**Predicting Popularity of Open Datasets**

CIND820 Final Project

Patrick Little

Contents

[Introduction 2](#_Toc68249512)

[Literature Review 2](#_Toc68249513)

[Movie Success Prediction Using ML 2](#_Toc68249514)

[Predicting the Number of Downloads of Open Datasets by Naïve Bayes Classifier 3](#_Toc68249515)

[Predicting the popularity of online news from content metadata 4](#_Toc68249516)

[Dataset 5](#_Toc68249517)

[Metadata Catalogue 5](#_Toc68249518)

[Open Government Analytics - Downloads per organization, last 12 months 8](#_Toc68249519)

[Approach 9](#_Toc68249520)

[Step 1: Loading the Data 10](#_Toc68249521)

[Step 2: Data Preparation and Feature Extraction 10](#_Toc68249522)

[Step 3: Exploratory Data Analysis 11](#_Toc68249523)

[Step 4: Model Building 11](#_Toc68249524)

[Step 5: Model Evaluation 12](#_Toc68249525)

[References 12](#_Toc68249526)

Predicting Popularity of Open Datasets

# Introduction

Open.Canada.ca, the Government of Canada’s Open Data Portal contains over 85,000 open datasets or open information resources. These datasets are published by many different government organizations and covers subject matter on a variety of topic areas. On Open.Canada.ca certain datasets receive several thousand downloads per month, while others receive little to no usage on a monthly basis.

Having the ability to predict the popularity of a dataset at the time of publication would enable open data publishers to surface the most relevant and in demand content to users on the open data portal, as well as determine which newly released datasets to promote via other channels such as social media.

In the last 12 months an average of 365 new open datasets were released each month on Open.Canada.ca[[1]](#footnote-1). The velocity of data release means that it would require significant effort from a person to monitor the release of all these new datasets and use their intuition or some other heuristic to determine what newly released datasets to promote or recommend to users. As such, this problem is well suited to be augmented with a predictive model that can identify newly published datasets at the time of publication which are likely to be popular.

The aim of my project was to develop a classifier to analyze the metadata of a newly published dataset on Open.Canada.ca and predict a popularity category that will correspond to the popularity of the dataset as determined by the number of downloads.

The analysis within this paper was conducted using R, and leverages the tidymodels framework for model building, training, and evaluation.

# Literature Review

While researching the problem I was able to find and reference a significant body of academic work relating to predicting popularity based on attributes such as metadata.

## Movie Success Prediction Using ML

In this paper Darapaneni et al. examine the performance of several machine learning algorithms in predicting the success of a movie, based on data from IMDB. Using the well-known Kaggle IMDB movie dataset, the authors engineered seven data features, based on the data, then assigned a weightage to each feature. The example the authors provide is “weightage for director = total movie success by the director / total movies directed.”[[2]](#footnote-2) In addition to engineering the predictor variables, the authors engineered a binary target variable based on 3 aspects of the movie’s performance including IMDB rating and commercial success.

After their feature engineering, the authors reported the following model accuracy by algorithm.

|  |  |
| --- | --- |
| Model | Accuracy |
| KNN | 82.04 |
| Random Forrest | 88.57 |
| Decision Tree | 88.83 |
| XGBoost | **90.39** |
| Gaussian Naïve Bayes | 80.91 |
| Simple Neural Network Model | 80.41 |

Based on the results of Darapaneni et al. the XGBoost algorithm will be considered for use as the classifier to predict the popularity of open datasets during this research.

## Predicting the Number of Downloads of Open Datasets by Naïve Bayes Classifier

In this paper, Šlibar details her success in using a Naïve Bayes Classifier to predict the number of downloads and page views open datasets from the UK government’s open data portal will receive. This paper is particularly relevant to the research as part of this project for several reasons. Firstly the domain similarity is quite high as both data.gov.uk and Open.Canada.ca are both national government open data portals, from countries that are highly regarded in terms of open data. For example, The World Wide Web Foundation’s Open Data Barometer ranks Canada and the UK as tied for 1st place in a ranking of 30 countries.[[3]](#footnote-3) Secondly the underlying data catalog software used on data.gov.uk and Open.Canada.ca is both CKAN, therefore the metadata attributes collected for each dataset are very closely aligned.

In her analysis Šlibar used the following predictor variables: ID, title, description, publisher, update frequency, licence, URL, domain, and machine-readability score.[[4]](#footnote-4) These variables contain factor, binary, or ordinal data. The author uses both number of page views and number of downloads as the target variables. The author then uses the k-means algorithm to bin the numbers of downloads and the number of pageviews into an optimal number of clusters. Based on this approach the author was able to achieve a model accuracy of 69.85% with the Naïve Bayes Classifier.

One critique of the author’s research is the fact that no explanation was given as to the predictive value of any of the individual metadata elements. For example, the author states that she encoded title, description, and dataset URL as true if any value was present, or false if this metadata element was missing from a dataset record. The author did not give any indication of a frequency distribution for those elements and having some domain knowledge of the software used to generate this metadata, it seems likely that those metadata elements would have been configured as mandatory metadata that must be populated by the data publisher. If these elements were all populated as true it seems the predictive value of those metadata attributes would be limited.

In section 4.2 of her paper, Šlibar describes her data collection methodology. Using the CKAN API the author uses the /action/package\_show/{ID} procedure to collect the metadata for each dataset, based of the IDs of the datasets contained in the dataset with the downloads and pageviews data. This approach will not return metadata information for datasets that received zero downloads or pageviews, and therefore it is likely that the author’s results may have limited predictive ability to predict downloads or pageviews for datasets that would receive little usage.

The CKAN API has a command /action/package\_list/ which returns the full list of dataset IDs within the catalogue.[[5]](#footnote-5) A more robust approach might be to use the /action/package\_list/ API call then use the /action/package\_show/ API call on each ID within the list. This would return metadata for all datasets within the catalogue, not only those which received some usage.

