Predicting If An Ice Cream Shop Will Be Successful In A Given Zip Code

Author
Yumel Hernandez

Advisor
Vishnu Venkatesh

Abstract:

The purpose of this project is to apply machine learning to a real-world problem. Specifically, an attempt to predict the success of an ice cream shop in a specific zip code area. The success of ice cream shop majority dependent on patronage and financial support from customers around their geographic location. The prediction of the success of the ice cream shop in the area was based on financial and other quantitative data in the publicly available IRS Statistics of Information (SOI) dataset which contains individual tax returns statistics aggregated to zip codes. This approach is helpful for those who desire to open up a small ice cream shop but are unsure as to the location of the business.

1 Introduction

To this day, most small business owners wanting to open up an ice cream shop do so by accessing the location easiness to travel to, the availability of sufficient parking space, and if there is a good amount of traffic in that area during the spring and summer. While this approach is important, it relies on qualitative subjective information. This project takes a quantitative approach to identify if an additional ice cream shop will be successful in a given zip code. Currently, there are multiple approaches to help solve this problem including the use of geolocation data. The approach we'll take is a novel approach of using publicly available financial information released by the IRS.

By attempting to classify the number of ice cream shops in a particular zip code, entrepreneurs could determine if the market is underserved or overserved. If the market is underserved, the likelihood of a successful ice cream store is high and thus a potentially profitable business could be built. This tool attempts to minimize risks to the entrepreneur or business manager and allow them to identify zip codes where there is a higher chance of success before any investment is made.

2 Data collection

The primary dataset is a publicly available IRS Statistics of Information (SOI) dataset which contains individual tax returns statistics aggregated to zip codes. For privacy reasons, individuals are not identified in this dataset. This dataset contains 153 features and 166,537 records. 152 of the features have a float data type meaning the vast majority of the dataset is a numerical number. The only feature that is not a float type, but an object type is the state feature which contains the name of the state the zip code is part of. The dataset occupies 200 MB of space. This dataset is important for small businesses as it provides important financial sustainability data about the geographic areas where the business will be located.

The secondary dataset was obtained through Yelp's API. We used the Yelp's API to gather the amount of ice cream shops for every zip code contained in the primary IRS dataset. The dataset occupies 247 KB of space. The number of ice cream shops in a particular zip code is the target variable in this project. Machine learning models were used to predict the number of ice cream shops using the features of a zip code given by the IRS SOI dataset.

3 Data Cleaning

The python library Pandas was used for data cleaning. To improve the readability of the IRS SOI dataset, each column was renamed, lowercased, and stripped of extra white spaces. In addition, the state feature is a categorical variable thus was converted into a dummy variable.

The dataset was reduced to occupy 106 MB from 200 MB of space by optimizing the data frame's memory footprint with the downcasting of all the features that were of type float. All quantitative values are stored as either a float64 (8 bytes) which is excessive as some numbers don't need to occupy much space and could be stored in a float16 saving a lot of space. under the hood, pandas represent numeric values as Numpy ndarrays, and stores it in a continuous block of memory. This storage model occupies less space and makes things faster by its ability to retrieve values quickly.

The IRS SOI dataset documentation guide mentioned that zip codes with less than 100 returns and those identified as a single building or nonresidential ZIP code were categorized as "other" with the zip code of 99999. This means that the record 99999 contained information from different zip codes across the US making the row useless thus was removed. Features that proved to have very little predictive power were removed such as "state" that abbreviate the state in which the zip code is located, and "statefips" which serve as an internal code the IRS uses to classify tax returns. This left the shape of the dataset to be 165,925 rows with 151 columns.

It's important to highlight that although the IRS SOI dataset contains 166,537 only 27,660 are unique zip codes. This means that each unique zip code is reproduced 5 or 6 times in the dataset for it to equal 166,537. The reason is that each unique zip code is broken down into 5 or 6 levels of income brackets since every zip code contains residents in different socioeconomic levels.

