|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| |  | | --- | |  | |  | |  | |  | |
| Population & Search  Understanding the relationship between population demographic trends, and the trends of our digital behavior. |
| Jake Schroeder  Springboard Intermediate Data Science  2018 |

Table of Contents

[I. Understanding the Problem 1](#_Toc512073938)

[II. Identifying the Client 2](#_Toc512073943)

[III. Describing the Data 3](#_Toc512073945)

[IV. Other Data Sources 4](#_Toc512073947)

[V. Initial Findings 5](#_Toc512073949)

[V. Machine Learning Application 6](#_Toc512073951)

[VI. Final Thoughts 9](#_Toc512073953)

## I. Understanding the Problem

## How can we leverage our digital footprint to better understand the demographics that drive search behavior?

### This project aims to illustrate the power of demographic information when aiming to make sense of online search engine behavior.

### Put into machine learning terms - which features (segments of the demographic spectrum) are suitable predictors for a given keyword? Does the selection of features from the model line-up with our intuitions? For example, if our intuitions tell us that the older demographic will be more interested in online banking (due to factors such as retirement, or mobility concerns), we would assume the model to reflect such intuitions in its selected features.

### This project will yield a product that uses the most granular demographic data available to fit models to the keyword of the user’s choice. With this functionality, any party that needs to leverage their target keywords strategically will have another layer of insight in their toolbelt. As will be discussed later, the parties this tool speaks to are largely content marketers and digital strategists.

## II. Identifying the Client

## Content marketing can take years to develop. Content marketers and digital marketing strategists could really use a tool that connects demographics to the high-value keywords they wish to target.

Both content marketers and digital marketing strategists are the target client for the project. These parties thrive on high-values keywords that they believe will drive value for their business. Without demographic insights related to these keywords, there is no way to know if the keywords they target will drive the right type of customer to their website.

Our target client relies on personas to drive conversations around digital content creation and optimization. For example, younger personas may tend to be more impulsive, use mobile devices, and prefer short, quick-hitting content that uses lists or other mechanisms to keep the content well-organized and easy to read. On the contrary, older personas may prefer using a desktop (or laptop) computer, doing extended research, and only committing when the timing absolutely calls for it. Each persona requires its own approach in digital marketing terms.

When developing content for the older, research-oriented persona, the target client cannot afford to miss on topic selection. Keywords may show high volumes or excellent conversion rates, but that does not warrant how they should be used. Placing keywords that the younger persona cares about in a resource article would be a big mistake, given the descriptions above. This project aims to mitigate those mistakes by providing demographic-infused insight for any keyword.

If the target client uses this product accordingly, they will be able to better craft their content to the personas who are most likely to engage with it.

## III. Describing the Data

## Retrieving the demographic and search data for both initial statistical analysis and the machine learning application.

Data wrangling came in two formats for this project. The first wrangling was for the initial statistical analysis – lining up a lighter demographic dataset with national Google Trends data. The second wrangling was for the machine learning application – parsing ASCII format county-by-county demographic data and lining it up with state-by-state Google Trends data.

1. **Data Wrangling for Statistical Analysis:**

The Google Trends data was retrieved using the unofficial Google Trends python library, pytrends. The data came in biweekly increments and was aggregated to a yearly form so that it could be compared alongside the population data. The population data was sourced from the census bureau in the form of population forecasts, since the output was yearly and considered the population demographics both past and anticipated.

Using pandas, the data was wrangled in such a manner than all unnecessary columns were dropped (notably, race, ethnicity and gender). With the data aggregated yearly, each row (year) from 2004 to 2018 shared both population and Google Trends data for the selected keyword(s). In this format, making statistical inference on the high-level relationships between these datasets was straightforward.

1. **Data Wrangling for Machine Learning**

The pytrends library was used again, but this time in a custom function that was iterated for each state. At the end, all states were concatenated together, and a pivot was used to setup a multilevel index, where the results were grouped by year and state and ordered accordingly.

The demographic data came in line-delimited ASCII format – so the creation of a custom parse was in order. Using the conventions of character placement, the parser creates a dictionary, which is then turned into a pandas DataFrame. After grouping the data by year and state, the parser aggregates by year and pivot to match the setup of the Google Trends data. The result is a robust .csv file ready for machine learning.