## Predicting the popularity of online news from content metadata

In this paper, Uddin et al. summarize that when predicting the popularity of online content such as news, there are generally two predictive scenarios. The first scenario is the post publication scenario, where early user behavior is measured via metrics such as social media interactions, comments analysis, and other clickstream data. The authors explain that the research in online content popularity prediction is dominated by this first scenario.[[6]](#footnote-6)The second scenario is the pre-publication scenario, where the only data available is the metadata of the content. The authors describe this scenario as a more challenging predictive scenario.

In this paper the authors use Gradient Boosting in a binary classification mode to predict whether a news article would be popular or not, as measured by the number of “shares” it received on the online platform mashable.com. The authors extracted 47 metadata features containing data types such as integer, ratio, Boolean, and nominal. Using more than 35,000 news articles as a training set, the authors were able to achieve 74.5% accuracy using GBM, which was a 1.8% improvement over using random forest on the same problem.[[7]](#footnote-7)

In this paper the authors are conducting relatively advanced feature engineering, using natural language processing techniques to generate a latent Dirichlet allocation closeness score for each of the topic categories based on the text of the article. Additionally, the authors used sentiment analysis to engineer several features relating to the negativity or positivity of the news article. This would be an interesting area to explore as part of future research, as these features made up a significant portion of the top 20 features in terms of importance score in their model.

According to Uddin et al. “We defined the popularity of an article based on a decision threshold D. For instance, when an article is shared more than 1400 times e.g. D ≥ 1400 , we labeled the article as Popular ; otherwise, we labeled the article as Unpopular…”[[8]](#footnote-8) Based on an analysis on the data used by the researchers for this paper, 1400 shares represents the median number of shares in their dataset.

> news<-read.csv("OnlineNewsPopularity.csv")  
> summary(news$shares)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Min | 1st Qu. | Median | Mean | 3rd Qu | Max |
| 1 | 946 | 1,400 | 3,395 | 2,800 | 843,300 |

Conducting further analysis of the numbers above, we can conclude that the authors determined a dataset to be popular at the 47th percentile by number of shares of a news article.

> news\_list<-as.list(quantile(news$shares, probs = seq(0, 1, by= 0.005)))  
> output<-as.tibble(news\_list[90:110])  
> output %>% as.data.frame() %>% write.table(file="newspercents.csv", sep=",")

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 46.50% | 47.00% | 47.50% | 48.00% | 48.50% | 49.00% | 49.50% | 50.00% | 50.50% | 51.00% |
| 1,300 | 1,400 | 1,400 | 1,400 | 1,400 | 1,400 | 1,400 | 1,400 | 1,400 | 1,500 |

As this applies to the current project, while a 47th percentile threshold may have been appropriate for the authors’ research, for this paper a higher percentile threshold will be used. For the use case supported by my current research we aim to highlight a fewer number of datasets likely to be more popular, at a higher percentile of the measured indicator variable denoting popularity. Based on the work by the authors of this paper, a binary classification seems to be an appropriate model that to use for the purposes of this project.

# Dataset

In order to predict the popularity of a new dataset based on its metadata, my project will rely on two source datasets. The first dataset is the metadata catalogue from Open.Canada.ca. The Government of Canada publishes an open dataset of the metadata for all the data and information resources available on Open.Canada.ca. This dataset is updated every night with the latest data.[[9]](#footnote-9) The second dataset to be used is a listing of the number of downloads from Open.Canada.ca for the last 12 months, by dataset. This dataset is published as an .xls workbook and is updated on the 1st business day of each month with data from the previous 12 months.

These datasets are licenced by the Government of Canada to the user under the Open Government Licence. This licence allows a use to ““Copy, modify, publish, translate, adapt, distribute or otherwise use the Information in any medium, mode or format for any lawful purpose.”[[10]](#footnote-10) The Open Government Licence is an attribution licence with similar terms and conditions to the Creative Commons Attribution Licence.

## Metadata Catalogue

This dataset is a JSON lines dataset, where each line of the file is a JSON object representing the metadata of an individual dataset on Open.Canada.ca. As of 2 Feb 2021, there were 87,331 metadata records contained within the metadata catalogue.

In addition to the metadata catalogue, the Government of Canada published a metadata application profile as a downloadable document on Open.Canada.ca. This document functions much as a data dictionary would for a dataset, but at the level of the application for Open.Canada.ca. According to this metadata application profile, there are 92 different metadata elements that can potentially be populated for anyone dataset. As a dataset record can consist of multiple resources, or files packaged within the same dataset record with optional and mandatory metadata for each of the resources, as the number of resources grows, the metadata can grow into a JSON object of significant size.

For example, if one examines a single record from the JSON lines, line 3333 for example, one can observe that there are 66 metadata elements populated for this record. In addition to the 66 elements, one can observe that several of the elements are populated with one or more lists of additional elements of varying lengths.

In addition to the metadata elements that are populated with a controlled vocabulary, there are several metadata elements that contain free text. Due to Government of Canada Official Languages policy, these elements need to be present in both English and French. For this project only the English metadata will be used.

> str(fromJSON(lines[[3333]]))

List of 66

$ aggregate\_identifier : chr ""

$ association\_type : list()

$ audience : list()

$ author : NULL

$ author\_email : NULL

$ collection : chr "fgp"

$ contact\_information : chr "{\"fr\": {\"pays\": \"Canada\", \"electronic\_mail\_address\": \"agri-geomatics-agrog@agr.gc.ca\"}, \"en\": {\"co"| \_\_truncated\_\_

$ contributor : Named list()

$ creator\_user\_id : chr "aa584ab4-544c-4c5c-81da-d1cff9bd96fa"

$ data\_series\_issue\_identification: Named list()

$ data\_series\_name : Named list()

$ date\_published : chr "2016-10-18 00:00:00"

$ display\_flags : list()

$ distributor : chr "{\"fr\": {\"pays\": \"Canada\", \"nom\_organization\": \"Gouvernement du Canada; Agriculture et Agroalimentaire "| \_\_truncated\_\_

$ file\_id : chr "09c983d4-6081-4896-9e29-5cda0125804f"

$ frequency : chr "as\_needed"

$ geographic\_region : list()

$ groups : list()

$ hierarchy\_level : chr "dataset; jeuDonnÃ©es"

$ id : chr "09c983d4-6081-4896-9e29-5cda0125804f"

$ imso\_approval : chr "true"

$ isopen : logi FALSE

$ jurisdiction : chr "federal"

$ keywords :List of 2

..$ en: chr [1:7] "Farmlands" "Floods" "Agriculture" "Crops" ...

