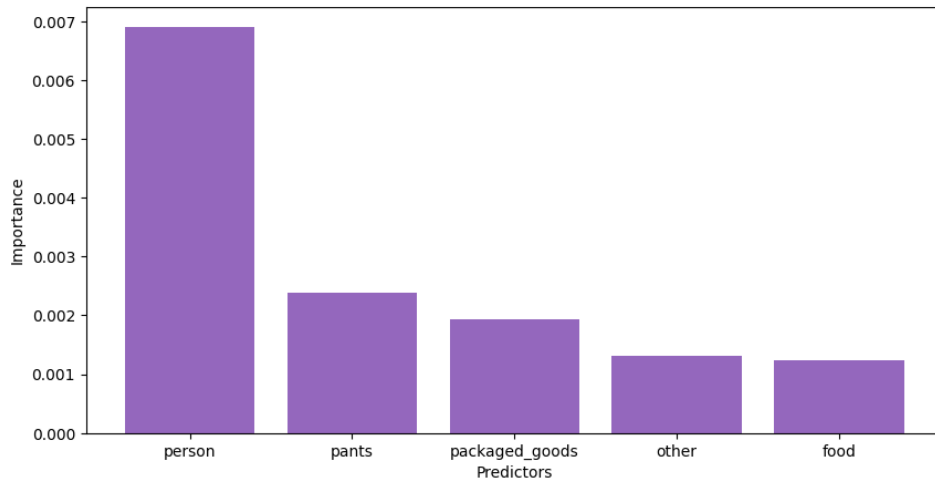


Figure 5: Predictors' Importance Baseline OLS Without Prominence

When you look at the range of the importance, it can be seen that the predictors' in OLS have a larger effect than those in the Ridge Regression model. There is one predictor that stands out in the Baseline OLS model, that is *person*. It has over two times the importance then the second variable *pants*. This indicates that when changing the values of *person*, it also changes the model performance significantly based on mean squared error.

As for the Ridge Regression predictors' importance, it shows a lot less variability in the model when the predictors' values are changed. There are three predictors that stand out. These are *packaged_goods*, *person*, and *pants*. These are the same three predictors' that are of highest importance in the baseline OLS model, however, with less variability.

The range of variability of OLS predictors with and without prominence are similar. The predictor *balloon* has the greatest importance in the baseline

