MEF UNIVERSITY

THE EFFECT OF BERT-BASED GRAMMATICAL ANALYSIS ON GOOGLE SEARCH RESULTS

Capstone Project

Oğuz Çolak

İSTANBUL, 2021

MEF UNIVERSITY

THE EFFECT OF BERT-BASED GRAMMATICAL ANALYSIS ON GOOGLE SEARCH RESULTS

Capstone Project

Oğuz Çolak

Advisor: Prof. Dr. Özgür Özlük

İSTANBUL, 2021

MEF UNIVERSITY

Name of the project: The Effect of BERT-based Grammatical Analysis on Google Search Results
Name/Last Name of the Student: Oğuz Çolak
Date of Thesis Defense: 10/06/2021

I hereby state that the graduation project prepared by Oğuz Çolak has been completed under my supervision. I accept this work as a "Graduation Project".

10/06/2021
Prof. Dr. Özgür Özlük

I hereby state that I have examined this graduation project by Oğuz Çolak which is accepted by his supervisor. This work is acceptable as a graduation project and the student is eligible to take the graduation project examination.

10/06/2021

Director
of
Big Data Analytics Program

We hereby state that we have held the graduation examination of _____ and agree that the student has satisfied all requirements.

THE EXAMINATION COMMITTEE

Committee Member	Signature
1. Prof. Dr. Özgür Özlük	
2. Asst. Prof. Tuna Çakar	……..

ACADEMIC HONESTY PLEDGE

I promise not to collaborate with anyone, not to seek or accept any outside help, and not to give any help to others.

I understand that all resources in print or on the web must be explicitly cited.

In keeping with MEF University's ideals, I pledge that this work is my own and that I have neither given nor received inappropriate assistance in preparing it.

Oğuz Çolak 10/06/2021

Name Date Signature

EXECUTIVE SUMMARY

THE EFFECT OF BERT-BASED GRAMMATICAL ANALYSIS ON GOOGLE SEARCH RESULTS

Oğuz Çolak

Advisor: Prof. Dr. Özgür Özlük

JUNE, 2021, 29 pages

This study aims to study the BERT, namely Bidirectional Encoder Representations from Transformers model, which is introduced by Google and is of great importance in content analysis, and to examine the role of grammatical accuracy in the process of content quality measurement and Search Engine Results Pages (SERP). BERT has an important role among the algorithms used by Google in order to maintain the quality of search results and to provide more relevant content to users by understanding the content more effectively.

In this study, CoLA data, which is accepted as the most reliable data in this field and therefore used frequently in similar BERT studies, is used. The main purpose here is to make a BERT-based grammatical evaluation of sentences in a content and then examine these results on pages with optimal ranking values, to examine the connection between search results and grammatical accuracy and the importance of this parameter.

In this context, the project consists of two phases. In the first phase, the content of the pages that are visible in the first 20 in 50 different queries are scored with the pre-trained BERT model. In the second phase, a dataset that includes different SEO-focused metrics of the same pages is created manually, and the importance of the BERT score among these features is investigated.

Key Words: BERT, Bidirectional Encoder Representations from Transformers, SEO, Search Engine Optimization, Content Quality, Natural Language Processing, Machine Learning.

ÖZET

BERT TABANLI GRAMER ANALİZİNİN GOOGLE ARAMA SONUÇLARINDAKİ ETKİSİ

Oğuz Çolak

Proje Danışmanı: Prof. Dr. Özgür Özlük

HAZİRAN, 2021, 29 sayfa

Bu çalışma Google tarafından arama sonuçlarında kullanıcılara kaliteli içerikler sunarak, kullanıcı deneyimini artırmak adına önemli bir model olan BERT yani Bidirectional Encoder Representations from Transformers üzerine yoğunlaşıp, içerik kalitesini ölçümleme sürecinde gramatik doğruluğun önemini incelemeyi amaçlamaktadır.

Çalışmada bu alanda en güvenilir kaynak olarak kabul edilen ve benzer BERT çalışmalarında da faydalanılan CoLA dataseti üzerinde çalışılmaktadır. Amaç ise bir içerikteki cümlelerin BERT gözüyle gramatik değerlendirmesini yapıp daha sonra bu sonuçları güncel arama sonuçlarında iyi konumlarda görünürlük kazanan sayfalara yansıtarak, arama sonuçları ile bu kriter arasındaki bağlantıyı ve bu kriterin önemini incelemektir.

Bu proje iki aşamadan oluşmakta ve ilk aşamada 50 farklı sorguda ilk 20 konumda görünürlük kazanan sayfaların içerikleri BERT modeli ile puanlanmaktadır, ikinci aşamada ise yine aynı sayfalara yönelik ve farklı SEO metriklerini de içeren bir veri seti oluşturularak, bu veri seti üzerinden BERT puanının diğer metriklere kıyasla önemi incelenmektedir.

Anahtar Kelimeler: BERT, Bidirectional Encoder Representations from Transformers, SEO, Arama Motoru Optimizasyonu, İçerik Kalitesi, Doğal Dil İşleme, Makine Öğrenmesi.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1. INTRODUCTION

Google manages to differ from other search engines, not only in terms of market share, but also in terms of its algorithm and user experience (Dritsa et al., 2020). Understanding the models and algorithms used by Google is now critical for Search Engine Optimization experts to come up with successful projects (Joshi et al., 2018). This situation can be compared to the position of a store. A store in a busy street and in a good location can reach many more customers and sell a lot more, while a store in a non-crowded neighborhood and location often has to be content with only customers who know that store. Sometimes even those loyal customers may change their minds and prefer stores on busy streets. For this reason, every store owner aims to be in the best street and the best location. In this context, if Google is considered as the busiest street, which responds to 3.5 billion queries a day, everyone will aim to be in the best location, namely in the first 3 positions and on the first page. In order to achieve this, it is necessary to be able to respond in accordance with the algorithms developed year by year. However, this requires a complete expertise, as erroneous work, especially techniques considered as black-hat, can lead to a troublesome process that can lead to sites being penalized and even removed from search results. For this reason, SEO experts support companies and individuals seeking support in this field, in in-house marketing teams, as an independent agency or as a freelancer. One of the main points of SEO is that a different strategy must be developed for each purpose. In other words, it is not possible to achieve success with a single type of strategy regardless of purpose and objective. SEO experts determine different strategies for every website, every page, every sector, every KPI with the experience they have gained over the years, and thus, they can be successful. However, as the development of the SEO industry directly improves the user experience offered by websites and offers higher quality content to Google to provide to users, Google provides basic information on topics such as Search Engine Optimization (SEO) Starter Guide through its own basic documents to people who want to have more knowledge in the field of SEO but do not yet have any knowledge, because a website should be built to benefit users and any optimization should be geared toward making the user experience better (“Search Engine Optimization (SEO) Starter Guide” n.d.), Google also shares relatively more comprehensive information such as the Search Quality Evaluator Guideline (which is prepared for the third-party Search Quality Raters working for Google) to explain what is important and what is not, and what has been changed for specific concepts (“Search

Quality Evaluator Guideline", 2020). However, due to the dynamism and constant change of the Google algorithm, many written documents and information may become out of date and ineffective in a short time. For this reason, Google regularly shares new information from various sources and social media accounts, especially Search Central, and holds interactive online meetings with webmasters.

The task of Search Engine Optimization experts can be briefly summarized as optimizing websites according to the criteria of search engines, structuring them in a user-friendly and bot-friendly manner and thus reaching more users by reaching optimal rankings and gaining visibility in the search results. If this process is to be exemplified by the criteria of Google, which is the search engine with the highest market share, it is of great importance for SEO experts to perform on-page, off-page and technical SEO optimizations and produce content that provides maximum benefit for the user, so that they can respond to more than 200 parameters in Google's algorithm in the most effective way. Although there is no official publication on the 200 ranking factor statement, this concept was first mentioned in 2009 when Google's Matt Cutts mentioned there were "over 200 variables" in the Google algorithm. However, considering the development and change of the Google algorithm since 2009, it would not be wrong to assume that the current algorithm is based on many more criteria than this.

Google's advancements through various ML and AI applications to maintain the quality of search results offer many options. However, this study will focus on Bidirectional Encoder Representations from Transformers (BERT), which provides a great advantage to Google in the field of content analysis, and will aim to measure the effect of grammatical analysis on search results. As mentioned before, considering that the quality of the search results depends on more than 200 parameters and neither these parameters nor their weights are fully explained, BERT-based grammatical analysis alone is not expected to give an effective result. However, it will still help to understand how important it is for BERT to understand the content especially by analyzing the pages that are visible on the first page. In addition, the result to be reached here will not include all the activities evaluated within the scope of the Google algorithm (which is an impossible situation for any institution or person other than Google), the result will only be analyzed through the basic features that can be measured by third party tools. In other words, it is not possible to see this study as a fully effective guide, and it only aims at a comparison between the discussed metrics.