..$ fr: chr [1:7] "Terre agricole" "Inondation" "Agriculture" "Cultures" ...

$ license\_id : chr "ca-ogl-lgo"

$ license\_title : chr "Open Government Licence - Canada"

$ license\_url : chr "http://open.canada.ca/en/open-government-licence-canada"

$ maintainer : NULL

$ maintainer\_email : chr "agri-geomatics-agrog@agr.gc.ca"

$ metadata\_contact :List of 2

..$ en: chr "Government of Canada; Agriculture and Agri-Food Canada,agri-geomatics-agrog@agr.gc.ca"

..$ fr: chr "Gouvernement du Canada; Agriculture et Agroalimentaire Canada,agri-geomatics-agrog@agr.gc.ca"

$ metadata\_created : chr "2017-01-19T16:50:26.024672"

$ metadata\_modified : chr "2020-12-09T19:55:00.618827"

$ name : chr "09c983d4-6081-4896-9e29-5cda0125804f"

$ notes : chr "Agriculture and Agri-Food Canada has created a model using the integral equation model (IEM) to process radar ("| \_\_truncated\_\_

$ notes\_translated :List of 2

..$ en: chr "Agriculture and Agri-Food Canada has created a model using the integral equation model (IEM) to process radar ("| \_\_truncated\_\_

..$ fr: chr "Agriculture et Agroalimentaire Canada a crÃ©Ã© un modÃ¨le Ã lâ\200\231aide du modÃ¨le dâ\200\231Ã©quation intÃ"| \_\_truncated\_\_

$ num\_resources : int 3

$ num\_tags : int 0

$ org\_section : Named list()

$ org\_title\_at\_publication :List of 2

..$ en: chr "Agriculture and Agri-Food Canada"

..$ fr: chr "Agriculture et Agroalimentaire Canada"

$ organization :List of 11

..$ approval\_status: chr "approved"

..$ created : chr "2016-09-23T17:24:47.909925"

..$ description : chr ""

..$ id : chr "2ABCCA59-6C57-4886-99E7-85EC6C719218"

..$ image\_url : chr ""

..$ is\_organization: logi TRUE

..$ name : chr "aafc-aac"

..$ revision\_id : chr "a7147b11-bb05-4011-b18b-429c990b6286"

..$ state : chr "active"

..$ title : chr "Agriculture and Agri-Food Canada | Agriculture et Agroalimentaire Canada"

..$ type : chr "organization"

$ owner\_org : chr "2ABCCA59-6C57-4886-99E7-85EC6C719218"

$ place\_of\_publication : list()

$ position\_name : Named list()

$ private : logi FALSE

$ program\_page\_url : Named list()

$ ready\_to\_publish : chr "true"

$ reference\_system\_information : chr "EPSG:3857,http://www.epsg-registry.org/,8.3.4"

$ relationships\_as\_object : list()

$ relationships\_as\_subject : list()

$ resources :'data.frame': 3 obs. of 22 variables:

..$ cache\_last\_updated: logi [1:3] NA NA NA

..$ cache\_url : logi [1:3] NA NA NA

..$ created : chr [1:3] "2020-12-09T19:55:00.738242" "2020-12-09T19:55:00.738255" "2020-12-09T19:55:00.738259"

..$ data\_quality :List of 3

.. ..$ : list()

.. ..$ : list()

.. ..$ : list()

..$ datastore\_active : logi [1:3] FALSE FALSE FALSE

..$ description : chr [1:3] "" "" ""

..$ format : chr [1:3] "GeoTIF" "PDF" "PDF"

..$ hash : chr [1:3] "" "" ""

..$ id : chr [1:3] "1659f2c8-9ace-46fe-8527-abbd0913a005" "6c427c24-1cc0-4cd9-a420-ee1a51fdc1ea" "73fc3000-04c9-480d-ac5f-948c415c634b"

..$ language :List of 3

.. ..$ : chr "zxx"

.. ..$ : chr "en"

.. ..$ : chr "fr"

..$ last\_modified : logi [1:3] NA NA NA

..$ mimetype : logi [1:3] NA NA NA

..$ mimetype\_inner : logi [1:3] NA NA NA

..$ name : chr [1:3] "Pre-packaged GeoTIF files (No linguistic component)" "Data Product Specification (English)" "Data Product Specification (French)"

..$ name\_translated :'data.frame': 3 obs. of 2 variables:

.. ..$ en: chr [1:3] "Pre-packaged GeoTIF files (No linguistic component)" "Data Product Specification (English)" "Data Product Specification (French)"

.. ..$ fr: chr [1:3] "Fichiers GeoTIF prÃ©emballÃ©s (aucun Ã©lÃ©ment linguistique)" "SpÃ©cifications du produit (Anglais)" "SpÃ©cifications du produit (FranÃ§ais)"

..$ package\_id : chr [1:3] "09c983d4-6081-4896-9e29-5cda0125804f" "09c983d4-6081-4896-9e29-5cda0125804f" "09c983d4-6081-4896-9e29-5cda0125804f"

..$ position : int [1:3] 0 1 2

..$ resource\_type : chr [1:3] "dataset" "guide" "guide"

..$ revision\_id : chr [1:3] "a587ad0a-8d99-490d-97cc-6aa002a2eb22" "a587ad0a-8d99-490d-97cc-6aa002a2eb22" "a587ad0a-8d99-490d-97cc-6aa002a2eb22"

..$ state : chr [1:3] "active" "active" "active"

..$ url : chr [1:3] "https://www.agr.gc.ca/atlas/data\_donnees/geo/radarsatSurfaceSoilMoisture/" "https://www.agr.gc.ca/atlas/supportdocument\_documentdesupport/radarsatSurfaceSoilMoisture/en/ISO\_19131\_RADARSAT"| \_\_truncated\_\_ "https://www.agr.gc.ca/atlas/supportdocument\_documentdesupport/radarsatSurfaceSoilMoisture/fr/Cartographie\_de\_hu"| \_\_truncated\_\_