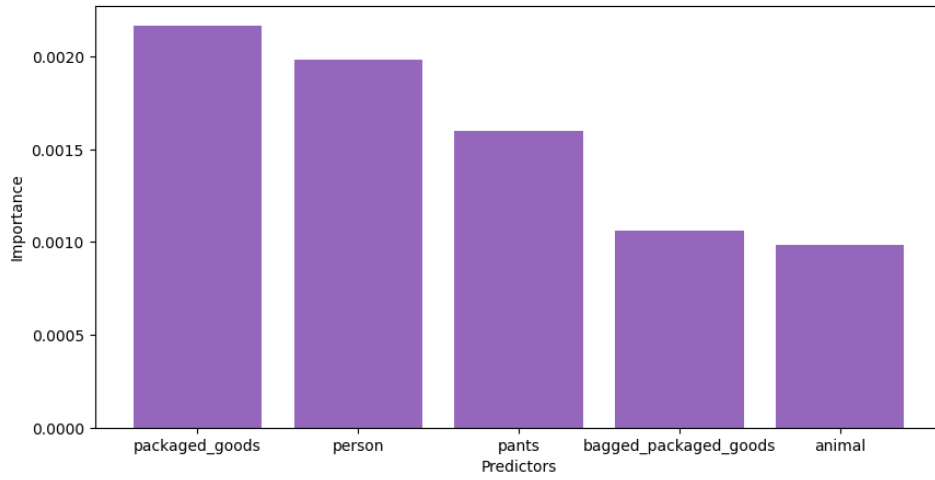


Figure 6: Predictors' Importance Ridge Regularized Without Prominence

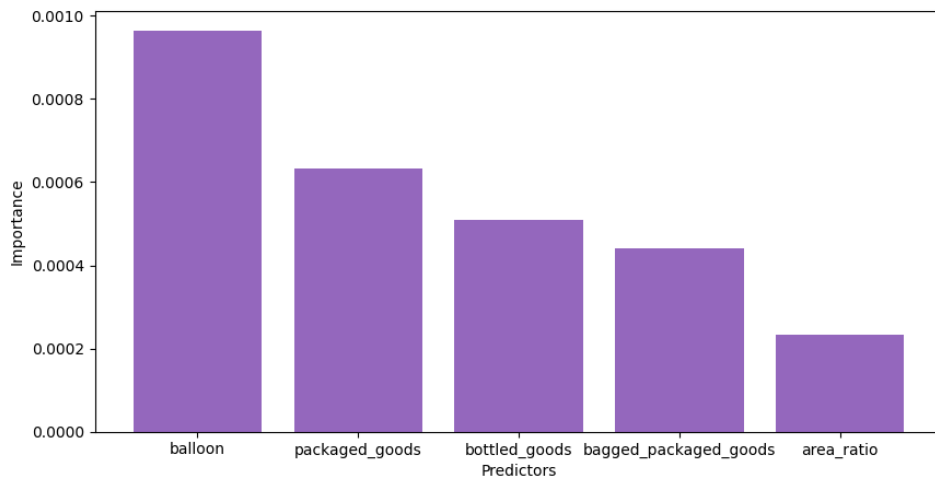


Figure 7: Predictors' Importance Baseline OLS With Prominence

OLS model. The predictor *packaged_goods* also scores a relatively high importance.

The best performing predictive model is the XGBoost with prominence. Interestingly, this is the model that also shows that *area_ratio* has the largest influence on predicting CTR. When these values are shuffled it influences the MSE with a rate of 0.009. The predictor *person* is also shown to be an important variable.

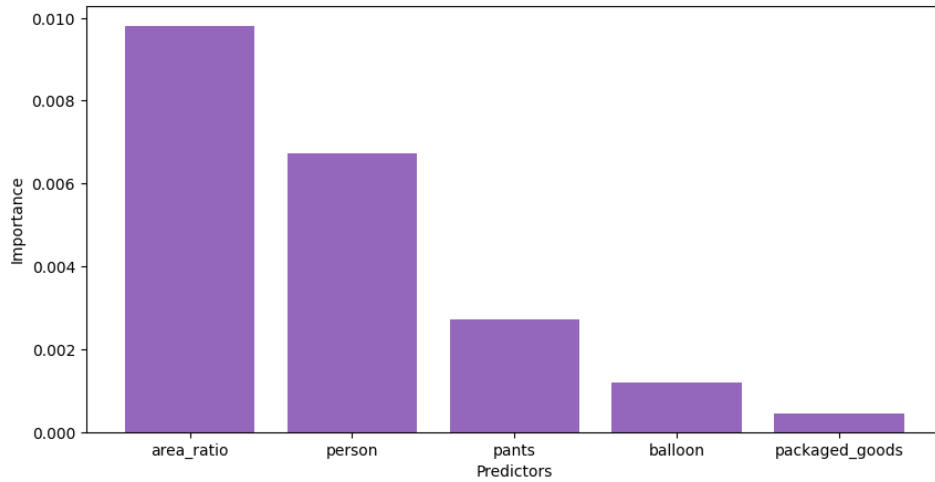


Figure 8: Predictors' Importance XGB With Prominence

5 DISCUSSION

The research goal for this thesis is to measure how accurately we can predict click-through rate by using the objects detected in a creative with its relative prominence. The sub-questions suggest a comparison between models, that is the tuned machine learning models: OLS with regularization and feature interactions, XGBoost and Support Vector Regressor with the baseline model: OLS without regularization and feature interactions. Also a comparison is made between the models with and without prominence to see if the variable adds more predictive power. The results on the predictive performance with and without prominence show that the MSE of all models is always less than one-tenth of a fraction (0.1). This metric is so low as we are dealing with a fraction as a target variable. The average CTR of the data set is 0.08114%.

5.1 Results Discussion

Results show that the regularized ridge model performs the best in predicting CTR without prominence with a MSE of 0.020003. The low penalty λ of 0.79424 indicates that it is more directed towards an OLS (λ of 0 = OLS). This also shows in the results as the difference between OLS and regularized ridge model only is 0.458%. Results show that the stand-alone objects can predict CTR to an extent where the error is around 0.02. This error is relatively high when you compare it to the average CTR of 0.08114%. The objects detected by themselves do not predict CTR very well. This indicates that the decision for a viewer to click on the creative is not very dependent

on the objects detected within a creative. A reason for this is that objects, as a stand-alone variable, are not the only element that influence whether a creative is visually appealing or not. An object within a creative can be shown in different ways. For instance, it can be shown as a tiny element with little contrast between the background and element color. However, it can also be shown as a large element with high contrast which makes it more stand out. The combination of the composition of an object within a creative is what makes it visually appealing, not the object as a stand-alone element (Landa, 2022).