1.1. Bidirectional Encoder Representations from Transformers (BERT) Literature Survey

BERT is an algorithm that increases Google's understanding of human language. Google started beta tests about 3 years ago and helped to improve the content evaluation and indexing processes very effectively by launching it about a year ago. It is a neural network-based technique for natural language processing (NLP) pre-training, and as Kim et al. (2020) states that at the point of Grammar induction, especially in the last few years, the Natural Language Processing (NLP) community has tended to make use of pre-trained language models (LMs). BERT (with ELMo) has proven to be surprisingly effective as a way to acquire contextualized word representations, and plays a key role in recent improvements to various models for various NLP tasks. Although Devlin et al. (2019) is the official document published on BERT and the most comprehensive information can be obtained from this document, papers provided by the universities upon the success of this model also provide valuable information about both the basis and the details of the model. In one of those papers by Rogers et al. (2020), it is stated that the traditional workflow for BERT consists of two stages: pre-training and fine-tuning. Pretraining uses two self-supervised tasks: masked language modeling (MLM, prediction of randomly masked input tokens) and next sentence prediction (NSP, predicting if two input sentences are adjacent to each other). In fine-tuning for downstream applications, one or more fully-connected layers are typically added on top of the final encoder layer.

1.2. Content Quality Overview

The quality of the content is critical for the pages presented in the search results to provide the user with the necessary information and to provide an effective user experience, but this is not the only issue. With the developing technology and tools, the use of a flawless language is an indispensable requirement in developing every information, conversation, and answer options offered to the user in the field of AI. At this point, this study will aim to develop a BERT model to check the grammatical accuracy of the content, to improve the user experience in every field, and to provide quality content optimization that offers an effective experience both in site content and tools such as chatbots.

The first step of this study will focus on understanding whether the sentences are grammatically correct by evaluating the content presented in the SERP (Search Engine Results Page) and ranking on the first page in important queries. It will provide useful results to understand if the content is meaningful and written by a human. This is one of the main criteria

Google pays attention to for measuring content quality, especially whether or not content is meaningful.

1.3. Sections

The rest of the report will proceed as follows. Section 2 will provide information about data and Exploratory Data Analysis (EDA) processes. Section 3 will provide information about basic model set up, train and test processes. In Section 4, the BERT model will be applied to the pages in the current search results, and the grammatical accuracy of the pages will be examined. In section 5, the second phase of the project will begin. In other words, a dataset will be created by providing different SEO metrics that can be measured with 3rd party tools for the pages whose BERT scores are measured. Later, this dataset will be introduced and EDA studies will be applied. In section 6, an appropriate model will be selected and applied to this dataset, and the importance of the BERT score compared to other features will be examined.

2. UNDERSTANDING THE DATA

CoLA is a collection of sentences from the linguistics literature with expert acceptability labels which, at over 10k examples, is by far the largest of its kind (Warstadt et al., 2019). The Corpus of Linguistic Acceptability (CoLA) consists of 10657 sentences compiled over 23 linguistics publications in total with acceptableness information. The dataset is split into an in-domain set with sentences from 17 sources and an out-of-domain set with the remaining 6 sources. The in-domain set is split into train/dev/test sections, and the out-of-domain set is split into dev/test sections. The test sets are not made public by the provider.

Due to the reasons mentioned above, the public data that will be used in this project is split into the following files:

- raw/in_domain_train.tsv (8551 lines)
- tokenized/in_domain_train.tsv (8551 lines)
- raw/in_domain_dev.tsv (527 lines)
- tokenized/in_domain_dev.tsv (527 lines)
- raw/out_of_domain_dev.tsv (516 lines)
- tokenized/out_of_domain_dev.tsv (516 lines)

2.1. Features

- Column 1: The source of the sentence.
- Column 2: The label for the decision regarding the acceptability.

(0 = not a grammatical sentence, 1 = a grammatical sentence).

- Column 3: The judgment for the decision regarding the acceptability.
- Column 4: The sentence.

Number of training sentences: 8,551

	sentence_source	label	label_notes	sentence
6020	c_13	1	NaN	The cat had eaten.
4312	ks08	0	*	It tried to be intelligent.
2727	l-93	0	*	We offered a job behind her.
2169	l-93	1	NaN	a The butcher cuts the meat.
3172	l-93	1	NaN	Marlene dressed herself.
4759	ks08	1	NaN	Edward's help, you can rely on.
1243	r-67	0	*	Myron is making Suzie's claim that dead is bet...

Figure 1: An overview of the features

2.2. Exploratory Data Analysis

This section will cover data cleaning, preprocessing for tokenization, hyperparameter tuning and tokenization processes.

2.2.1. Data Cleaning

The data cleaning process started with the examination of null values. The CoLA dataset is a very clean and widely used data set, especially in the BERT field. This is clearly seen when null values are examined. No null value is detected in columns other than label_notes column (Figure 2). The null values in the label_notes column are actually a natural result in cases where the data in the label column is 1, that is, the sentence examined is accepted as grammatically correct.

At this stage, a subset is created for grammatically incorrect sentences and a new null test is performed on that subset to make sure that only grammatically accepted sentences have null values for the label_notes. As a result, it is noticed that one of the sentences in the data set does not contain any notes, although it is incorrect (Figure 3).

```
sentence_source       0
label                 0
label_notes        6024
sentence              0
```

Figure 2: The total number of the null values

```
sentence_source    0
label              0
label_notes        1
sentence           0
```

Figure 3: Null values for the grammatically incorrect sentences

The data is cleaned by leaving this null value out of the data frame, even if it was a single line. As a result, the total number of rows dropped to the level of 8550 in which only grammatically acceptable sentences with a label value of 1 have null values in the label_notes column and the others have not (Figure 4).

In the current version of dataset, 70.4% of the sentences reviewed are labeled as grammatically correct and 29.6% are labeled as grammatically incorrect.

```
RangeIndex: 8550 entries, 0 to 8549
Data columns (total 4 columns):
 #   Column           Non-Null Count  Dtype
---  ------           --------------  -----
 0   sentence_source  8550 non-null   object
 1   label            8550 non-null   int64
 2   label_notes      2527 non-null   object
 3   sentence         8550 non-null   object
```

Figure 4: The cleaned version of the data

2.2.2. Pre-processing for Tokenization

The first thing to do for a successful input representation is tokenization. First of all, the sentence and labels are assigned and then the BERT tokenizer is imported over transformers.

In order to train the classifier effectively, each input must contain only one sentence. For this, [CLS] tokens are added to the beginning of each sentence and [SEP] tokens at the end. In this way, it was aimed for the model to understand effectively where the sentence begins and ends (Quijano et al., 2021).

Among the constraints of BERT there are also constraints of sentence lengths. In this context, in order to process the data, all sentences must have the same length and a maximum of 512 tokens. Since the sentences in the CoLA dataset have different lengths, [PAD] token is used in order to achieve this (Sun et al., 2020). After determining the maximum sentence length with [PAD], attention mask is used in order to separate the pad tokens from the rest of the sentence. Attention mask is a method that separates which tokens should be processed in the model and which should not, thus leaving the pad tokens out of the model.

2.2.3. Hyperparameter Tuning

Hyperparameters are undoubtedly an important factor in machine learning models, especially in my project. At this point, wandb and sweeps provide a great advantage to make the model better over the most effective hyperparameters.

It is very important that hyperparameters are designed to return the most suitable results, especially the number of epochs. For example, keeping the number of epochs at a low level may cause underfit problems, and keeping them at very high levels may cause overfit problems.

For this reason, it is necessary to determine the ideal number of epochs, the number of layers and the number of nodes per layer, and to optimize the model in the most appropriate way in order to give a more effective result.

Although there are models other than Grid search (Naive Bayes etc) among the most preferred hyperparameter tuning methods, the Grid search, which is accepted as the most effective hyperparameter tuning method in the NLP area (Hutter et al., 2015) and is preferred in many studies both presented in papers and shared on the web, is used in this project. For this, wandb is imported into the project and configured over the metrics (name, goal) and parameters (learning_rate, batch_size, epochs) criteria. As a part of this process, 5e-5, 3e-5 and 2e-5 values

are assigned to learning_rate, 16 and 32 values are assigned to batch_size, 2, 3, and 4 values are assigned to epochs. It's aimed to maximize the val_accuracy and and values are set on the best recommended hyper parameters in order to perform transfer learning.

2.2.4. Tokenization

In this phase, the tokenization process, which was started as part of the preparation process, is completed. Tokenization can be defined as the step of breaking a text into sentences or words and calling them tokens, making them workable in smaller scales. The importance in NLP is, of course, that every word in the text has a separate meaning and significance.

3. MODEL SET UP

Pre-trained BERT model is used in this project because this model reduces the process that will take days when it is performed with Recurrent Neural Networks (RNN) to only a few hours and this method is generally preferred in BERT-bases analyzes. In addition, since the transfer learning will be performed, the data points in the CoLA dataset will be useful in order to give an effective result.

3.1. Train and Test Split

Model set up process is started by splitting my train and test data. The train-test split phase is a process for making predictions by running the model on the part of the data that is not used in the train process, and it is useful and important to measure how effectively the algorithm works in predictive modeling projects. At this phase, 85% of the data is splitted for training and 15% for the test. As a result, the data was divided into 7,267 samples for train and 1,283 samples for validation.

3.2. Pre-trained Model for BERT

At this stage, BertForSequenceClassification is imported over transformers and the model is defined such that the num_labels value will be 2, the output_attentions value (option to returns attentions weights) and the output_hidden_states value (returns all hidden-states) to be False. BertForSequenceClassification is one of the classes used for fine-tuning and it is used in order to modify the pre-BERT model to provide trained outputs for classification before training the model on the dataset to be well-suited for the project. Basically it is a normal BERT model with an additional linear layer for classification that is used as a sentence classifier.