Various functions were used to loop through the 27,660 unique zip codes, insert each unique zip code in Yelp's API, collect the number of ice cream in each unique zip code, and automatically exports it into a CSV after the data collection. After the collection, each unique zip code contains the number of ice cream shops available in that specific zip code.

An inner merge between the cleaned original IRS SOI dataset and the newly collected dataset from Yelp was done. The inner merge resolves discrepancies between the two datasets in terms of one being 165,925 rows and the other 27,660 rows. The new combined dataset is then 165,925 records with 153 features.

The zip code feature was then removed as the models were overfitting this feature. This makes sense as we want the model to predict the number of ice cream shops based on the features a given zip code contains, therefore, the zip code number itself has very little value.

4 Data Exploration

In order to learn about the data, the Seaborn and Matplotlib libraries in python were used to create graphs. These graphs represent the distribution of each randomly selected features in the dataset since it's inefficient to properly analyze 153 graphs and derive meaning.

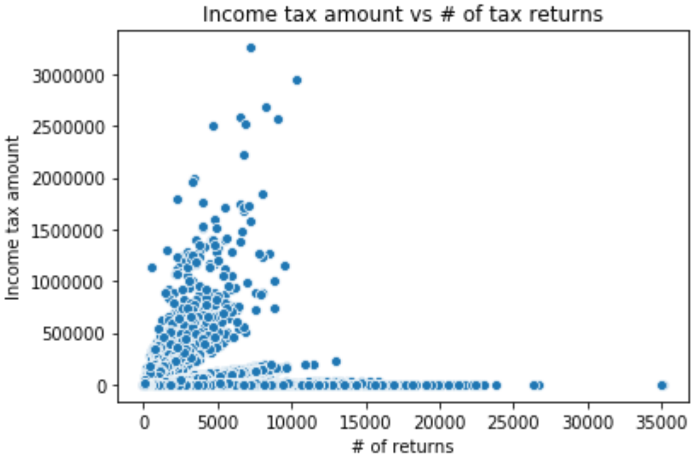


Figure 1: Scatterplot of Number of returns against Income Tax Amount

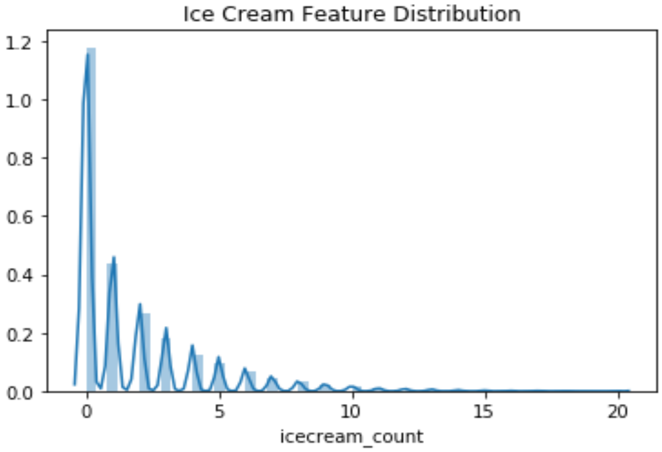


Figure 2: The distribution of the Target Variable Ice cream count

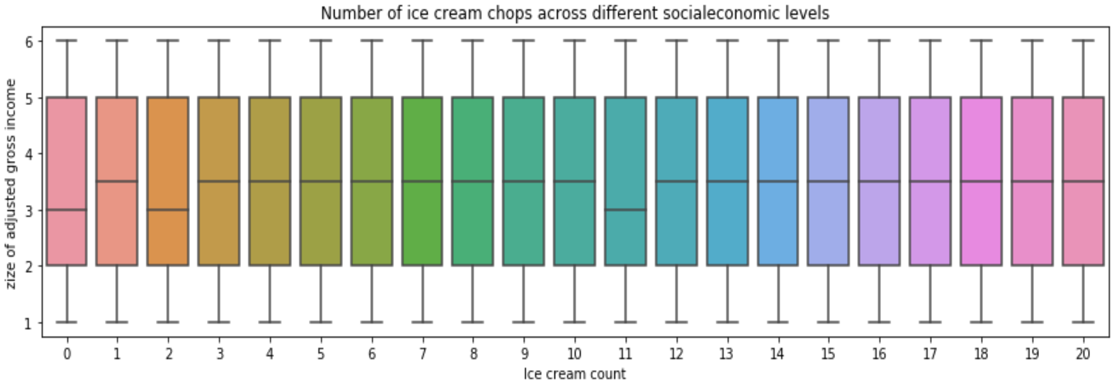


Exhibit C: Boxplot of the number of ice cream shops across different various economic levels