On the other hand, three out of four models show that the variable *person* is deemed important in predicting CTR. For this element, arguments can be made that this does have a direct impact on CTR. Guido, Pichierri, Pino, and Nataraajan (2019) have researched the effect of having a person in a creative. It revealed that having a person in a creative increases advertisement preference. It increases the amount of attention that a viewer pays to the advertisement which results in a higher probability for a click.

When adding prominence to the objects, the error rate is still high when compared to the average CTR of the dataset. It also shows that the improved models consistently outperform the baseline models in predicting CTR with an average of 18.87%. By adding prominence to the best-performing model (XGBoost) it now can predict CTR with an error of around 0.016. The error is still pretty high, however, it is a significant improvement when compared to the baseline model without prominence. It shows a predictive improvement of 26.47%. It can therefore be said that adding prominence leads to a better predictive performance.

Interestingly, the optimal hyperparameter *max_depth* shows that there is a feature interaction of the order 3. It has raised it by a magnitude of 1 when comparing this to the XGBoost without prominence. This indicates that adding prominence has lead to a more complex relationship between the predictors of the model. When we look at the features' importance of the model with prominence, we can see that the most important features include *area_ratio* and *person*. This reveals that interaction effects between *area_ratio* and objects does have a significant importance. The variability of the MSE of the model is 0.009 when the values are permuted. Also, *person* again shows to be an important predictor in CTR. Having a person on a creative in relation with prominence also seems to have an effect. This is in line with the study by J. Chen et al. (2016) that revealed that having multiple persons in a creative (higher prominence) perform less well in terms of CTR than creatives with only one person.

5.2 Limitations

Given the dataset, we could only study the relationship of 416 unique creatives. In total, the Cloud Vision extracted 1204 objects of which 62 were unique. With more data, more categories could have been explored that might prove to be related with prominence and CTR. Most of the objects extracted from the creatives are low-involvement products where the effect of prominence might not come out as much.

Also, the size of the dataset has limited the modeling options to be used. Deep learning models have proven to be highly successful in predicting CTR, however, they require a substantially larger dataset. These models can generate interaction effects of way higher magnitude. Also, factorization machines, that have also been proven to predict CTR very well, could also not be used.

5.3 Future Research

This thesis is a first step towards a research domain with many opportunities. This research took the approach of predictive modeling of a variable that has never been tested before; it gives more attention to what the predictive implications are then explainability of the effects. This could be a future research direction by using statistical inference to see in what way prominence can have an effect on predicting CTR. Statistical inference can also reveal what combinations of prominences can contribute to a higher CTR.

Also, this research can be extended to domains other than that of a supermarket. Now, many of the detected objects are related to creatives of a supermarket, like *packaged_goods*. When exploring different industries, especially those that carry emotional weight, this could potentially discover different patterns in predicting CTR.

6 CONCLUSION

The aim of this study was to measure the effect that prominence has on predicting CTR with objects. The idea was based on the problem in advertising design that there are no data-driven approaches to decisions made on visual features. Also, within the domain of marketing analytics there can be a lot of improvements made to the low average percentage of CTR.

RQ1 To what extent can we predict click-through rate based on the objects detected within a creative.

The first sub-research question aims to predict click-through rate without prominence. The predictors therefore only are one-hot-encoded categorical variables. It can be seen that the baseline model achieved an average MSE of 0.020095. The only model that scored better was the regularized ridge OLS that included feature interactions. It scored an average MSE of 0.020003. The improved model shows a very minor error reduction of 0.458%. From the results it can be concluded that CTR can be predicted to an extent where the error is around 0.02. This number is relatively high when you compare it to the average CTR of 0.08114%. The objects detected by themselves do not predict CTR very well and this does not improve when tuning different models. This indicates that the decision for a viewer to click on the creative is not very dependent on the stand-alone objects detected.

RQ2 Does adding prominence lead to more predictive power?