An implementation of AdamW optimizer which helps to optimize weight decay and learning rate separately is used at this phase.

Learning rate (with lr=wandb.config.learning_rate) was imported from wandb.config. A learning rate scheduler is used in order to perform learning rate decay. The training epoch, a hyperparameter, is imported (with epochs=wandb.config.epochs) via wandb.config.

Then, by starting the train process, the hyperparameters that give the best value on the validation set are determined. After performing 85 different runs in almost 10 hours, 49^{th} run (batch_size:16, epochs:3, learning_rate:0.00003) is determined as the most effective model

with a val_accuracy value of 0.8588. It is determined that epochs is the most useful parameter in terms of both importance and correlation. The training loss values which are showing the errors in the training set and the validation loss values which are showing the errors in the validation set are also reviewed.

Figure 6 represents the time distribution of validation accuracy scores obtained in different runs. In this figure, it can be briefly seen that the best results are obtained not at the beginning or at the end of the training process, but at different hyperparameters regardless of time.

Figures 6, 7 and 8 represent the distribution of validation accuracy, average training loss and average validation loss values obtained in different runs compared to the time spent for each run. In these figures, it is clearly seen that the average training loss and average validation loss values change according to the time spent in each run.

Figure 9, on the other hand, represents how long it takes for processes that progress through different batch size, epochs and learning rates, and finally, what validation accuracy values they reach.

Figure 10 represents how much each parameter affects the result. And according to these results, it is observed that runtime is the parameter that has the most effect on the result, followed by epochs, learning rate and batch size parameters respectively.