5 Feature Selection

Due to the size of the dataset, which contains153 features, dimensionality reduction techniques must be used. It must be resized to be smaller for not only speed and storage purposes but to also remove multi-collinearity which removes highly correlated value and improves the interpretation of the parameters for machine learning models. A lot of features make it hard to understand the relationship between each feature. Additionally, having so many features increases the likelihood of overfitting machine models to the data.

Variance threshold was the first method that attempted. Variance threshold was used to select the most important features out of 153 features. This method removes features below a certain cutoff, the standard 0.5 cutoff was used. The rationale behind this method is that when a feature does not change much within itself, it generally has low predictive power thus could be removed. This method does not consider the relationship of features with the target variable. This method was unsuccessful as none of the features were removed.

Principal component analysis (PCA) was the second method used for dimensionality reduction. This method does not eliminate features but condenses information. Each vector is created in a way and order to hold the same information of 151 features into just a few vectors. Additionally, PCA ensures each feature is independent of one another. To apply PCA, the data was standardized since some features would overshadow others due to their bigger size and scale.

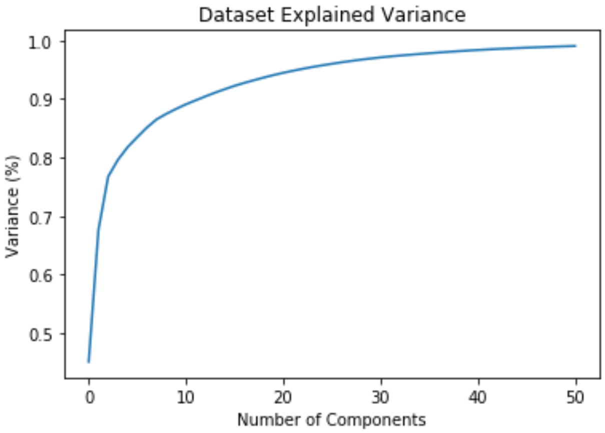


Figure 1: The he explained variation by each component

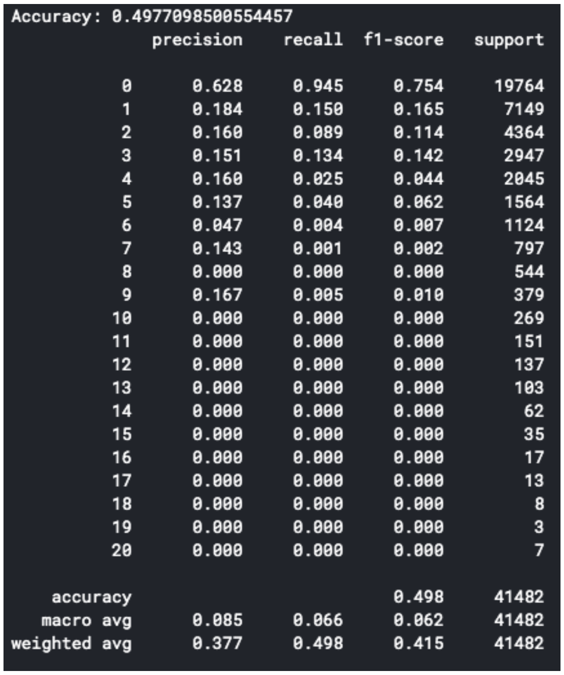
A PCA value of 20 was chosen since it explains 94% of the variation of the dataset. As soon in the graph above, after this point the benefit for each component increases at a decreasing rate.