The second sub-research question aims to predict click-through rate by adding prominence to the predictors. All the models show improvements in predictive performance as compared to the models that do not include prominence. The average predictive performance of all models increased by 18.87% when you added this variable. The mean squared error rate decreased to around 0.016 which is still pretty high when you compare it to the average CTR. It can therefore be concluded that although prominence significantly improves predictive performance, the results still show that you cannot predict CTR based only on objects and prominence solely.

RQ3 Do the machine learning algorithms outperform the baseline model in predicting click-through rate?

The third sub-research question aims to measure the difference in predictive modeling between the best performing model and the baseline model. The baseline model is the OLS regression without prominence, regularization and feature interaction and has an average MSE score of 0.020095. The best performing model is the tuned XGBoost model with prominence and has an average MSE score of 0.015889. The model improvement between these two is 26.47% which is a significant contribution. However, we should be aware that the error rate of 0.016 is still high when compared to the average CTR of 0.08114%.

Main How accurately can we predict click-through rate by using the prominence and the objects detected within a creative with machine learning algorithms?

The final conclusion of this study is that adding prominence has led to a significant improvement of predicting CTR based on the objects detected within a creative. The best predictive model, XGBoost that included prominence, however still has an average mean squared error rate of 0.016. Future research should be done on different data sets that include more creatives to analyze as this has been a limitation of the thesis.

REFERENCES

- Akanbi, O. A., Amiri, I. S., & Fazeldehkordi, E. (2015). Chapter 4 - feature extraction. In O. A. Akanbi, I. S. Amiri, & E. Fazeldehkordi (Eds.), *A machine-learning approach to phishing detection and defense* (p. 45-54). Boston: Syngress. Retrieved from <https://www.sciencedirect.com/science/article/pii/B9780128029275000046> doi: <https://doi.org/10.1016/B978-0-12-802927-5.00004-6>
- An, H., & Ren, J. (2020, 04). Xgbdeepfm for ctr predictions in mobile advertising benefits from ad context. *Mathematical Problems in Engineering*, 2020, 1-7. doi: 10.1155/2020/1747315
- Azimi, J., Zhang, R., Zhou, Y., Navalpakkam, V., Mao, J., & Fern, X. (2012). The impact of visual appearance on user response in online display advertising. In *proceedings of the 21st international conference on world wide web* (pp. 457-458).
- Boehm, B. (2022). *Regularized regression*. Retrieved from https://uc-r.github.io/regularized_regression#why
- Bradley, S. (2015, Feb). *Design principles: Dominance, focal points and hierarchy*. Retrieved from <https://www.smashingmagazine.com/2015/02/design-principles-dominance-focal-points-hierarchy/>
- Brownlee, J. (2021, Mar). *Xgboost for regression*. Retrieved from <https://machinelearningmastery.com/xgboost-for-regression>
- Buitinck, L., Louppe, G., Blondel, M., Pedregosa, F., Mueller, A., Grisel, O., ... Varoquaux, G. (2013). API design for machine learning software: experiences from the scikit-learn project. In *Ecml pkdd workshop: Languages for data mining and machine learning* (pp. 108-122).
- Chakrabarti, D., Agarwal, D., & Josifovski, V. (2008). Contextual advertising by combining relevance with click feedback. In *Proceedings of the 17th international conference on world wide web* (p. 417-426). New York, NY, USA: Association for Computing Machinery. Retrieved from <https://doi.org/10.1145/1367497.1367554> doi: 10.1145/1367497.1367554
- Chang, Y.-W., Hsieh, C.-J., Chang, K.-W., Ringgaard, M., & Lin, C.-J. (2010, 04). Training and testing low-degree polynomial data mappings via linear svm. *The Journal of Machine Learning Research*, 11, 1471-1490.