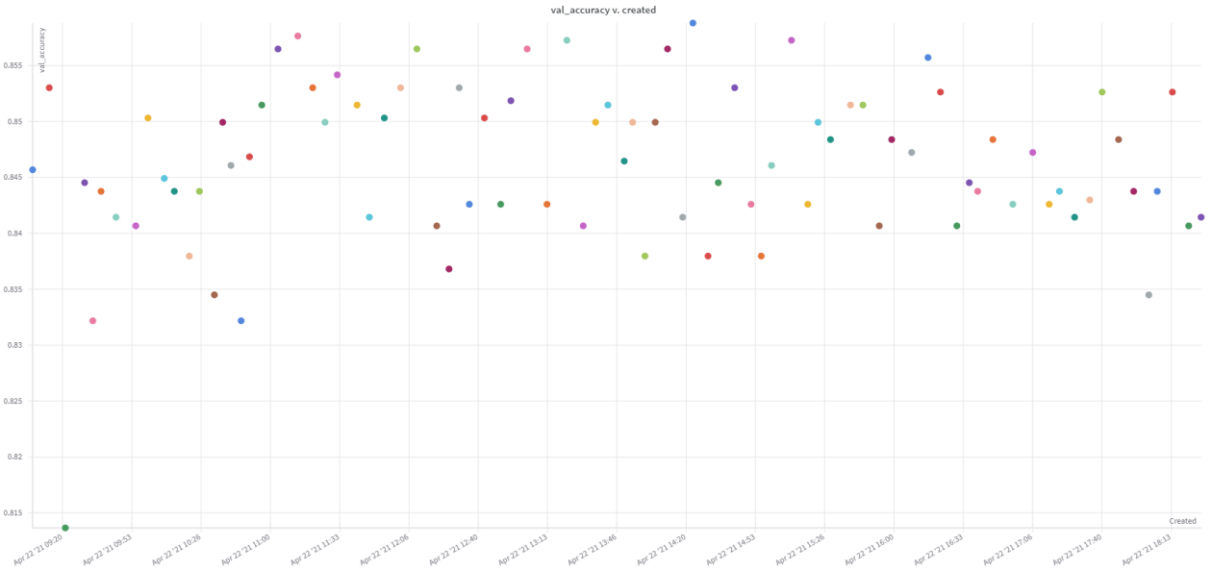


Figure 5: Validation accuracy timeline

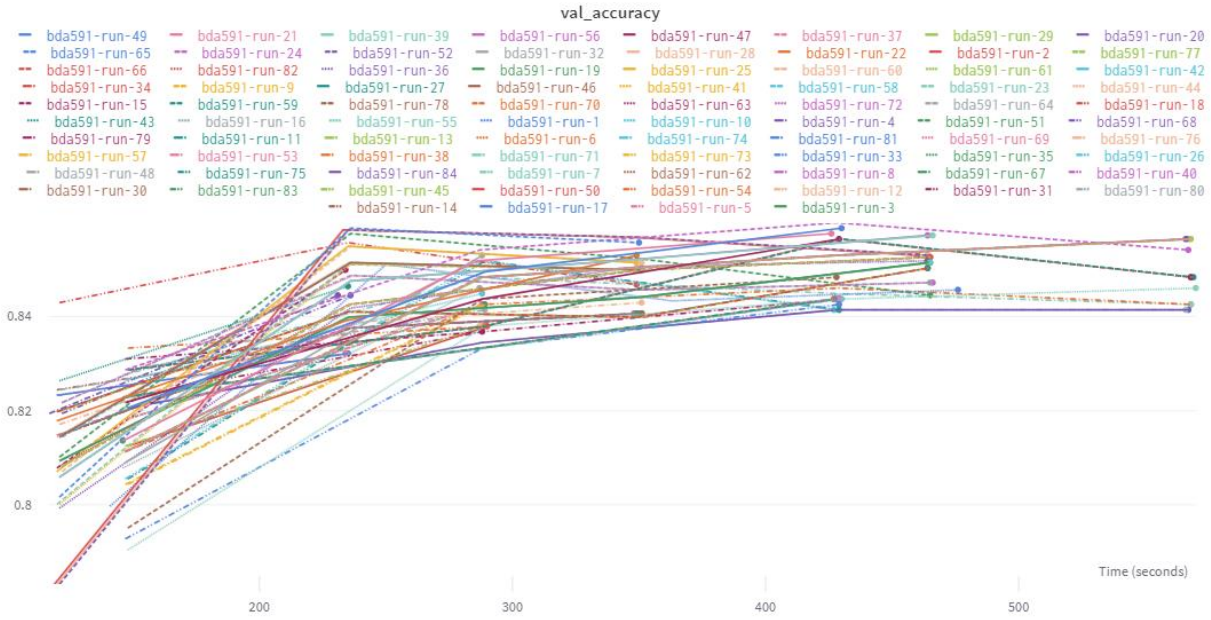


Figure 6: Validation accuracy

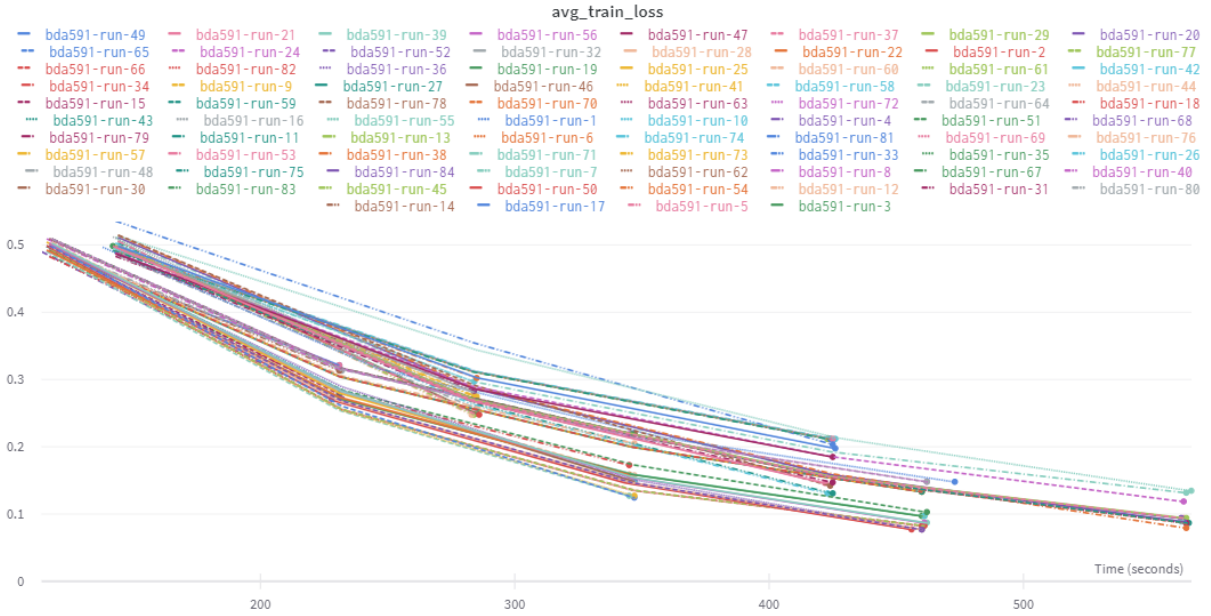


Figure 7: Average training loss

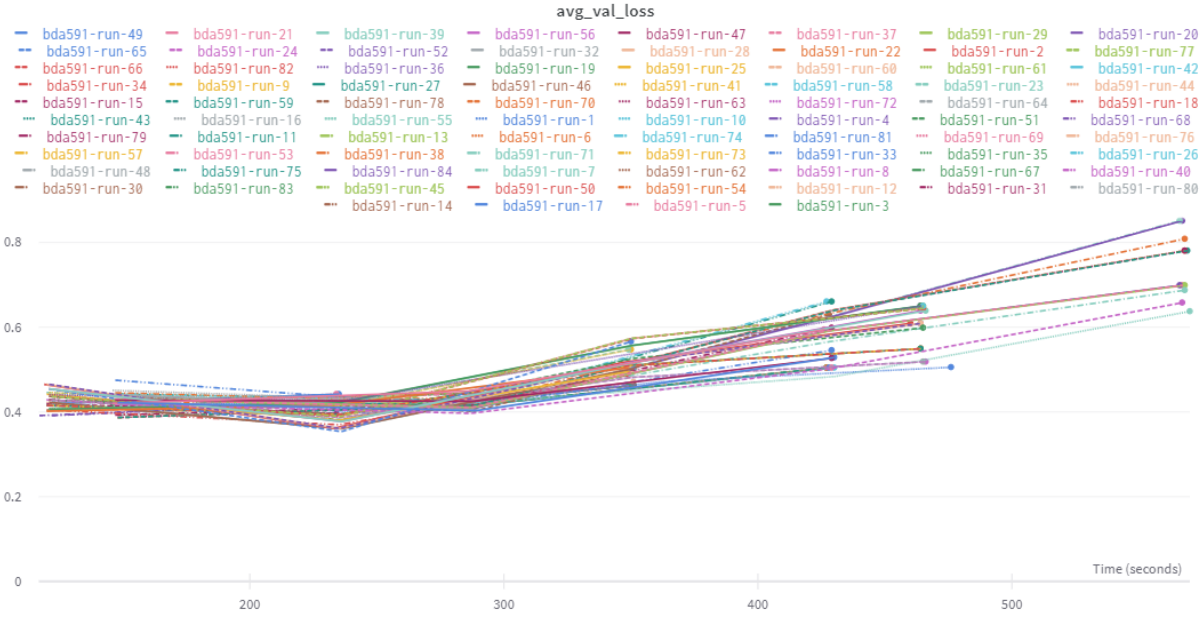


Figure 8: Average validation loss

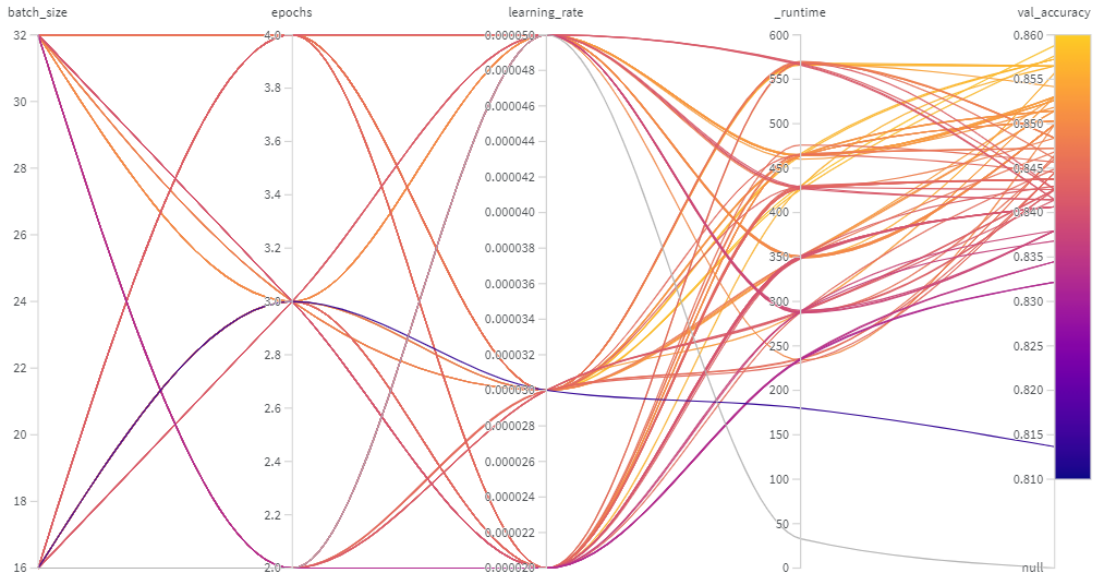


Figure 9: Batch size, epochs, learning Rate

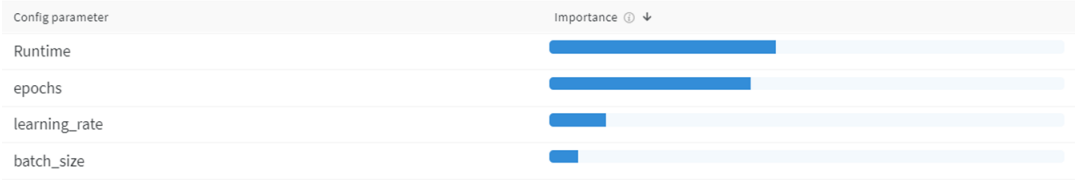


Figure 10: Parameter importance with respect to validation accuracy

4. APPLICATION OF THE MODEL TO SEARCH RESULTS

As the last work of the first phase of the project, the model is applied to the contents that appear in the first 20 positions in the 50 keywords that are important in the field of telecommunications industry, that is, 1000 content in total, and the scores of these contents are obtained. The reason for focusing on keywords in the field of telecommunications can be summarized as possibility of applying the results of this project to daily tasks while working as an SEO expert in this field.

In this phase of the study, the text in the p tags in the page content is scraped over each URL as the first step. Later, these contents are tokenized with nltk. Finally, these tokenized contents are transferred to a data frame and fed to the model.

```
print ('\n-----\n'.join(tokenizer.tokenize(text)))
```

```
Our mission is to enable our clients to turn ideas into action faster.
-----
```

	sentence
0	Our mission is to enable our clients to turn i...
1	Cprime transforms businesses with consulting, ...
2	We believe in a more productive future, where ...
3	Agile software development refers to software...
4	The ultimate value in Agile development is tha...

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 203 entries, 0 to 202
Data columns (total 1 columns):
 #   Column    Non-Null Count  Dtype
---  ------    --------------  -----
 0   sentence  203 non-null    object
dtypes: object(1)
memory usage: 1.7+ KB
```

Figure 11: A sample of the tokenized sentences and df

Tokenized content is then labeled as grammatically correct or incorrect. The BERT score data is created by determining the total number of sentences for each content and the ratio of sentences accepted as correct from the results obtained. This score is included in the data created in the second stage of the project, ie the excel document via vlookup, as a new column and feature.