6 Machine Learning Models

This is a supervised classification problem as we are using a lot of features from the IRS SOI dataset to predict the number of ice cream shops in a zip code. There are 20 categories the model can choose as a zip code can have up to 20 ice cream shops. Models were compared by cross-validation but also splitting the data in such that 75% of the data was allocated to training and 25% of the data was allocated to validation. The models used 20 principal component features. The Scikit-learn python library was used for the below machine learning models, except neural networks in which TensorFlow and Keras were used.

6.1 Multinomial Logistic Regression

Logistic regression finds relationships between the independent and dependent variables and concludes with a set of optimal coefficients for each variable. Default parameters were used including the maximum number of iterations being 100 and the regularization strength is 1. The first parameter changed was the solver being set to “saga” as it handles multinomial loss efficiently with large datasets. Additionally, the “multiclass” parameter was changed to multinomial as the loss minimized is the multinomial loss fit across the entire probability distribution.



```
Accuracy: 0.4977098500554457
              precision    recall  f1-score   support

           0      0.628     0.945     0.754     19764
           1      0.184     0.150     0.165      7149
           2      0.160     0.089     0.114      4364
           3      0.151     0.134     0.142      2947
           4      0.160     0.025     0.044      2045
           5      0.137     0.040     0.062      1564
           6      0.047     0.004     0.007      1124
           7      0.143     0.001     0.002       797
           8      0.000     0.000     0.000       544
           9      0.167     0.005     0.010       379
          10      0.000     0.000     0.000       269
          11      0.000     0.000     0.000       151
          12      0.000     0.000     0.000       137
          13      0.000     0.000     0.000       103
          14      0.000     0.000     0.000        62
          15      0.000     0.000     0.000        35
          16      0.000     0.000     0.000        17
          17      0.000     0.000     0.000        13
          18      0.000     0.000     0.000         8
          19      0.000     0.000     0.000         3
          20      0.000     0.000     0.000         7

    accuracy                          0.498     41482
   macro avg      0.085     0.066     0.062     41482
weighted avg      0.377     0.498     0.415     41482
```

Figure 2: Logistic Regression performance summary

6.2 K-Nearest Neighbors (KNN)

KNN uses Euclidian distance to find the proximity of records and identifies the label of a new record based on the dominant class of its nearest records. The number of records required to make a decision is determined by us hence a loop was built to try different "k" or neighbor values. KNN is data-driven, not model-driven and makes no assumptions about the data. Default parameters were used for the model.

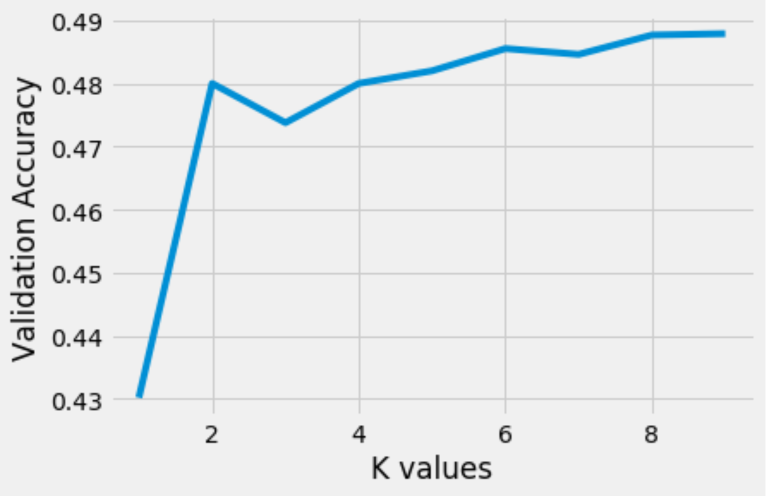


Figure 3: Accuracy across different K values for the KNN model

6.3 Random Forest

Random Forest allows us to build multiple large decision trees that operate together to make a predictive outcome. Each individual tree in the random forest spits out a class prediction and the class with the most votes becomes our model's prediction. Basically, the wisdom of crowds will allow for a powerful prediction. After parameter tuning to determine the best parameters to be used, the model was instantiated with 100 trees, and allowed bootstrap samples to be used for a more efficient model instead of using the whole dataset to build each tree.