- Chen, J., Sun, B., Li, H., Lu, H., & Hua, X.-S. (2016). *Deep ctr prediction in display advertising*. arXiv. Retrieved from <https://arxiv.org/abs/1609.06018> doi: 10.48550/ARXIV.1609.06018
- Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining* (pp. 785–794). New York, NY, USA: ACM. Retrieved from <http://doi.acm.org/10.1145/2939672.2939785> doi: 10.1145/2939672.2939785
- Cheng, H., Zwol, R. v., Azimi, J., Manavoglu, E., Zhang, R., Zhou, Y., & Navalpakkam, V. (2012). Multimedia features for click prediction of new ads in display advertising. In *Proceedings of the 18th acm sigkdd international conference on knowledge discovery and data mining* (p. 777–785). New York, NY, USA: Association for Computing Machinery. Retrieved from <https://doi.org/10.1145/2339530.2339652> doi: 10.1145/2339530.2339652
- CXL. (n.d.). *What is a "good" click-through rate? click-through rate benchmarks*. Retrieved from <https://cxl.com/guides/click-through-rate/benchmarks/>
- Gareth, J., Daniela, W., Trevor, H., & Robert, T. (2013). *An introduction to statistical learning: with applications in r*. Springer.
- Guido, G., Pichierri, M., Pino, G., & Natarajan, R. (2019). Effects of face images and face pareidolia on consumers' responses to print advertising. *Journal of Advertising Research*, 59(2), 219–231. Retrieved from <https://www.journalofadvertisingresearch.com/content/59/2/219> doi: 10.2501/JAR-2018-030
- Guo, H., Tang, R., Ye, Y., Li, Z., & He, X. (2017). *Deepfm: A factorization-machine based neural network for ctr prediction*. arXiv. Retrieved from <https://arxiv.org/abs/1703.04247> doi: 10.48550/ARXIV.1703.04247
- Jiang, Z., Gao, S., Shi, Y., Wang, G., Chen, L., & Sidibe, A. (2018). A ctr prediction approach for advertising based on embedding model and deep learning. In *2018 ieee intl conf on parallel distributed processing with applications, ubiquitous computing communications, big data cloud computing, social computing networking, sustainable computing communications (ispa/iucc/bdcloud/socialcom/sustaincom)* (p. 744–750). doi: 10.1109/BDCloud.2018.00112
- Juan, Y., Zhuang, Y., Chin, W.-S., & Lin, C.-J. (2016). Field-aware factorization machines for ctr prediction. In *Proceedings of the 10th acm conference on recommender systems* (pp. 43–50).
- Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., ... Liu, T.-Y. (2017). Lightgbm: A highly efficient gradient boosting decision tree. In I. Guyon et al. (Eds.), *Advances in neural infor-*

- tion processing systems (Vol. 30). Curran Associates, Inc. Retrieved from <https://proceedings.neurips.cc/paper/2017/file/6449f44a102fde848669bdd9eb6b76fa-Paper.pdf>
- Landa, R. (2022). *Advertising by design: Generating and designing creative ideas across media*. Wiley.
- Li, Z., Wu, S., Cui, Z., & Zhang, X. (2021). *Graphfm: Graph factorization machines for feature interaction modeling*. arXiv. Retrieved from <https://arxiv.org/abs/2105.11866> doi: 10.48550/ARXIV.2105.11866
- Livne, A., Dor, R., Mazuz, E., Didi, T., Shapira, B., & Rokach, L. (2020). *Iterative boosting deep neural networks for predicting click-through rate*. arXiv. Retrieved from <https://arxiv.org/abs/2007.13087> doi: 10.48550/ARXIV.2007.13087
- Minnium, P. (2021, May). *The tyranny of a blank page: Why creative matters most*. Retrieved from <https://martech.org/tyranny-blank-page-creative-matters/>
- Molnar, C. (2020). *Interpretable machine learning*. Lulu. com.
- Molnar, C. (2022). Chapter 8 - global model-agnostic methods. In *Interpretable machine learning: A guide for making black box models explainable*. Christoph Molnar.
- Natekin, A., & Knoll, A. (2013, 12). Gradient boosting machines, a tutorial. *Frontiers in neurorobotics*, 7, 21. doi: 10.3389/fnbot.2013.00021
- Newbold, C. (2017, Mar). *What is visual rhetoric?* Retrieved from <https://thevisualcommunicationguy.com/2014/02/13/what-is-visual-rhetoric/>
- Ouyang, W., Zhang, X., Li, L., Zou, H., Xing, X., Liu, Z., & Du, Y. (2019). Deep spatio-temporal neural networks for click-through rate prediction. In *Proceedings of the 25th acm sigkdd international conference on knowledge discovery & data mining* (pp. 2078–2086).
- Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12, 2825–2830.
- Pieters, R., Wedel, M., & Zhang, J. (2007). Optimal feature advertising design under competitive clutter. *Management Science*, 53(11), 1815–1828. doi: 10.1287/mnsc.1070.0732
- Ren, J., Zhang, J., & Liang, J. (2020). Feature engineering of click-through-rate prediction for advertising. In Q. Liang, X. Liu, Z. Na, W. Wang, J. Mu, & B. Zhang (Eds.), *Communications, signal processing, and systems* (pp. 204–211). Singapore: Springer Singapore.
- Seabold, S., & Perktold, J. (2010). statsmodels: Econometric and statistical modeling with python. In *9th python in science conference*.
- Statista. (2022, Jun). *Digital banner advertising - worldwide: Statista market forecast*. Retrieved from <https://www.statista.com/>