```
pred_df.groupby("label_predictions").count()
```

label_predictions	sentence
0	11
1	192

sentence	label_predictions
Our mission is to enable our clients to ...	1
The ultimate value in Agile development is ...	1
Agile development refers to any development ...	1
Scrum is a subset of Agile.	1

Figure 12: A sample of scoring the contents

5. THE SECOND PHASE: FEATURE IMPORTANCE

In the second phase of the project, in addition to the ranking values and the BERT scores obtained before, 77 different features are determined and a dataset is created over these data points. These features, which will be introduced in more detail in the following section, are obtained through third party tools. These tools are SEMrush, Ahrefs, Screaming Frog, Page Speed Insights and GTmetrix. In most cases, especially in performance metrics, the most effective options are used by comparing the data obtained through a tool with the results obtained from different tools that can provide data for the same feature.

As mentioned before, these features constitute only a small part of the features that are assumed to be included in the Google algorithm, but the data that can be obtained without access to first party tools such as the administration of the site and webmaster tools are limited with these features.

5.1. Understanding the Data

The dataset studied in the second phase and used to measure the effect of BERT scores on ranking values consists of a total of 79 features, 7 of which are in boolean format.

- Ranking: The ranking value of the page in search results. The position values here are grouped in order to obtain a more effective result due to the size of the dataset. Pages ranked in the first 3 positions are labeled as 1; Other pages ranked between 4 and 10 (most of the time) on the first page are labeled as 2; Pages ranked between 11-20 (most of the time) on the second page are labeled as 3.
- Https: The SSL security of the page. This feature is in boolean format and indicates whether the condition is met or not.
- Domain_Rating: Domain Rating (DR) shows the strength of a target website's backlink profile compared to the others on a 100-point scale.
- Ref_Domains: The total number of unique domains linking to target page.
- Ref_domains_Dofollow: The number of unique referring domains pointing to target page via value-passing links.
- Ref_domains_Governmental: The number of unique governmental domains linking to target page. These include not only .gov TLDs, but also domains that considered to be governmental.

- Ref_domains_Educational: The number of unique educational domains linking to target page. These include not only .edu TLDs, but also domains that considered to be educational.
- Ref_IPs: Shows the number of unique IP addresses with at least one domain pointing to a target website or page. Several domains can share one IP address.
- Ref_SubNets: Shows the number of c-class IP networks (AAA.BBB.CCC.DDD) with at least one link to the target website or page. Example: 151.80.39.61 is the website IP address where 151.80.39.XXX is the subnet.
- Linked_Domains: Shows the total number of unique domains linked from the target website, subsection, or page.
- Total_Backlinks: The total number of links from other websites pointing to target page.
- Backlinks_Text: The number of links in <a href=""> HTML tags pointing to target page. This includes links using an image as an anchor and links with an empty anchor.
- Backlinks_NoFollow: The number of links with the rel="nofollow" attribute pointing to target page.
- Backlinks_Redirect: The number of links pointing to target page via a redirect.
- Backlinks_Image: The number of links pointing to target page where an image is used as the hyperlink.
- Backlinks_Frame: Shows the number of backlinks pointing to the target website, subsection, or page from <iframe src=""> HTML tag.
- Backlinks_Form: The number of links in <form action=""> HTML tags pointing to target page.
- Backlinks_Governmental: The number of links from governmental domains pointing to target page. These include not only .gov TLDs, but also domains that we consider to be governmental.
- Backlinks_Educational: The number of links from educational domains pointing to target page. These include not only .edu TLDs, but also domains that we consider to be educational.
- Total_Keywords: The total number of keywords that target page ranks for in the top 100 organic search results across all countries in database.
- Total_Traffic: The target website, subsection, or page's estimated monthly organic traffic from search.

- Redirect_Chain: The presence of redirection between the result clicked in the search results and the page reached. Pages that are no longer active but have a certain power are redirected to their active counterpart, and in this case, Google will remove the relevant pages from the index over time and replace them with a different page that benefits the user. This feature is in boolean format and indicates whether the condition is met or not.
- Title_Tag_Focus_KW: An evaluation made on the keyword that the page appears in the search results, and whether this keyword is included in the title tag. This feature is in boolean format and indicates whether the condition is met or not.
- Title_Tag_Length: The length of the title tag, in characters.
- Title_Tag_Pixel_Width: The length of the title tag, in pixels.
- Meta_Description_Tag_Focus_KW: An evaluation made on the keyword that the page appears in the search results, and whether this keyword is included in the description tag. This feature is in boolean format and indicates whether the condition is met or not.
- Meta_Description_Length: The length of the Description tag in characters.
- Meta_Description_Pixel_Width: Length of the Description tag in pixels.
- Meta_Keywords_Focus_KW: Although meta keywords are a feature that has been ignored by Google for a long time, it continues to be presented at the source of most pages. In this context, this feature provides the information whether the focus keyword is included in the meta keywords.
- Meta_Keywords_Length: The total length, in characters, of the content presented in the meta keywords field.
- H1_Focus_KW: An evaluation made on the keyword that the page appears in the search results, and whether this keyword is included in the h1 tag. This feature is in boolean format and indicates whether the condition is met or not.
- H1_Length: The length of the H1 tag in characters.
- Second_H1: Information on whether a second H1 tag exists. This feature is in boolean format and indicates whether the condition is met or not.
- H2_Focus_KW: Information on whether there is an H2 tag on the page and whether this tag contains the focus keyword. This feature is in boolean format and indicates whether the condition is met or not.
- H2-1_Length: The length of the first H2 tag, in characters, if the H2 tag or tags exist.

- H2-2_Length: The length of the second H2 tag, in characters, if the H2 tag or tags exist.
- Size_(bytes): The total size of the page, in bytes.
- Word_Count: The number of all words inside the body tag of the page, excluding HTML markup.
- Text_Ratio: Number of non-HTML characters found in the HTML body tag on a page (the text), divided by the total number of characters the HTML page is made up of, and displayed as a percentage.
- Inlinks: Number of internal hyperlinks to the page. 'Internal inlinks' are links pointing to a given page from the same subdomain that is being crawled.
- Unique_Inlinks: Number of 'unique' internal inlinks to the page. 'Internal inlinks' are links pointing to a given page from the same subdomain that is being crawled. For example, if 'page A' links to 'page B' 3 times, this would be counted as 3 inlinks and 1 unique inlink to 'page B'.
- Outlinks: Number of internal outlinks from the page. 'Internal outlinks' are links from a given page to other pages on the same subdomain that is being crawled.
- Unique_Outlinks: Number of unique internal outlinks from the page. 'Internal outlinks' are links from a given page to other pages on the same subdomain that is being crawled. For example, if 'page A' links to 'page B' on the same subdomain 3 times, this would be counted as 3 outlinks and 1 unique outlink to 'page B'.
- External_Outlinks: Number of external outlinks from the page. 'External outlinks' are links from a given page to another subdomain.
- Unique_External_Outlinks: Number of unique external outlinks from the page. 'External outlinks' are links from a given page to another subdomain. For example, if 'page A' links to 'page B' on a different subdomain 3 times, this would be counted as 3 external outlinks and 1 unique external outlink to 'page B'.
- Response_Time: This feature measures how long it takes to load the necessary HTML to begin rendering the page
- Performance_Score: This score is determined by running Lighthouse to collect and analyze lab data about the page.
- First_Contentful_Paint_Time_(sec): FCP measures how long it takes the browser to render the first piece of DOM content after a user navigates to the page.

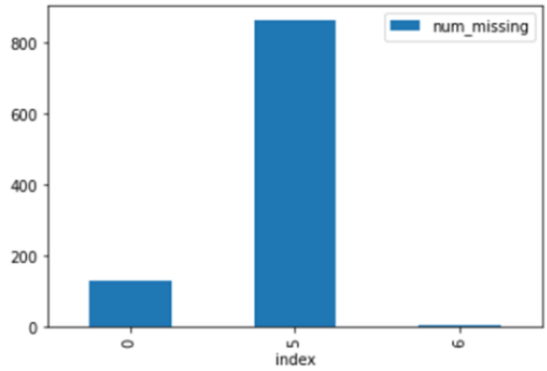
- Speed_Index_Time_(sec): Speed Index measures how quickly content is visually displayed during page load.
- Largest_Contentful_Paint_Time_(sec): The Largest Contentful Paint (LCP) metric reports the render time of the largest image or text block visible within the viewport, relative to when the page first started loading.
- Time_to_Interactive_(sec): TTI measures how long it takes a page to become fully interactive.
- Total_Blocking_Time_(ms): TBT measures the total amount of time that a page is blocked from responding to user input, such as mouse clicks, screen taps, or keyboard presses.
- Cumulative_Layout_Shift: CLS measures the sum total of all individual layout shift scores for every unexpected layout shift that occurs during the entire lifespan of the page.
- Total_Size_Savings: Total size savings that can be achieved through optimizations to be applied to improve performance.
- Total_Time_Savings_(ms): Total time savings that can be achieved through optimizations to be applied in order to improve performance.
- Total_Requests: The total number of requests during the loading process of the page. This number includes js, css, font, image and all other resources.
- Total_Page_Size: The total file size of the page in bytes
- HTML_Size: The total size of html resources on the page is included in the page's file size.
- HTML_Count: The total number of html resources on the page is included in the number of requests during the loading of the page.
- Image_Size: The total size of image resources on the page is included in the page's file size.
- Image_Count: The total number of image resources on the page is included in the number of requests during the loading of the page.
- CSS_Size: The total size of the css resources on the page is included in the file size of the page.
- CSS_Count: The total number of css resources on the page is included in the number of requests during the loading of the page.
- JavaScript_Size: The total size of javascript resources on the page is included in the page's file size.
- JavaScript_Count: The total number of javascript resources on the page is included in the number of requests during the loading of the page.

- Font_Size: The total size of font resources on the page is included in the page's file size.
- Font_Count: The total number of font resources on the page is included in the number of requests during the loading of the page.
- Media_Size: The total size of media resources on the page is included in the page's file size.
- Media_Count: The total number of media resources on the page is included in the number of requests during the loading of the page.
- Other_Size: The total size of the resources on the page other than html, image, css, javascript, font and media is included in the file size of the page.
- Other_Count: The total number of resources on the page other than html, image, css, javascript, font and media is included in the number of requests during the loading of the page.
- Third_Party_Size: The total size of third party resources on the page is included in the file size of the page.
- Third_Party_Count: The total number of third party resources on the page is included in the number of requests during the loading of the page.
- Core_Web_Vitals_Assessment: Core Web Vitals will be part of Google's page experience score to size up a page's overall UX and they are a set of specific factors (that considered as important in a webpage's overall user experience) will be rolled out in mid-June 2021. (Expected to be rolled out int May 2021 but postponed.)
- CrUX_First_Contentful_PaintTime_(sec): FCP metric provided by Chrome User Experience Report. The Chrome User Experience Report provides user experience metrics for how real-world Chrome users experience popular destinations on the web.
- CrUX_First_Input_DelayTime_(ms): FID metric provided by Chrome User Experience Report. The Chrome User Experience Report provides user experience metrics for how real-world Chrome users experience popular destinations on the web.
- CrUX_Largest_Contentful_PaintTime_(sec): LCP metric provided by Chrome User Experience Report. The Chrome User Experience Report provides user experience metrics for how real-world Chrome users experience popular destinations on the web.
- CrUX_Cumulative_Layout_Shift: CLS metric provided by Chrome User Experience Report. The Chrome User Experience Report provides user experience metrics for how real-world Chrome users experience popular destinations on the web.
- BERT_Score: BERT score obtained for each pages in the first phase of the project.

5.2. Exploratory Data Analysis

Analysis and optimizations within the scope of EDA are started by checking the null values first. In this context, when the number of null values in each column is analyzed, it is detected that there are over 800 null values in 5 of the features. Since replacing this amount of null values will negatively affect the dataset, it is decided to remove these columns from the dataset. The fact that Core Web Vitals Assessment, CrUX First Contentful Paint Time (sec), CrUX First Input Delay Time (ms), CrUX Largest Contentful Paint Time (sec) and CrUX Cumulative Layout Shift columns (with 867 null values each) have such a high number of problems is an expected result. Considering that third party tools used within the scope of the project will not be able to perform an effective core web vitals assesment for each page and cannot access Chrome UX data, it is an expected result.

After the 5 columns with a high number of null values are removed from the dataset, the other columns with a small number of null values are filled with mean values. As a result, all null values in the dataset are cleared.