Precision	0.091
Recall	0.079
F1 Score	0.078
Accuracy	0.495

Table 1: Results for Random Forest classification

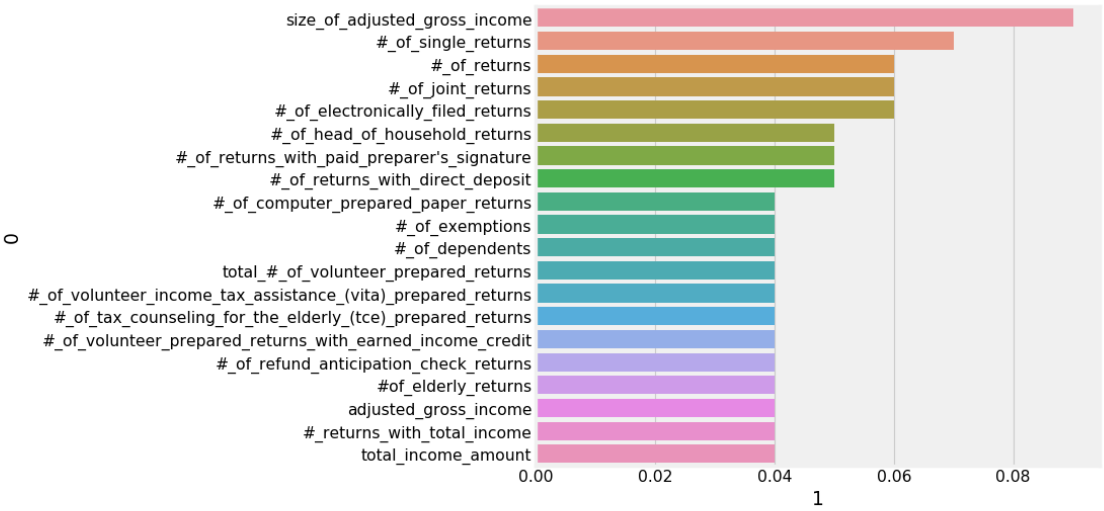


Figure 4: The 20 most important features according to Random Forest Classification

6.4 XGBoost

Gradient Boosting is an approach where new models are trained to predict the residuals (i.e errors) of prior models. XGBoost is a little more complex than other boosting models like AdaBoost as it comes along with more customizable parameters as a result. XGBoost It's also a faster algorithm than other boosting algorithms like AdaBoost. In complex and high-dimension problems XGBoost works better, given our complex dataset, it's no wonder it performs much better. It's important to highlight XGBoost is based on the idea of converting weak learners to a strong learner by updating their weights based on its residuals.

Parameter tuning optimization was used to find the algorithm optimal parameters including its learning rate, gamma, maximum tree depth, minimum child weight, and subsampling ratio of columns when constructing each tree.

During the parameter tuning process, I learned that grid-search can take a very long time. For computationally intensive tasks, grid search and random search can be incredibly time-consuming and generally less successful in finding the optimal parameters. These methods barely rely on previous information that the model learned during earlier optimizations. On the other hand, Bayesian Optimization is constantly learning from previous optimizations to find the ideal optimized parameters while requiring fewer samples to learn and come to a conclusion.

