Documentation

Darren Carvalho (301251637)

Professor: *Bilal Hasanzadah, David Parent*

Due date: August 18, 2023

Table of Contents

Executive Summary

0.1 Executive Introduction

In the competitive restaurant industry, accurate sales predictions are crucial for boosting profits, improving operations, and making customers happy. Traditional methods often miss important factors like time, weather, and economic changes that affect sales. This creates an opportunity to create a strong sales prediction system tailored just for restaurants. By using advanced data analysis and modern forecasting methods, we aim to help decision-makers make smarter choices.

Our project could lead to many positive changes. Better operations will come from smarter scheduling, managing supplies, and preparing food, which will save money and provide better service. Smart decisions will benefit from accurate sales analysis, helping with promotions, recognizing the brand, understanding trends, and pricing strategies. Optimized revenue will come from better pricing and resource management, increasing the money a business makes. Happier customers will result from accurate predictions of what they want, making their experience better.

0.2 Executive Objective

Our goal is to use machine learning to predict customer visits, optimizing inventory, staffing, and sales strategies for restaurants. We'll analyze factors like past sales, demographics, competitors, and social media to enhance decision-making. By employing supervised models, we aim to

predict visit numbers as continuous values or categories with predefined thresholds. This predictive tool will empower management, improve strategies, and reveal growth prospects. We assume historical sales matter, even if open data is scarce, and we'll explore alternative factors. We'll use the Yelp check-in count to estimate customer entries, comparing it with online orders, dine-in, and take-out. We assume reviews are unbiased.

Limitations include potential data gaps and the imperfect representation of actual customers with the Yelp check-in count. Generalizing findings to various restaurants might be complex.

For success, we'll assess classification models with precision and recall. Our approach uses supervised machine learning models, starting with regression-based models, and later categorizing for enhanced performance. We'll engage in data exploration, feature engineering, enrichment, external data integration, and feature selection. Model evaluation through cross-validation and the impact of scaled data will conclude our approach.

0.3 Executive Model Description

The project to predict how busy restaurants might be. By combining data and focusing on the important "flag_target," we built a dataset with 52 predictors and 1 main target. We had 42 categorical predictors (40 with two choices, 2 with more) and 12 numerical ones. After splitting the data into training and testing portions and tuning the models, we chose the best based on Accuracy, Precision, and Recall. This process led us to a solid predictive model for restaurant busyness.

Among the models, Random Forest performed the best with a 90% accuracy score. It found 'competitors_count' had a big impact and 'review_count' mattered a lot. Decision Tree reached an accuracy of 87.2%, highlighting the importance of 'review_count,' 'competitors_count,' and

'duration.' Logistic Regression, Support Vector Machine, and Naive Bayes also gave good results with accuracy around 89.79%, 89.78%, and 76.33% respectively. Random Forest's precision and recall scores, at 0.95 and 0.82 respectively, along with its 90% accuracy, made it the strongest choice. Our work offers solid insights into restaurant busyness prediction, with Random Forest as the top performer.

0.4. Executive Recommendations

Our recommended predictive models provide a comprehensive approach to enhance various aspects of restaurant operations and strategic decision-making. By utilizing these models, businesses can identify high-demand regions and optimal restaurant locations, considering factors like cuisine preferences and operating hours. Aligning promotional efforts with forecasted demand surges can strategically maximize their impact, driving customer turnout. The models also enable businesses to predict busy periods, optimizing staff scheduling to avoid overstaffing during slower times while ensuring resource efficiency. Moreover, accurate demand forecasting empowers effective inventory management, leading to reduced waste and operational costs. The implementation of just-in-time supply strategies based on demand predictions helps prevent stockouts, ensuring seamless operations.

After successfully implementing the models, businesses can strategically leverage their insights to drive actionable improvements and optimizations. This includes expanding strategically into regions of high demand potential, aligning promotions with forecasted demand surges, optimizing resource allocation through staff scheduling, enhancing inventory management efficiency, elevating customer experiences based on predicted influx, and consistently refining the models based on real-world performance data. Gathering feedback from stakeholders will

play a crucial role in fine-tuning model predictions and adapting strategies to evolving business needs. This iterative process ensures that the models remain robust, relevant, and beneficial for making informed decisions in the dynamic landscape of the restaurant industry.

1.0 Background

In the restaurant industry, precise sales forecasting holds immense value for boosting revenues, efficiency, and customer satisfaction. Conventional methods often miss the intricacies that impact sales, like time, weather, and economic factors. This presents an ideal chance to create a robust sales forecasting system exclusively for restaurants. By using advanced data analysis and forecasting methods, we can make smarter decisions.

System Design and Development:

Using insights from research, we plan to build an advanced sales forecasting system. This system will blend various forecasting methods, including multiple regression, Poisson regression, exponential smoothing, ARIMA, neural networks, Bayesian networks, and hybrid methods. We'll also use association rule mining to uncover hidden patterns in sales data.

Tailored Implementation:

We'll work closely with pilot restaurants to fine-tune the sales forecasting system. By gathering feedback from restaurant owners, managers, and stakeholders, we'll ensure the system meets their needs. This way, the system will offer practical insights for operations, planning, and revenue optimization.