```
df1.isnull().sum().sum()
```

```
0
```

Figure 13: The number of null values before and after

After clearing the null values, the EDA processes continue with the detection and optimization of the outliers. Some features in the dataset that do not have a certain upper limit, especially the values within the scope of the backlink, can show great differences between strong and weak domains. For this reason, a high number of outliers in the dataset is also a possible result.

For this, first of all, the table containing min, max, quantile, standard deviation and mean values of all columns in the data is printed and reviewed. Later, some features were analyzed separately to be examined in more detail.

	Domain Rating	Ref Domains	Ref domains Dofollow	Ref domains Governmental	Ref domains Educational
count	1000.000000	1.000000e+03	1.000000e+03	1000.000000	1000.000000
mean	73.044400	8.993896e+04	8.100665e+04	98.191000	564.945000
std	21.835615	2.584154e+05	2.338738e+05	320.291907	1412.362551
min	0.000000	6.000000e+00	2.000000e+00	0.000000	0.000000
25%	68.000000	1.518250e+03	1.098000e+03	0.000000	0.000000
50%	78.000000	4.755000e+03	3.968000e+03	3.000000	42.000000
75%	90.000000	1.510100e+04	1.244300e+04	26.000000	155.000000
max	98.000000	1.365662e+06	1.181159e+06	3934.000000	6465.000000

Figure 14: Description of the dataset

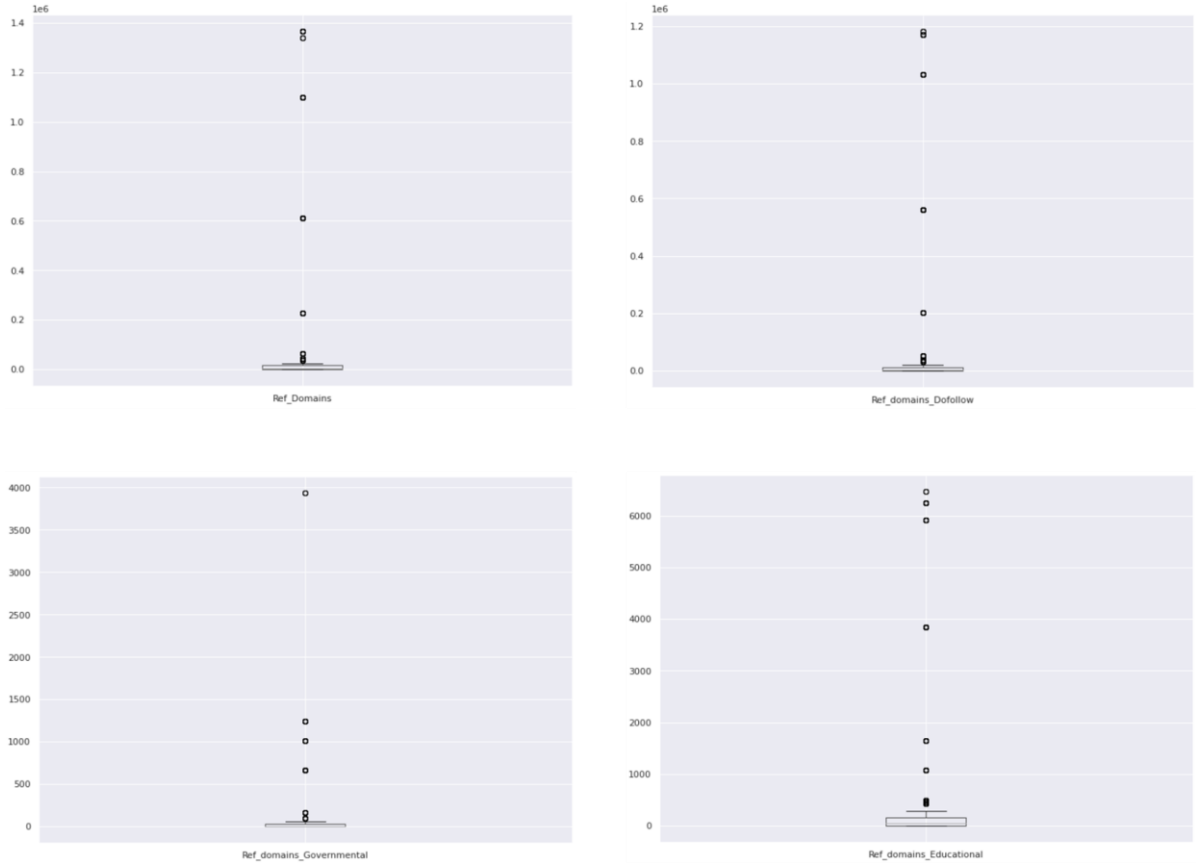


Figure 15: Outliers in backlink focused features

After a detailed examination of the values in the dataset, it is decided that there are outliers in 41 columns and they need to be cleaned. In this context, all values above 0.95 quantile in the relevant columns are equalized to 0.95, and all values below 0.05 quantile are equalized to 0.05. The reason why .90 and 0.1 values are not selected here is that it is not desired to get rid of all the values seen as outlier. Because, in these features, it is aimed to protect the superiority of pages with extreme values against other pages within the dataset. After most of the outliers have been cleared, all features are also standardized by importing StandardScaler from sklearn.preprocessing.

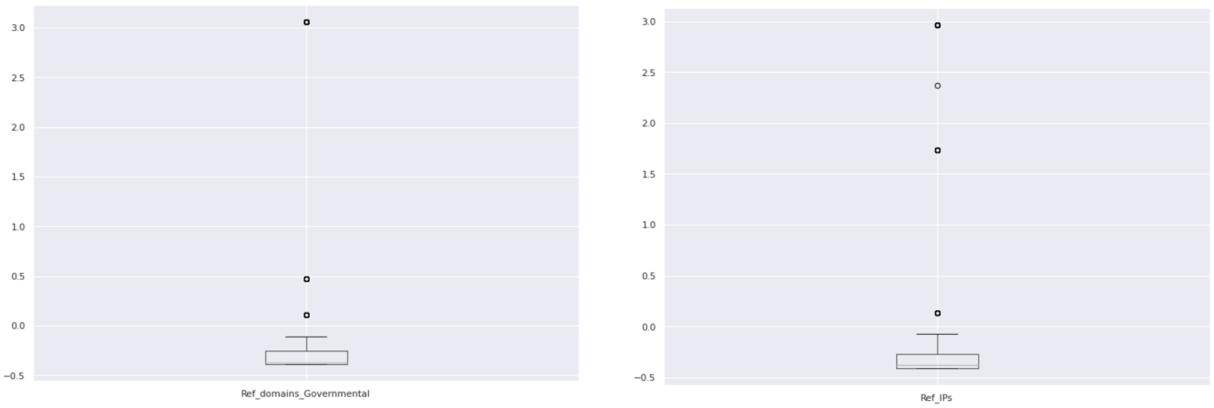


Figure 16: Outliers in backlink focused features after EDA

6. MODEL SET UP AND RESULTS

In this section, after clearing the data and completing the exploratory data analysis processes, the project continues with the selection and application of the models that can give the best results. In this context, the focus is primarily on supervised learning models, since data has labels, namely ranking values. Supervised learning is based on training a data sample from data source with correct classification already assigned (Sathya et al., 2013). Then, classification models are preferred in line with the purpose of the model. As will be explained in more detail in the following section, K Neighbors Classifier and SVM are preferred, but since the kernel functions are one of the major tricks of SVM (Amami et al., 2020)., 3 different types of kernel functions, SVM/kernel=poly (will be referred to as SVM Poly in the following section), SVM/kernel=rbf (will be referred to as SVM RBF in the following section) and SVM/kernel=sigmoid (will be referred to as SVM Sigmoid in the following section) are tested.

In addition to K Neighbor, which is frequently used in such tasks, SVM is chosen mainly because it is an algorithm that can be used in both classification and regression tasks and gives successful results. However, it is observed that none of the models can exceed 0.65 accuracy. For this reason, the ranking labels are reduced from 3 to 2 and it is seen that the accuracy score goes above 0.75 in all models, and 0.9 level in Poly and RBF models. For this reason, it is decided to move forward by reducing the number of labels to 2. In this context, instead of labeling 1-3, 4-10, 11-20 position ranges differently, pages that are ranked on the first page, that is in the range of 1-10 and on the second page, that is in the range of 11-20, are labeled as 1 and 2. Before the models are tested and feature importance is evaluated, the distribution of the BERT score according to the labels is examined and it is observed whether there is a visible difference in this distribution.

When the results are examined, it is observed that most of the results on the second page are below the score of 88, and the results on the first page are particularly dominant above this score.

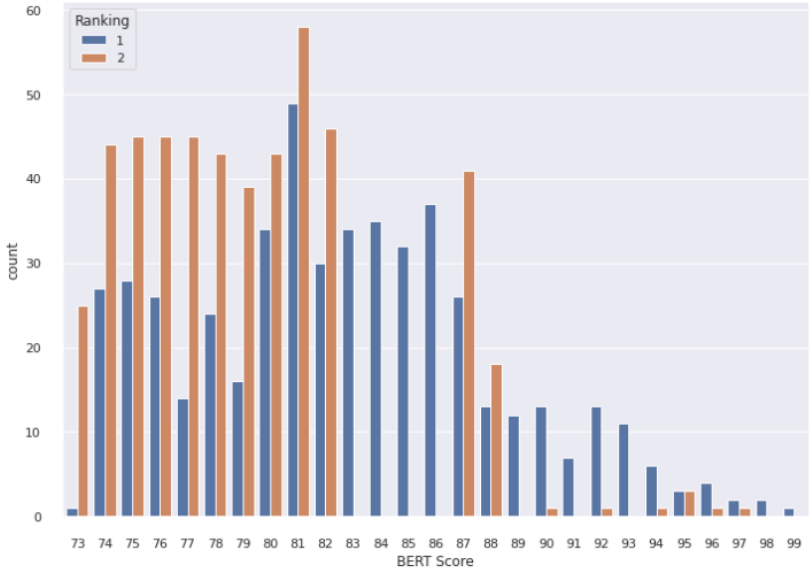


Figure 17: The distribution of the BERT score

6.1. KNeighborsClassifier

KNeighborsClassifier is chosen as the first model to test. As the first step, 85% of the data is split for training and 15% for testing to apply the model to the data. The n_neighbors value is set to 1. The reason for this is that the most effective result is obtained with a value of 1 in studies performed on different n_neighbors values. After that, two different models are run first with cross validation and then without cross validation. As a result, in the model where the cross validation value is set as 50, the average accuracy score is measured at the level of 0.79. In the model that is run without cross validation, the accuracy score is also measured as 0.79. Although the accepted approach is generally to take more than 1 neighbor (Cunningham et al., 2015), in this model, 1 neighbor provides the best result in tests performed with different parameters.