..$ url\_type : logi [1:3] NA NA NA

$ responsible\_role : chr "RI\_414"

$ restrictions : chr "unrestricted"

$ revision\_id : chr "2c3a2b56-df55-4381-9e4f-3ce8c47d465b"

$ spatial : chr "{\"type\": \"Polygon\", \"coordinates\": [[[-110, 45.3], [-75, 45.3], [-75, 51], [-110, 51], [-110, 45.3]]]}"

$ spatial\_representation\_type : chr "grid"

$ state : chr "active"

$ status : chr "completed"

$ subject : chr [1:3] "form\_descriptors" "nature\_and\_environment" "science\_and\_technology"

$ tags : list()

$ time\_period\_coverage\_start : chr "2015-04-01 00:00:00"

$ title : chr "RADARSAT-2 Surface Soil Moisture"

$ title\_translated :List of 2

..$ en: chr "RADARSAT-2 Surface Soil Moisture"

..$ fr: chr "HumiditÃ© de surface du sol avec RADARSAT-2"

$ topic\_category : chr "geoscientific\_information"

$ type : chr "dataset"

$ url : NULL

$ version : NULL

One might hypothesize that either date published or date last modified might have a significant relationship with the number of downloads for a dataset. From a 1st principles perspective, date fields will be excluded from use as predictor variables. As the aim of the project is to produce a reliable prediction of how popular a dataset will be from the metadata, at the time of publication. Including dates into the model would be introducing information into the model will not be useful for this use case, as all datasets being evaluated will have a very similar time elapsed since publication when they are evaluated by the production model.

Elements used for our predictive model was organization name, update frequency, keywords, number of resources, collection, subject, and other features were added or modified depending on our initial results and further refinements.

Based on the fact that certain datasets did not exist for the entire duration of the 12 months of downloads data collection, scaling the number of downloads proportionally with the amount of time they existed during the data collection period was used to maximize the amount of records the model is based on.

## Open Government Analytics - Downloads per organization, last 12 months

The second dataset that used in this project is a listing of downloads per dataset over the last 12 months. This dataset is an MS Excel workbook that is generated on the 1st business day of each month, containing the number of downloads for each dataset. Where a dataset contained in the metadata is missing from the downloads dataset it is presumed that no downloads occurred for that given dataset for the previous 12 months.

This dataset is not packaged in a convenient way for analysis as this excel workbook contains 88 tabs. There is a summary tab that give a roll up of the total number of downloads per government department, then a tab that lists the downloads for each of the 86 Government of Canada departments or agencies whose data received downloads from Open.Canada.ca. As, such it required significant data manipulation to transform the data into a format that can be combined with the metadata.

If we examine the downloads data, one can see that the ID, title, and number of downloads is present for each record in the dataset.

> head(dl\_df)

# A tibble: 6 x 3

`ID / Identificateur` `Title English / Titre en angl~ `Number of downloads / ~

<chr> <chr> <dbl>

1 ba2645d5-4458-414d-b1~ Annual Crop Inventory 3820

2 292646cd-619f-4200-af~ Canadian Drought Monitor 1695

3 0c113e2c-e20e-4b64-be~ Canada Land Inventory (CLI) 1478

4 9e1efe92-e5a3-4f70-b3~ Land Use 2010 1434

5 abf04733-8225-4d3c-83~ Canada Land Inventory (CLI) 1:~ 812

6 ade80d26-61f5-439e-89~ Terrestrial Ecoregions of Cana~ 631

>

If we look at the summary of the downloads we can easily observe that the dataset is heavily right skewed, with the mean being six times greater than the median. This suggests that a small number of downloads get a disproportionately high number of downloads.

> summary(sum\_dls)

ID / Identificateur Title English / Titre en anglais Number of downloads / Nombre de téléchargements

Length:11004 Length:11004 Min. : 4.00

Class :character Class :character 1st Qu.: 4.00

Mode :character Mode :character Median : 12.00

Mean : 72.42

3rd Qu.: 28.00

Max. :25509.00

>

From a research perspective I was hopeful that the model we developed was able to identify which metadata elements can be used to identify this smally number of highly popular datasets, then predict that newly published datasets sharing those factors will also perform well in terms of number of downloads.

Additionally, we can observe from the data summary that there are only 11,004 datasets contained within the downloads, while there are 87,731 contained within the metadata catalogue. One can therefore deduce that approximately 87% of datasets received no downloads, further contributing to the extreme right skewedness of the dataset.

# Approach

A visual depiction of my project shows the following steps.

## Step 1: Loading the Data

In this step we are fetch two files from Open.Canada.ca.  
1. The Open Data Portal Metadata Catalogue from <https://open.canada.ca/data/en/dataset/c4c5c7f1-bfa6-4ff6-b4a0-c164cb2060f7>

2. The Downloads per organization, last 12 months file from <https://open.canada.ca/data/en/dataset/2916fad5-ebcc-4c86-b0f3-4f619b29f412>

We use the Gunzip function from the R::utils package to un-Gzip and untar the metadata catalogue. As an alternate step one could make the API call, <https://open.canada.ca/data/api/action/package_list> save that output, then iterate over the API call <https://open.canada.ca/data/api/action/package_show?id=> {ID} for each ID that was returned in the package\_list call. Fetching the Gziped file is a significant time savings.

## Step 2: Data Preparation and Feature Extraction

In this step we are going to convert our two raw datasets into one dataset suitable for further analysis.

In the first part of this step we are taking our downloads excel note book and removing the summary tabs at the front which we do not require using the readxl package. We are then mapping each of the remaining sheets into a data frame using the map\_dfr function from the purrr package. In this step we purposefully retained the “title English” field from the excel sheets. Since there are summary rows on each sheet with no title populated, once we have all sheets into the data frame we can use the na.omit function to remove the summary rows from our dataframe.