## IV. Other Data Sources

## The current datasets may not meet the needs of statistically rigorous machine learning and forecasting. Therefore, an alternative dataset is provided, below.

The National Cancer Institute, in cooperation with the US Census Bureau, grants access to the annually released population estimates at the county level. At the time of this writing, the estimates span the years 1990-2016. This data is available in 19 age groups (5-year spans) or by single year age groups.

Below are the links to the downloads and documentation for parsing the population data:

<https://seer.cancer.gov/popdata/download.html>

<https://seer.cancer.gov/popdata/popdic.html>

## V. Initial Findings

## The strong trends shown by the population data are mirrored in statistically sound Pearson Correlation Coefficients. The results urge for further investigation and discovery.

Through a two-sample permutation t-test, the analysis produced high confidence in the repeatability of strong Pearson Correlation Coefficients. Thus, we rejected the null hypothesis that the absolute value of the Pearson Correlation Coefficient would be less than .25.

The permutation test involved randomly sampling pairs of data, such that:

(x = Population of Age Group, y = Popularity of Keyword)

This was accomplished through random sampling of the index that the two datasets share (2004-2018). This entire process, including the production of a Pearson Correlation Coefficient, was repeated 10,000 times per keyword, per age group. A total of 20,000,000 iterations and calculations were made during this inferential exercise.

Each permutation test yielded a mean Pearson Correlation Coefficient and a p-value. The p-value was calculated based on the number of permutation test replicates with an absolute value correlation coefficient less than .25, divided by the total number of permutation test replicates (10,000). For example, if a selected keyword / demographic combination had 1,000 instances where the relationship (positive or negative) was weaker than .25, the p-value would by 0.10 and the null hypothesis would be accepted.

This analysis showed many age groups to have strong correlations with very low p-values. This indicates that strong correlations between the datasets were highly repeatable, granting increased confidence in the underlying relationship. The confidence this test invoked in the relationship between the two datasets warranted further analysis via machine learning methodologies

## V. Machine Learning Application

## Finding the right classifier for the job, testing its performance, then tuning the best performer before it all comes together.

After the data wrangling put the data in-place for regression analysis, it was time to test every sensible model until a winner could be determined. Since sci-kit learn offers an efficient pipeline setup, where data can be preprocessed and trained all at once, this project leaned wholly on sci-kit’s catalogue of models.

Each model was tested using its R-squared and mean squared error as performance metrics.

**Regression Test Results**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Model | **Ridge** | Lasso | Elastic Net | Lasso Lars |
| R2 | **0.745863** | 0.638055 | -0.36284 | -6.1527 |
| MSE | **22.97876** | 32.72665 | 123.2259 | 646.738 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Model | **Orthogonal Matching Pursuit** | **Bayesian Ridge** | **Stochastic Gradient Descent** | Passive Aggressive |
| R2 | **0.789698** | **0.747389** | **0.75918** | -0.32612 |
| MSE | **19.0152** | **22.84076** | **21.77466** | 119.9059 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Model | Theil Sen Regressor | KNeighbors Regressor | **Linear SVR** | SVR | Random Forest |
| R2 | -1.74819 | -0.11051 | **0.812741** | 0.384621 | 0.35391 |
| MSE | 248.4878 | 100.4113 | **16.93174** | 55.64179 | 58.41862 |

As illustrated in the table above, many models performed reasonably well. All well-performing models were from the linear library in sci-kit learn; SVR, Random Forest and KNeighbors all performed poorly on the dataset.

The intricacies of each model was taken into consideration; for example, the PassiveAggressive model gives the option to shuffle the data, which is not acceptable for a time-series regression. When tuning the parameters of the top-performing models, none could compare to Linear SVR. Tuning proved to be a game of hundredths in terms of R-squared, a scale of improvement too small to allow competing models to catch Linear SVR.

As an added measure to ensure data integrity, each model was tested against the dataset with the population figures in logarithmic form. This was a scaling measure recommended by my mentor, Andrew. The results across the board suffered; all models scored a negative R-squared. With StandardScalar in the sck-kit learn pipeline, this preprocessing step could be ignored with confidence.