```
[[57 14]
 [18 61]]
              precision    recall  f1-score   support

           1       0.76      0.80      0.78        71
           2       0.81      0.77      0.79        79

    accuracy                           0.79       150
   macro avg       0.79      0.79      0.79       150
weighted avg       0.79      0.79      0.79       150
```

Figure 18: The results of the K Neighbors Classifier

6.2. SVM Poly

The second model tested is SVM Poly and analysis is started by splitting test and train sets first. However, this time, unlike KNeighbors, 80% of the data is split for train and 20% for testing. The reason for the ratio change is that the 80/20 ratio gives a better result compared to the 85/15 ratio in this model. The model is run over these parameters: C=1.0, break_ties=False, cache_size=200, class_weight=None, coef0=0.0, decision_function_shape='ovr', degree=3, gamma='scale', kernel='poly', max_iter=-1, probability=False, random_state=None, shrinking=True, tol=0.001, verbose=False, and the accuracy score reaches 0.91.

```
[[90  8]
 [10 92]]
              precision    recall  f1-score   support

           1       0.90      0.92      0.91        98
           2       0.92      0.90      0.91       102

    accuracy                           0.91       200
   macro avg       0.91      0.91      0.91       200
weighted avg       0.91      0.91      0.91       200
```

Figure 19: The results of the SVM Poly

6.3. SVM RBF

The third model is SVM's RBF kernel function. In this model, the same parameters are preferred as the previous Poly model, but only the C value, that is the regularization parameter, is set as 1000. The reason for this is that the best result is obtained with a value of 1000 in the tests performed between 1 and 1000 (1-10-100-1000). As a result, an accuracy score of .92 is obtained in the SVM RBF model.

```
[[91  7]
 [10 92]]
              precision    recall  f1-score   support

           1       0.90      0.93      0.91        98
           2       0.93      0.90      0.92       102

    accuracy                           0.92       200
   macro avg       0.92      0.92      0.91       200
weighted avg       0.92      0.92      0.92       200
```

Figure 20: The results of the SVM RBF

6.4. SVM Sigmoid

As a result of the high accuracy scores obtained in SVM models, SVM Sigmoid is chosen as the last choice. In this model, too, exactly the same parameters are selected as RBF, and the 0.88 accuracy score is measured.

```
[[86 12]
 [12 90]]
              precision    recall  f1-score   support

           1       0.88      0.88      0.88        98
           2       0.88      0.88      0.88       102