- [outlook/amo/advertising/digital-banner-advertising/worldwide#ad-spending](#)
- Steen, M. v. d. (n.d.). Ad vision 2.0 - using machine learning power of google's cloud vision api.
- Wang, F., Suphamitmongkol, W., & Wang, B. (2013). Advertisement click-through rate prediction using multiple criteria linear programming regression model. *Procedia Computer Science*, 17, 803-811. Retrieved from <https://www.sciencedirect.com/science/article/pii/S1877050913002366> (First International Conference on Information Technology and Quantitative Management) doi: <https://doi.org/10.1016/j.procs.2013.05.103>
- Wang, Z., Yu, Q., Shen, C., & Hu, W. (2016). Feature selection in click-through rate prediction based on gradient boosting. In H. Yin et al. (Eds.), *Intelligent data engineering and automated learning – ideal 2016* (pp. 134–142). Cham: Springer International Publishing.
- Wu, S., Yu, F., Yu, X., Liu, Q., Wang, L., Tan, T., ... Huang, F. (2020, jul). TFNet: Multi-semantic feature interaction for CTR prediction. In *Proceedings of the 43rd international ACM SIGIR conference on research and development in information retrieval*. ACM. Retrieved from <https://doi.org/10.1145/2F3397271.3401304> doi: 10.1145/3397271.3401304
- Xu, W., Manavoglu, E., & Cantu-Paz, E. (2010). Temporal click model for sponsored search. In *Proceedings of the 33rd international acm sigir conference on research and development in information retrieval* (p. 106–113). New York, NY, USA: Association for Computing Machinery. Retrieved from <https://doi.org/10.1145/1835449.1835470> doi: 10.1145/1835449.1835470
- Yan, C., Chen, Y., Wan, Y., & Wang, P. (2021, 06). Modeling low- and high-order feature interactions with fm and self-attention network. *Applied Intelligence*, 51, 1-13. doi: 10.1007/s10489-020-01951-6
- Yang, Y., Li, X., Zeng, D., & Jansen, B. J. (2018). Aggregate effects of advertising decisions: A complex systems look at search engine advertising via an experimental study. *Internet Research*.
- Yang, Y., & Zhai, P. (2022). Click-through rate prediction in online advertising: A literature review. *Information Processing & Management*, 59(2), 102853.
- Ye, K., Nazari, N. H., Hahn, J., Hussain, Z., Zhang, M., & Kovashka, A. (2019). Interpreting the rhetoric of visual advertisements. *IEEE transactions on pattern analysis and machine intelligence*, 43(4), 1308–1323.
- Yu, F., Liu, Z., Liu, Q., Zhang, H., Wu, S., & Wang, L. (2020a). Deep interaction machine: A simple but effective model for high-order

- feature interactions. In *Proceedings of the 29th acm international conference on information amp; knowledge management* (p. 2285–2288). New York, NY, USA: Association for Computing Machinery. Retrieved from <https://doi.org/10.1145/3340531.3412077> doi: 10.1145/3340531.3412077
- Yu, F., Liu, Z., Liu, Q., Zhang, H., Wu, S., & Wang, L. (2020b). Deep interaction machine: A simple but effective model for high-order feature interactions. In *Proceedings of the 29th acm international conference on information amp; knowledge management* (p. 2285–2288). New York, NY, USA: Association for Computing Machinery. Retrieved from <https://doi.org/10.1145/3340531.3412077> doi: 10.1145/3340531.3412077
- Zdaniuk, B. (2014). Ordinary least-squares (ols) model. In A. C. Michalos (Ed.), *Encyclopedia of quality of life and well-being research* (pp. 4515–4517). Dordrecht: Springer Netherlands. Retrieved from https://doi.org/10.1007/978-94-007-0753-5_2008 doi: 10.1007/978-94-007-0753-5_2008