**Tuning Linear SVR**

The following LinearSVR hyperparameters were tunable for the dataset:

* **C :** float, optional (default=1.0)

*Penalty parameter C of the error term. The penalty is a squared l2 penalty. The bigger this parameter, the less regularization is used.*

* **loss :** string, ‘epsilon\_insensitive’ or ‘squared\_epsilon\_insensitive’ (default=’epsilon\_insensitive’)

*Specifies the loss function. ‘l1’ is the epsilon-insensitive loss (standard SVR) while ‘l2’ is the squared epsilon-insensitive loss.*

* **epsilon :** float, optional (default=0.1)

*Epsilon parameter in the epsilon-insensitive loss function. Note that the value of this parameter depends on the scale of the target variable y. If unsure, set epsilon=0.*

* **fit\_intercept :** boolean, optional (default=True)

*Whether to calculate the intercept for this model. If set to false, no intercept will be used in calculations (i.e. data is expected to be already centered).*

Using GridSearchCV, and passing the pipeline as the model, the project experimented with tuning each of these hyperparameters. Epsilon was the only hyperparameter excluded from tuning in the final product, since its default setting consistently yielded better results than any tuned setting. The reason for this is that LinearSVR sets the epsilon value incredibly low (near 0) by default, and the parameters for tuning (even when starting at 0.0001) were too large to make any positive impact on the model.

In test runs, both C and the loss function operate best at their default values (1.0 and “epsilon\_insensitive”). This is perhaps due to the naturally good fit between the dataset and model; that is, the dataset in many ways meets the ideal for what LinearSVR was intended to do. Another interesting note is that the intercept is consistently fit. This makes sense since the data is not perfectly centered, and having the model fit the intercept makes the most sense.

**Final optimized scores:**

|  |  |
| --- | --- |
| Model | LinearSVR |
| R2 | 0.8149 |
| MSE | 16.7295 |

**Communicating Results in Real Terms**

The model communicates findings in real terms by retrieving the coefficients and finding the three most positively scored features and the three most negatively scored features. Using a dictionary of age ranges and their feature index (being mindful to exclude the previous year feature index), the model communicates these features as easy-to-understand age ranges. For example, “bank account” yields the following age groups as the most positively associated:

65-69, 60-64, 85+

And the three most negatively associated:

40-44, 75-79, 20-24

This allows the user to intuitively understand the demographics they should and should not consider as targets for the keyword “bank account.” While the adjacent features 75-79 and 85+ represent a misnomer of sorts, the model communicates that young and middle age people are not searching for “bank account” online like the older 60-something demographic is.

This aligns with our intuitions about which people would search for this term; older demographics may look for good deals to place their retirement savings, and younger demographics may not care to research or overcomplicate their choice in bank account.

## VI. Final Thoughts

## Thoughts on why SVR was the best performer, and the results of the test experiment. Discussing future considerations to improve the model’s performance.

What makes Support Vector Regression special is its ability to group features / data points together during the creation of its model. When dealing with 5-year age groups, this is highly useful, since the characteristics of adjacent age groups are likely to overlap. Therefore, this dynamic feature weighting lends itself to a more effective model than any of the more traditional regression methods.

Since the project is linear in nature – demographics as x and keyword popularity as y – it is little surprise that the non-linear SVR and RandomForestRegressor did not perform as well. It may have been looking for dimensions that the dataset simply did not have.

The results of the test experiment were very exciting. The test keyword “bank account” returning positively-scored demographics of 60-year-olds and the very-old (85+) line up with intuitions about online banking previously mentioned. It is important to note that the very-old demographic may have received an advantage in selection since its population numbers are so inherently low.

An attempt was made to take phenomena like this into consideration. Recall that during the test phase, the model was tested with using the log of the demographic information. This yielded a far-less accurate model. Also recall the use of StandardScaler in the pipeline.

With these measures in mind, such a result from the 85+ age group asserts that factoring in bias toward low-population demographics may be a very complicated enterprise. A simpler remedy would be to remove the demographic altogether. Since the 85+ demographic is a very valuable demographic, leaving it out of the model was not a valid alternative for this project.

As it is often said, “All models are wrong, but some models are useful.” This model has proven to be useful, but room for improvement still exists. Instead of just taking age demographic information, we could factor in gender, race and ethnicity as well. Adding these extra features runs the risk of muddying the model, but also holds the promise for more granular demographic insight. The potential for running regression on a more local basis (the state level, for example) is also intriguing, especially since the focus on hyperlocal content in the digital marketing space is more prevalent than ever.