    accuracy                           0.88       200
   macro avg       0.88      0.88      0.88       200
weighted avg       0.88      0.88      0.88       200
```

Figure 21: The results of the SVM Sigmoid

Table 1: The comparison of the results from different models

Model	**Accuracy Score**
K Neighbors Classifier	0.79
SVM Poly	0.91
SVM RBF	0.92
SVM Sigmoid	0.88

6.5. Grid Search

At this stage, after trying different SVM models suggested in the literature, grid search is applied over the following parameters in order to obtain the most effective result and in addition to observe the benefit of grid search.

{'kernel': ['rbf'], 'gamma': [1e-3, 1e-4, 4e-4], 'C': [1, 10, 100, 1000], 'degree': [1, 2, 3]},
{'kernel': ['poly'], 'gamma': [1e-3, 1e-4, 4e-4], 'C': [1, 10, 100, 1000], 'degree': [1, 2, 3]},
{'kernel': ['sigmoid'], 'gamma': [1e-3, 1e-4, 4e-4], 'C': [1, 10, 100, 1000], 'degree': [1, 2, 3]}

As a result, the following scores are obtained, and it is determined that the best result is 0.942 (+/-0.042) and this score is obtained from the model run with {'C': 100, 'degree': 1, 'gamma': 0.0004, 'kernel': 'rbf'} parameters.

Table 2: Top 50 results obtained from grid search

Accuracy	C	Degree	Gamma	Kernel
0.942 (+/-0.042)	100	1	0.0004	rbf
0.942 (+/-0.042)	100	2	0.0004	rbf
0.942 (+/-0.042)	100	3	0.0004	rbf
0.940 (+/-0.055)	10	1	0.0004	rbf
0.940 (+/-0.055)	10	2	0.0004	rbf
0.940 (+/-0.055)	10	3	0.0004	rbf
0.936 (+/-0.049)	10	1	0.001	poly
0.936 (+/-0.049)	100	1	0.0001	poly
0.936 (+/-0.049)	100	1	0.0001	sigmoid
0.936 (+/-0.049)	100	2	0.0001	sigmoid
0.936 (+/-0.049)	100	3	0.0001	sigmoid
0.935 (+/-0.048)	10	1	0.001	rbf
0.935 (+/-0.048)	10	2	0.001	rbf
0.935 (+/-0.048)	10	3	0.001	rbf
0.935 (+/-0.047)	10	1	0.001	sigmoid
0.935 (+/-0.047)	10	2	0.001	sigmoid
0.935 (+/-0.047)	10	3	0.001	sigmoid
0.935 (+/-0.045)	100	1	0.001	poly
0.935 (+/-0.045)	1000	1	0.0001	poly
0.935 (+/-0.045)	1000	1	0.0001	sigmoid
0.935 (+/-0.045)	1000	2	0.0001	sigmoid
0.935 (+/-0.045)	1000	3	0.0001	sigmoid
0.934 (+/-0.052)	100	1	0.0004	poly
0.934 (+/-0.052)	100	1	0.0004	sigmoid
0.934 (+/-0.052)	100	2	0.0004	sigmoid
0.934 (+/-0.052)	100	3	0.0004	sigmoid
0.934 (+/-0.037)	100	1	0.001	rbf
0.934 (+/-0.037)	100	2	0.001	rbf
0.934 (+/-0.037)	100	3	0.001	rbf
0.933 (+/-0.057)	10	1	0.0004	poly
0.933 (+/-0.057)	10	1	0.0004	sigmoid
0.933 (+/-0.057)	10	2	0.0004	sigmoid
0.933 (+/-0.057)	10	3	0.0004	sigmoid
0.932 (+/-0.046)	1000	1	0.0004	rbf
0.932 (+/-0.046)	1000	2	0.0004	rbf
0.932 (+/-0.046)	1000	3	0.0004	rbf
0.931 (+/-0.048)	1000	1	0.0001	rbf
0.931 (+/-0.048)	1000	2	0.0001	rbf
0.931 (+/-0.048)	1000	3	0.0001	rbf
0.930 (+/-0.052)	100	1	0.001	sigmoid
0.930 (+/-0.052)	100	2	0.001	sigmoid
0.930 (+/-0.052)	100	3	0.001	sigmoid
0.929 (+/-0.044)	1000	1	0.0004	poly
0.929 (+/-0.044)	1000	1	0.0004	sigmoid
0.929 (+/-0.044)	1000	2	0.0004	sigmoid
0.929 (+/-0.044)	1000	3	0.0004	sigmoid
0.928 (+/-0.050)	100	1	0.0001	rbf
0.928 (+/-0.050)	100	2	0.0001	rbf
0.928 (+/-0.050)	100	3	0.0001	rbf

6.6. The Results and Discussion

As the last step, feature importance analysis is performed on the SVM RBF model with the best parameters obtained from grid search, in which the highest accuracy score is reached, because accuracy is among the common methods used for comparing performance of one algorithm over the other (Omary et al., 2010).

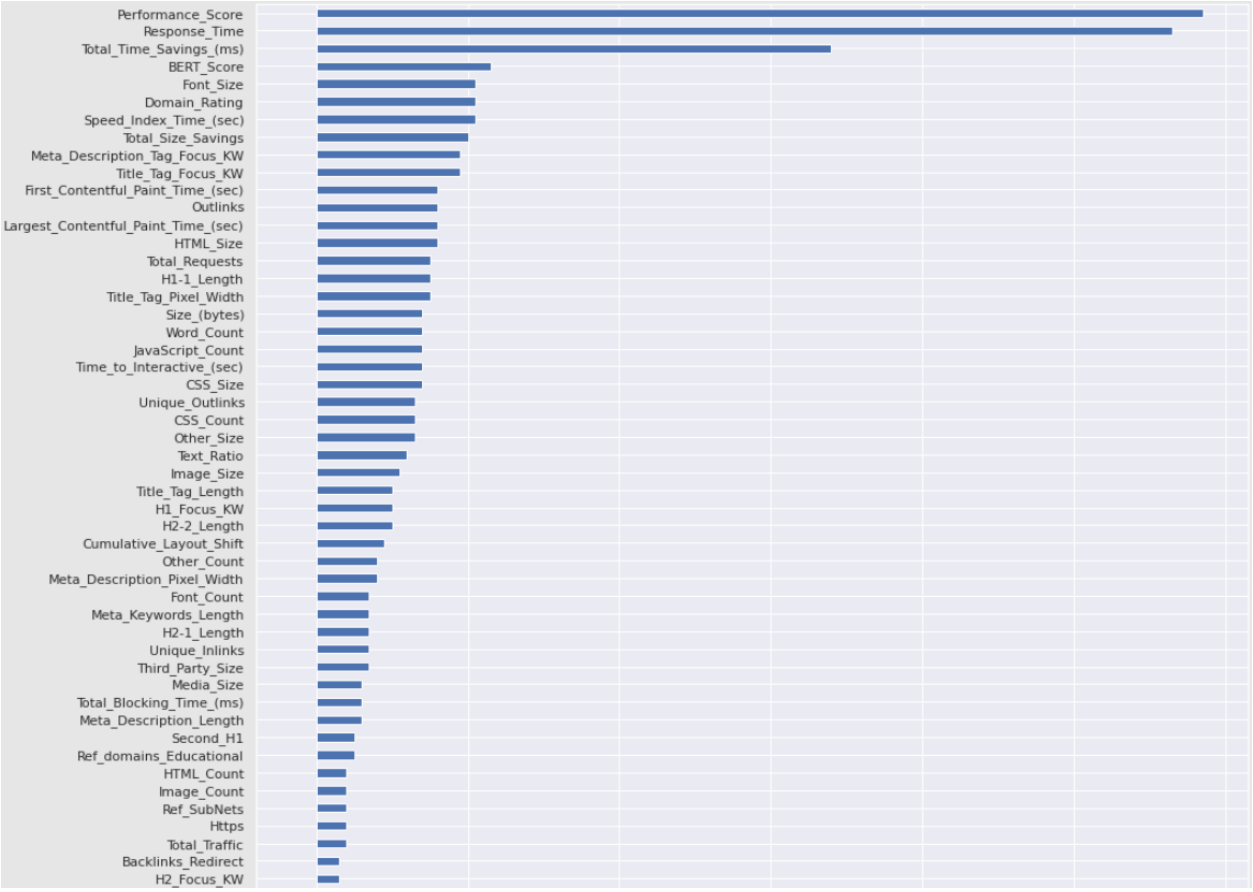


Figure 22: The most important 50 features

According to this analysis, it is seen that the most important features are the Performance Score and Response Time values. Since these values are factors that directly affect user experience, it is a known fact that they directly affect the visibility of the pages, so the first two results are not surprising when evaluated within the scope of domain knowledge. Similarly, it is an expected result that the Speed Index Time and Domain Rating features are among the most important features. The Title Tag Focus Keyword feature is also a factor that is assumed to affect the search results, so it is also an expected result that this feature is in the top 10.

It is not surprising that the other features in the top 10 and not yet mentioned are also on this list, because it is already known that these features are among the important criteria due to the SEO experience and the project managed over the years. However, the fact that the Total Time Savings feature is in the top 3 can be accepted as an unexpected result. Google is known to place higher value on key criteria that affect the user experience, but it is a surprising result that a feature showing possible optimizations that could strengthen the page might be accepted as more important than the current scores the page has. The fact that the BERT score is in the top 10 can be accepted as an expected result due to the correlation observed when the distribution showing the relationship between the BERT score and the ranking values was evaluated before. At the same time, it is a result that will enable us to accept the results obtained in the project as successful despite the small size of the dataset.

If an additional analysis is desired to be applied to this project in the next stage, this work may be to add Core Web Vitals oriented values to the features. Core Web Vitals, which will be rolled out by mid-June 2021, will have a great impact on the search results, however, although it is desired to work on the relevant features in this project, due to the inadequacy of the third party tools at this stage and unstable results, a critical amount of missing data error has been encountered and these features were removed from the dataset.

REFERENCES

[1] Anuj Joshi and Priyanka Patel (2018). *Google Page Rank Algorithm and It's Updates*. Retrieved from https://www.researchgate.net/publication/328529814_Google_Page_Rank_Algorithm_and_It's_Updates

[2] Konstantina Dritsa, Thodoris Sotiropoulos, Haris Skarpetis, and Panos Louridas (2020). *Search Engine Similarity Analysis: A Combined Content and Rankings Approach*. Retrieved from https://arxiv.org/pdf/2011.00650.pdf

[3] Search Engine Optimization (SEO) Starter Guide. (n.d.). Retrieved from https://developers.google.com/search/docs/beginner/seo-starter-guide

[4] Search Quality Evaluator Guideline (2020). Retrieved from https://static.googleusercontent.com/media/guidelines.raterhub.com/en//searchqualityevaluatorguidelines.pdf

[5] Taeuk Kim, Jihun Choi, Daniel Edmiston & Sang-goo Lee (2020). *Are Pre-trained Language Models Aware of Phrases? Simple but Strong Baselines for Grammar Induction.* Retrieved from https://arxiv.org/pdf/2002.00737.pdf

[6] Jacob Devlin, Ming-Wie Chang, Kenton Lee, Kristina Toutanova (2019). *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding.* Retrieved from https://arxiv.org/pdf/1810.04805.pdf

[7] Anna Rogers, Olga Kovaleva, Anna Rumshisky (2020). *A Primer in BERTology: What We Know About How BERT Works.* Retrieved from https://arxiv.org/pdf/2002.12327.pdf

[8] Alex Warstadt, Amanpreet Singh and Samuel R. Bowman (2019). *Neural Network Acceptability Judgments.* Retrieved from https://www.mitpressjournals.org/doi/full/10.1162/tacl_a_00290

[9] Alex John Quijano, Sam Nguyen, Juanita Ordonez (2021). *Grid Search Hyperparameter Benchmarking of BERT, ALBERT, and Longformer on DuoRC.* Retrieved from https://arxiv.org/pdf/2101.06326.pdf

[10] Chi Sun, Xipeng Qiu, Yige Xu and Xuanjing Huang (2020). *How to Fine-Tune BERT for Text Classification?* Retrieved from https://arxiv.org/pdf/1905.05583.pdf

[11] Frank Hutter, Jörg Lücke, and Lars Schmidt-Thieme (2015). *Beyond manual tuning of hyperparameters*. Retrieved from https://www.researchgate.net/publication/282539639_Beyond_Manual_Tuning_of_Hyperparameters

[12] R. Sathya and Annamma Abraham (2013). *Comparison of Supervised and Unsupervised Learning Algorithms for Pattern Classification.* Retrieved from https://thesai.org/Downloads/IJARAI/Volume2No2/Paper_6-Comparison_of_Supervised_and_Unsupervised_Learning_Algorithms_for_Pattern_Classification.pdf

[13] Rimah Amami, Dorra Ben Ayed and Noureddine Ellouze (2020). *Practical Selection of SVM Supervised Parameters with Different Feature Representations for Vowel Recognition*. Retrieved from https://arxiv.org/ftp/arxiv/papers/1507/1507.06020.pdf

[14] Pádraig Cunningham and Sarah Jane Delany (2020). *k-Nearest Neighbour Classifiers 2nd Edition (with Python examples)*. Retrieved from https://arxiv.org/pdf/2004.04523.pdf

[15] Zanifa Omary and Fredrick Mtenzi (2010). *Machine Learning Approach to Identifying the Dataset Threshold for the Performance Estimators in Supervised Learning*. Retrieved from https://www.researchgate.net/publication/228548543_Machine_Learning_Approach_to_Identifying_the_Dataset_Threshold_for_the_Performance_Estimators_in_Supervised_Learning