In the second part of this step we are taking our JSON lines metadata and selecting relevant features that seem worthwhile for exploring as input to our predictive model. We are first initializing a data frame with the relevant metadata elements as columns, then we are looping over the remaining lines in the dataset to populate our data frame.

In the third part of this step we are going to merge our two newly created data frames. As both dataframes have the ID for the dataset, we can preform a left join of the downloads data onto the metadata dataframe.

In the fourth part of this step we are going to convert our textual category data into numeric factor data.

The lastly in this step we are going to use encode the data into a one hot encoding. In one hot encoding we create new columns in the dataframe for each of the values of the factor data, then binary encode the factors as either true or false.

My code for this step is available at: <https://github.com/PatLittle/Ryerson-Big-Data-Analytics-Final-Project/blob/master/data_load_and_prep.R>

## Step 3: Exploratory Data Analysis

In this step we will understand our data in order to determine how to prepare and normalize our data. As well as to determine which data features are worth including in our model, and potentially discovering new features that could be engineered.

This step may be an iterative approach, where we move up and down the staircase, either adding or pruning from our main dataset, or down to the model building phase

In our initial data exploration, we have examined the structure of our data by looking at the number of records in each of the two source datasets, and realized there is a significant left skew in the dataset.

## Step 4: Model Building

Based on the literature review conducted, I selected XGBoost in a binary classification mode as the model to use for this paper.

The first step of this phase is to create a copy of our prepared dataframe, then remove the features that will not be used as predictor or target variables in the model. At this step we are removing the ID, date created, and date last modified from the initial data. We are also removing unused features that we engineered such as the number of days it was present during the downloads measurement, number of days since last modified, and the time adjusted downloads. We then convert our factor variables into an numeric encoding.

The second step is to split the data into the test and training set. To accomplish this step we will use the initial\_split function from the rsample package within the tidymodels framework. We will accept the default proportion offered by the function of 25% test, 75% training. In this stage we will populate the stratification parameter with the target variable. This will ensure that testing and training splits retain a proportional count of popular and non-popular datasets within the splits.

The third step is to define the model specification. Using the boost\_tree interface for boosted trees within the parsnip library we define the specification. This boost\_tree specification defines the hyperparameters for the model. For each hyperparameter, the specification requires that we provide either a value or indicate that it will be optimized using tuning at a later stage. For our project we will specify the number of trees as a static value, then tune the following hyperparameters: tree depth, minimum number of data points, number of randomly sampled predictors at each split, loss reduction, sample size, and learn rate.

The fourth step is the define the search space for the hyperparameters that will be tuned. Using the dials package within the tidymodels framework we can specify the parameter space. By default a regular grid is used for hyperparameter tuning, where every combination of values for each parameter is included within the search space. This is a computationally expensive experimental design, as no optimization is preformed in defining the search space, it is a brute force approach to discover the optimal combination of hyperparameters.[[11]](#footnote-11) For our implementation we will define a latin hypercube search space, that will use a space filling design to reduce the number of values in the search space, such that the search space does not grow exponentially. In the dials package we specify the size parameter as the total number of parameter combinations that should be returned, therefore implementing a latin hyper cube ensures we get an improved diversity of possible parameters over a basic grid search.

The fifth step is setting up a tidymodels workflow. In this step we indicate what formula to use, and what model specification to use. For this paper, we are using the formula of binary popularity, predicted by all other variables in the dataset, and we are using the model specification as defined a previous step.

The sixth step is to define our cross validation. For this paper we will use the vfold\_cv function within the rsample package. For this function we define the number of cross validations we want to preform, and we also specify our stratification variables, which we want the function to represent proportionally within each of our samples of the data.

The seventh step is the train our model and tune our hyperparameters on our training data. In this step we will take advantage of parallel processing to dramatically improve our training time. To leverage parallel processing we will use the doParallel library and specify the number of CPU cores we want to devote to model training. In my research I found that devoting 50% of the available CPU cores gave reliable training performance, while devoting 75% of CPU cores often caused the system be overwhelmed and resulted in the model training failing. As some of the model training times for the various models explored as part of the research were in excess of 12 hours, having reliable model training performance was essential to completing the research within the allotted timeframe.

In order to train the model and tune the hyperparameters, we use the tune\_grid function with the tune package. At this stage we supply the function with the workflow, resamples, and search grid defined in previous steps.

## Step 5: Model Evaluation

Our last step was to evaluate the predictions of our model against our test set, and hopefully confirm that we have achieved the aim of the project, to develop a classifier to analyze the metadata of a newly published dataset on Open.Canada.ca and predict a popularity category that will correspond to the popularity of the dataset as determined by the number of downloads.

The first step in the model evaluation is to collect the model evaluation metrics from the results object generated during the training and tuning process.

As the second step in the model evaluation stage, we visualize how the hyperparameter values tried as part of the tuning process influenced the model performance. We can visualize the results in order to gain an understanding of what hyperparameter values resulted in better model performance.

The third step is the collect the best combination of parameters by using the select\_best function from the tune package. In this step we specify that we want the best result as measured by the area under the receiver operator curve.

The fourth step is to finalize our workflow, by defining that we want to use the hyperparameters from our best performing combination, and apply that to the workflow defined in previous steps.

As fifth step we examine the variable importance of each of the variables contained in our training data and gain an insight into which of the variables had the best predictive value.

As a sixth step we take our finalized model and fit it to our test data. We then collect the performance metrics from the resulting results object. This will provide us with the model accuracy and area under the receiver operator curve metrics.

As a seventh step we can visualize the results and examine the confusion matrix. During this step we want to examine the area under the receiver operator curve, as our primary model evaluation metric, but we also want to examine the confusion matrix to determine if the model is making proportional type 1 and type 2 errors. Depending on desired outcome of the analysis, one might be willing to accept a higher proportion of each of the error types. Our error consequences can be summarized as below.

|  |  |  |
| --- | --- | --- |
| Prediction / Truth | False – Not Popular download | True – Popular download |
| Predicted not popular | Correct prediction | Here datasets that are popular are not identified by the model as being popular, therefore the model will not highlight these to the relevant stakeholders at the time of creation. |
| Predicted popular | Here datasets that are not popular are identified by the model as being popular, therefore the model will be feeding bad predictions to the stakeholders. | Correct prediction |

For our research problem, we are willing to accept significant error of both type 1 and type 2 types, as the stakes are relatively low, the aim is to surface relevant content to promote, however we want to maximize the reliability of our predictions. Additionally for our use case, we may be demand limited by the stakeholders in terms of the number of predictions that they are able to action by promoting these datasets to their audience. We can adjust the predictions by changing the popularity percentile we set as the threshold, and retrain the model based on the demand for model output predictions.