Precision	0.023
Recall	0.048
Accuracy	0.476

Table 2: The XGBoost model results

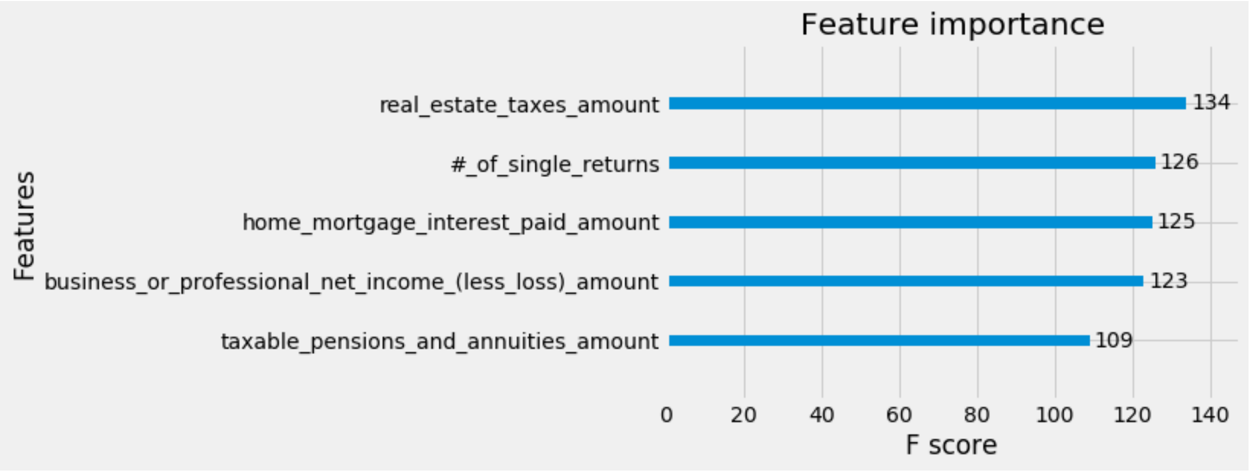


Figure 5: XGBoost model most important features

Our accuracy is slightly less than 50% but our precision and recall are low. A low recall tells us the model does not do a good job of predicting the true positives correctly. A low precision tells us there is a lot of false positives.

6.5 Neural Networks

Neural networks mimic the brain that is designed to recognize complex patterns. These algorithm uses different layers of neurons adjusted with different weights to derive at a conclusion. The framework, Keras, in TensorFlow was used to enable fast and easy deployment of the model. Four layers were used, the first being the shape of the features, and the proceeding layers size decreased by a half. In addition, the loss parameter was set to “categorical_crossentropy” as this is a multiclass classification problem, the output neuron was the shape of the categorical classes which in this case is 20.

loss	1.363
Validation loss	1.384
Log loss	1.368
accuracy:	0.510

Table 3: Neural Network model results

Accuracy is like a final exam with no partial credit. However, neural networks can predict the probability of each of the target classes. Neural networks will give high probabilities to predictions that are more likely. Log loss is an error metric that penalizes confidence in wrong answers. Lower log loss values are desired. In our case, it’s above the desired outcome of below

7 Next Steps

To improve on this process, it’s important to incorporate a way for the model to understand what a zip code means. A solution might be to use geospatial data to help the model understand the features from a geographic lens, helping it recognize richer patterns to predict the number of ice cream shops a zip code should have.

Furthermore, additional data such as costs, average revenues per ice cream stores across geographic regions, weather data, and publicly available traffic & population data could be used to enhance the prediction models. Additional data enhance models’ opportunities to find correlations and connections in the dataset thus increasing predictive power.

8 Concluding Remarks

Additional data needs to be used to improve the model’s performance. This project exercised some of the most sophisticated models in the ML field but yet it resulted in a poor predictive power. The models failed at predicting the number of ice cream shops a zip code should have. This means that additional high-quality data must be collected. As additional data is collected, different methods of data cleaning must be used to take into account these new features.

This was a big project with a lot of learning. From sophisticated dimensionality reduction techniques to new cutting-edge ML models to fast & efficient ways to deal with large datasets. This field is undergoing a lot of transformative changes to help solve the problem of site selection. It’s an ongoing problem, yet to be solved which will help thousands of future entrepreneurs form profitable businesses.

Dataset

"SOI Tax Stats - Individual Income Tax Statistics - 2017 ZIP Code Data (SOI)." *Internal Revenue Service*, www.irs.gov/statistics/soi-tax-stats-individual-income