# Results

Using the steps outlined in the section above, we have trained an evaluated a model that should preform acceptably for our use case.

The code for this step is available at <https://github.com/PatLittle/Ryerson-Big-Data-Analytics-Final-Project/blob/master/final-report/final-rmd.Rmd>

## Model Building

Based on initial results, it was determined that we would narrow the scope of the final model to discard datasets from the “geogratis” collection as this collection consisted of a large number of records that received little usage. After narrowing the scope we are left with 18,316 observations in our dataset. For our data preparation we set a 95th percentile in terms of the number of downloads to consider that a popular download. This requires that a dataset receive atleast 18 downloads per year to be considered popular. While this is a relatively low number, due to the skewedness of the data it was determined that this is an acceptable threshold for our research problem.

As described within the model building section previously we split the data into the test and training sets, accepting the default proportion of 25% test, 75% training. We also set the downloads as our stratification variable to ensure proportional representation of the popular downloads in each of the splits.

We then set our model specification as a boost\_tree, and set the number of trees parameter to 2000 trees. We also set our latin hyper cube with size 50. We setup our tidymodels workflow to predict popularity based on all other variables and added the model specification above to the workflow. We set our cross validation at 10 folds, and specified our stratification as the popularity variable.

While training our model and tuning our hyperparameters we specified the number of CPU to devote to training as 4, as this model was trained on an 8 core system. This resulted in an approximately 3 hour training and tuning time.

## Model Evaluation

As defined within the step above, the area under the receiver operator curve was used as the primary evaluation metric.

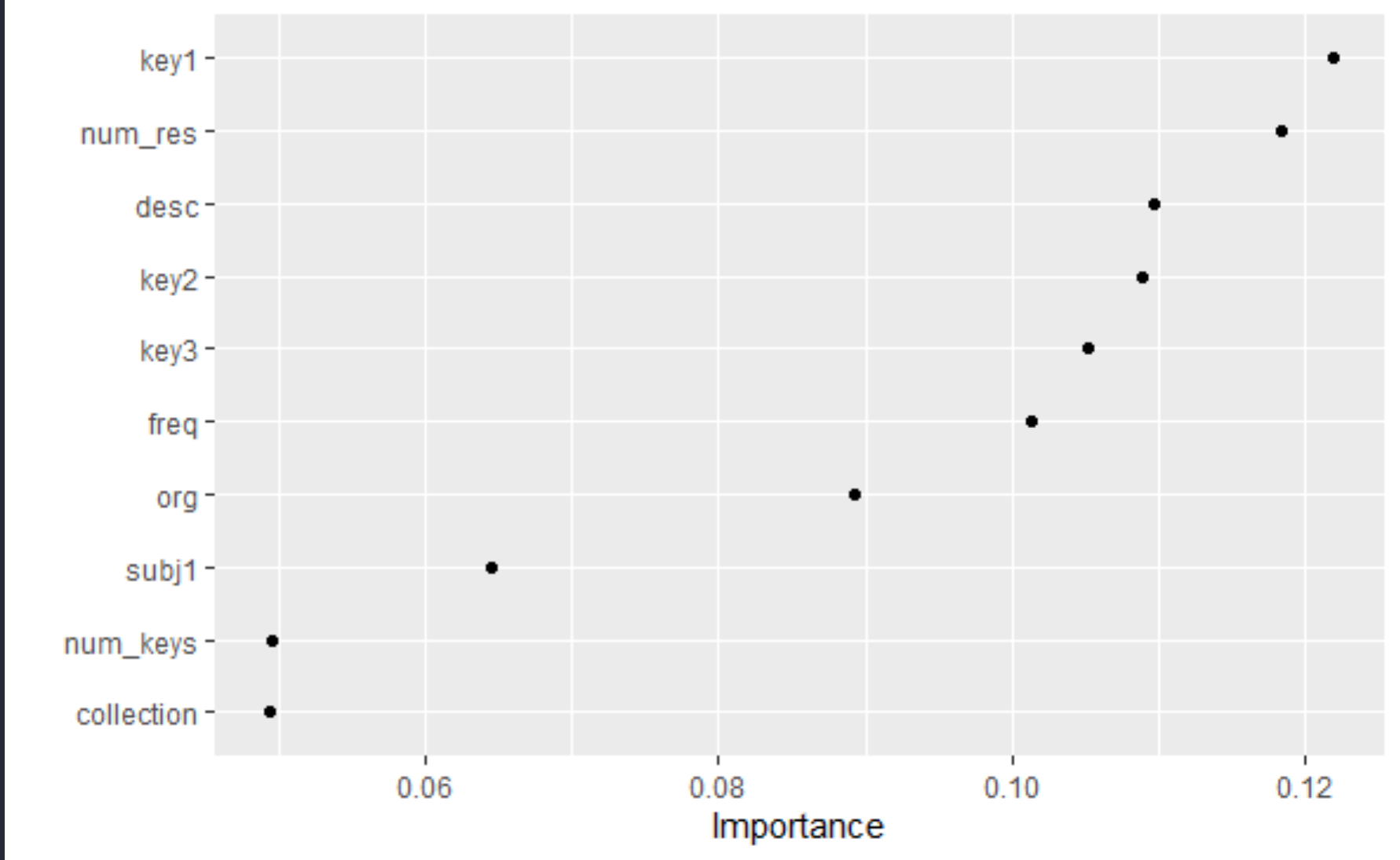
With a model as defined above, we were able to achieve a model ROC\_AUC of 84.42% and an accuracy of 82.38%.

With this tuned model we achieved the hyperparameter values of: mtry: 1, min\_n: 17, tree-depth: 8, learn\_rate 0.007581093, loss\_reduction: 2.644965e-09, and sample\_size: 0.619886.

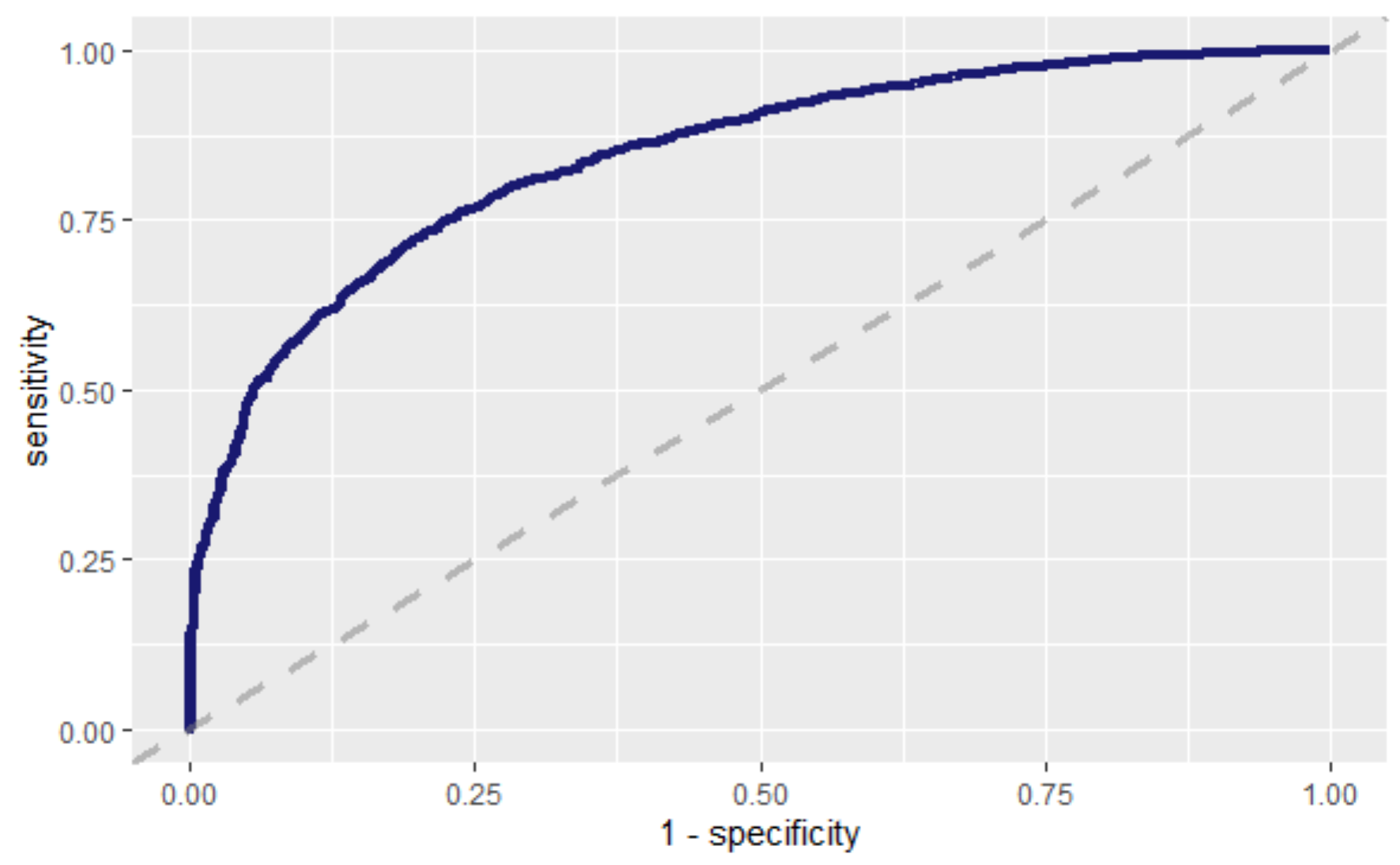
With the model we were able to achieve a confusion matrix as follows:

|  |  |  |
| --- | --- | --- |
| Truth / prediction | Predicted – not popular | Predicted - popular |
| True not popular | 3,427 | 621 |
| True popular | 174 | 357 |

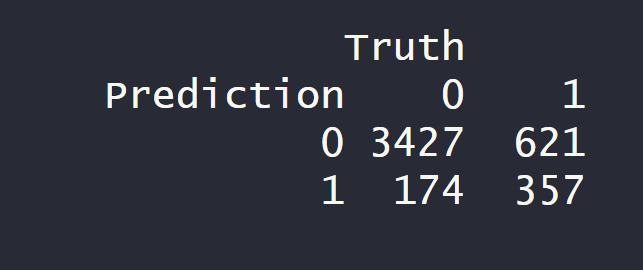
Within this model we were able to determine that the factor value of the 1st key word was the most important variable used by our model. Other important variables included the number of files contained within the dataset, the number of words used in the description, as well as other variables. A visualization of the importance of the variables is as follows:



With our model we were able to achieve a sensitivity/1-specificity plot as follows:



With a confusion matrix of:



We conclude that the number of type two errors is still relatively high, even with a model accuracy of over 82%. Based on this result one can assume that certain datasets with the potential to be popular remain undiscovered, while a small proportion of non-popular datasets may be surfaced to stakeholders as potentially popular.

Based on benchmarking to studies within out literature review, the paper we examined also using the XGBoost model was able to achieve a model accuracy of 90.39%. Within this research we were able to approach our colleagues’ results achieving a lower model accuracy of 82.38%. While this is a significantly lower measure of model accuracy, based on the highly skewed source data, this seems to result in a useful model that can serve the intended purpose.

# Further Research

In order to capitalize on the model developed as part of this research an initial attempt to put this model into production was made.

Exporting the model trained during this research as an .Rds object, it can be easily uploaded to any online file store. We can then consume the model in the .Rds format with low compute costs.

Using the CKAN API native to Open.Canada.ca, one can return a list of newly added datasets by using the /action/package\_search command with a parameter to sort by recently added datasets. We can then extract the newest dataset added from the JSON object returned. Using the code developed to do the feature extraction to generate the initial dataset for the research we can fit the new datasets into a format that is consumable by our model.

We can then simply use the Predict function within R and feed in our trained model object, and our new data as arguments to receive a prediction of popularity. This code can be easily packaged into a standalone .R script that can run automatically.

Using Github actions we can dynamically provision a Ubuntu virtual machine and run our production model at a scheduled frequency. We can control the frequency and other parameters by providing a yaml specification file that defines this workflow. We can also save our prediction outputs to a file to reference at our convenience.

The model in production has been running hourly thus far and outputting predictions to a .CSV uploaded on Github. The code for this portion is available at: <https://github.com/PatLittle/Ryerson-Big-Data-Analytics-Final-Project/blob/master/final-report/model-in-production.R> and the workflow definition for Github actions is available at <https://github.com/PatLittle/Ryerson-Big-Data-Analytics-Final-Project/tree/master/.github/workflows>

While the data pipeline required for this step has successfully been implemented, modifications to our model need to occur in order to instruct the model how to reliably deal with new factor levels from the data that the model has not been trained on. This required modification of the model will be required before the production model will be suitable to feed popularity predictions to our stakeholders.

# Conclusions

Overall, we were able to achieve a useful model that can predict with an acceptable level of accuracy, datasets that are likely to be popular downloads from Open.Canada.ca. While our final model developed as part of this research achieved a lower measure or AUC and accuracy compared with similar research examined as part of our literature review, this model still remains useful for predicting popularity of open datasets.

Compared to the best direct comparison of the research by Šlibar, we significantly outperform a similar research problem of predicting the popularity of open datasets.

# References

“Open Government Analytics.” Open Government, Government of Canada, 1 February 2021, <http://open.canada.ca/en/content/open-government-analytics>

2 N. Darapaneni et al., "Movie Success Prediction Using ML," 2020 11th IEEE Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON), New York, NY, USA, 2020, pp. 0869-0874, doi: 10.1109/UEMCON51285.2020.9298145.

3 World Wide Web Foundation. "Table 1: Open Data Barometer scores for Open Data Charter adopters and G20 members (minus EU) – Champions, Contenders and Stragglers groups on green, yellow or red background respectively." September 2018. *https://opendatabarometer.org/.* 12 Feb 2021. <https://opendatabarometer.org/leadersedition/report/#table1>

4 Šlibar, Barbara . “Predicting the Number of Downloads of Open Datasets by Naïve Bayes Classifier.” TEM Journal, 2019. https://doi.org/10.18421/TEM84-33.

5 “API Guide¶.” API Guide - CKAN 2.9.1 Documentation, The CKAN Association, 2021, <https://docs.ckan.org/en/2.9/api/>.

6 M. T. Uddin, M. J. A. Patwary, T. Ahsan and M. S. Alam, "Predicting the popularity of online news from content metadata," 2016 International Conference on Innovations in Science, Engineering and Technology (ICISET), Dhaka, 2016, pp. 1-5, doi: 10.1109/ICISET.2016.7856498.

7 <https://open.canada.ca/data/en/dataset/c4c5c7f1-bfa6-4ff6-b4a0-c164cb2060f7>

8 “Open Government Licence - Canada.” Open Government, Government of Canada, 26 April 2013, http://open.canada.ca/en/open-government-licence-canada

9. Urban, Nathan & Fricker, Thomas. “A comparison of Latin hypercube and grid ensemble designs for the multivariate emulation of an Earth system model.” Computers & Geosciences. 36. 746-755. 10.1016/j.cageo.2009.11.004.

1. “Open Government Analytics.” Open Government, Government of Canada, 1 February 2021, http://open.canada.ca/en/content/open-government-analytics [↑](#footnote-ref-1)
2. N. Darapaneni et al., "Movie Success Prediction Using ML," 2020 11th IEEE Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON), New York, NY, USA, 2020, pp. 0869-0874, doi: 10.1109/UEMCON51285.2020.9298145. [↑](#footnote-ref-2)
3. World Wide Web Foundation. "Table 1: Open Data Barometer scores for Open Data Charter adopters and G20 members (minus EU) – Champions, Contenders and Stragglers groups on green, yellow or red background respectively." September 2018. *https://opendatabarometer.org/.* 12 Feb 2021. https://opendatabarometer.org/leadersedition/report/#table1 [↑](#footnote-ref-3)
4. Šlibar, Barbara . “Predicting the Number of Downloads of Open Datasets by Naïve Bayes Classifier.” TEM Journal, 2019. https://doi.org/10.18421/TEM84-33. [↑](#footnote-ref-4)
5. “API Guide¶.” API Guide - CKAN 2.9.1 Documentation, The CKAN Association, 2021, <https://docs.ckan.org/en/2.9/api/>. [↑](#footnote-ref-5)
6. M. T. Uddin, M. J. A. Patwary, T. Ahsan and M. S. Alam, "Predicting the popularity of online news from content metadata," 2016 International Conference on Innovations in Science, Engineering and Technology (ICISET), Dhaka, 2016, pp. 1-5, doi: 10.1109/ICISET.2016.7856498. [↑](#footnote-ref-6)
7. Ibid. [↑](#footnote-ref-7)
8. Ibid. [↑](#footnote-ref-8)
9. https://open.canada.ca/data/en/dataset/c4c5c7f1-bfa6-4ff6-b4a0-c164cb2060f7 [↑](#footnote-ref-9)
10. “Open Government Licence - Canada.” Open Government, Government of Canada, 26 April 2013, http://open.canada.ca/en/open-government-licence-canada [↑](#footnote-ref-10)
11. Urban, Nathan & Fricker, Thomas. “A comparison of Latin hypercube and grid ensemble designs for the multivariate emulation of an Earth system model.” Computers & Geosciences. 36. 746-755. 10.1016/j.cageo.2009.11.004. [↑](#footnote-ref-11)