Comprehensive Review:

A thorough review of forecasting methods used in the restaurant industry over the past two decades will serve as our foundation. This review will help us identify the most effective approaches for restaurant sales forecasting.

Expected Impact:

Our project anticipates a range of positive outcomes. Improved operations through optimized labor scheduling, inventory management, and product preparation will lead to cost savings and better service. Strategic decisions will gain insights from accurate sales analysis, aiding promotions, brand recognition, trends, and pricing strategies. Revenue optimization will lead to smarter pricing and resource allocation, increasing overall revenue. Enhanced customer satisfaction will stem from accurate demand prediction, which reduces waiting times and improves the dining experience.

1.1 Supporting Insights

Changing consumer preferences are driving the growth of healthier dining options and quick-service restaurants. Global food chains like Starbucks and McDonald's are expanding, targeting universities and colleges. Machine learning has potential in the Canadian food industry, especially in demand forecasting using past sales, weather data, and economic indicators to manage inventory and costs.

1.2 Project Gains

Our efforts promise significant gains. Precise demand prediction will guide marketing and pricing strategies, resulting in higher sales. Quality improvements will come from analyzing

customer reviews. Optimized resource allocation will reduce costs, while automation will speed up decision-making. These improvements will help restaurants thrive.

2.0 Problem Statement

The primary objective of this study is to employ machine learning techniques to accurately predict the proportion of customers visiting a restaurant. This prediction will be utilized to optimize inventory levels, allocate staff efficiently, and set sales targets. Key factors influencing customer influx include historical sales data, demographic information, the presence of competitors in the vicinity, and insights from social media. The study will use supervised models, aiming to predict the total number of customers in either a continuous or categorical format, facilitated by a threshold to determine customer significance. The model's outputs will empower management to make informed decisions, optimize business goals, and identify areas of improvement.

3.0 Objectives & Measurement

The objectives of this project include:

1. Identifying the impact of factors like reviews, review context, and ratings on customer flux.
2. Evaluating the influence of restaurant demographics such as location, operating hours, and cuisine type on its performance.
3. Understanding the key drivers that contribute to the success of certain restaurants over others.

4.0 Assumptions and Limitations

4.1 Assumptions

1. Historical sales data significantly impacts customer influx; however, open-source availability of such data is limited. Consequently, a comprehensive analysis considering other factors will be conducted.
2. The Yelp check-in count will represent the number of customers entering a restaurant. Efforts will be made to correlate this count with parameters like online delivery, dine-in, and take-out to estimate actual customer visits.
3. Reviews are assumed to be unbiased and reflect genuine customer opinions.

4.2 Limitations

1. Availability of open-source sales data might be limited, leading to reliance on alternative variables for analysis.
2. The Yelp check-in count might not be a perfect representation of actual customers, potentially impacting the accuracy of the model.
3. Generalizing results to all restaurants may be challenging due to variations in location, cuisine, and customer preferences.

4.3 Success Measures/Metrics

For regression models, Root Mean Square Error (RMSE) will be used to measure the accuracy of customer visit predictions. A lower RMSE indicates more precise forecasts. For classification models, precision will assess the accuracy of identifying significant customer visits, ensuring

efficient resource allocation. Recall will gauge the model's ability to capture all significant customer visits, minimizing missed opportunities.

4.4 Methodology and Approach

Supervised machine learning models will be employed, both regression and tree-based. Regression models will be initially tested, followed by categorizing the target variable to improve performance. The targets for classification can be binary or multiclass. Data exploration, feature engineering, data enrichment, external data incorporation, and feature selection will be integral steps. Model testing and evaluation will involve regression/classification models with cross-validation to assess performance. The effect of scaled data on model performance will also be evaluated.

5.0. Data Set Introduction

This dataset comprises a selection of Yelp's business, review, and user information. Initially compiled for the Yelp Dataset Challenge, this dataset offers students an opportunity to analyze Yelp's data and unveil insights. The dataset encompasses details about businesses in 8 metropolitan regions across the United States and Canada.

The restaurant dataset contains features about individual restaurants like location, types of cuisine offered, number of reviews received, operating hours, Wi-Fi, good for kids and other demographics. The dataset contains a total of 209,393 records and 14 features for restaurants in USA and Canada.

The reviews dataset contains details about the reviews received given to each restaurant over the years by different guests. Rating, useful, funny, cool are other parameters for each review. The dataset contains 432,479 records and 9 attributes.

The user dataset has information about each user, their activity on yelp like number of reviews, date since which they are using yelp, elite status that is provided to certain yelp users, friends, fans, and other compliment factors like hot, more, profile, cute, note, plain, cool, funny, writer, photos. The dataset contains 1,968,703 records and 22 features.

The target dataset for this use case is the check-in dataset which contains each restaurants unique id and the total number of check-ins between 2010 and 2019. There are 175,187 records for this dataset.

The primary and foreign keys for these datasets are business_id, user_id, review_id which would be used to merge the datasets post preprocessing and data cleaning for final modelling.

6.0. Exclusions

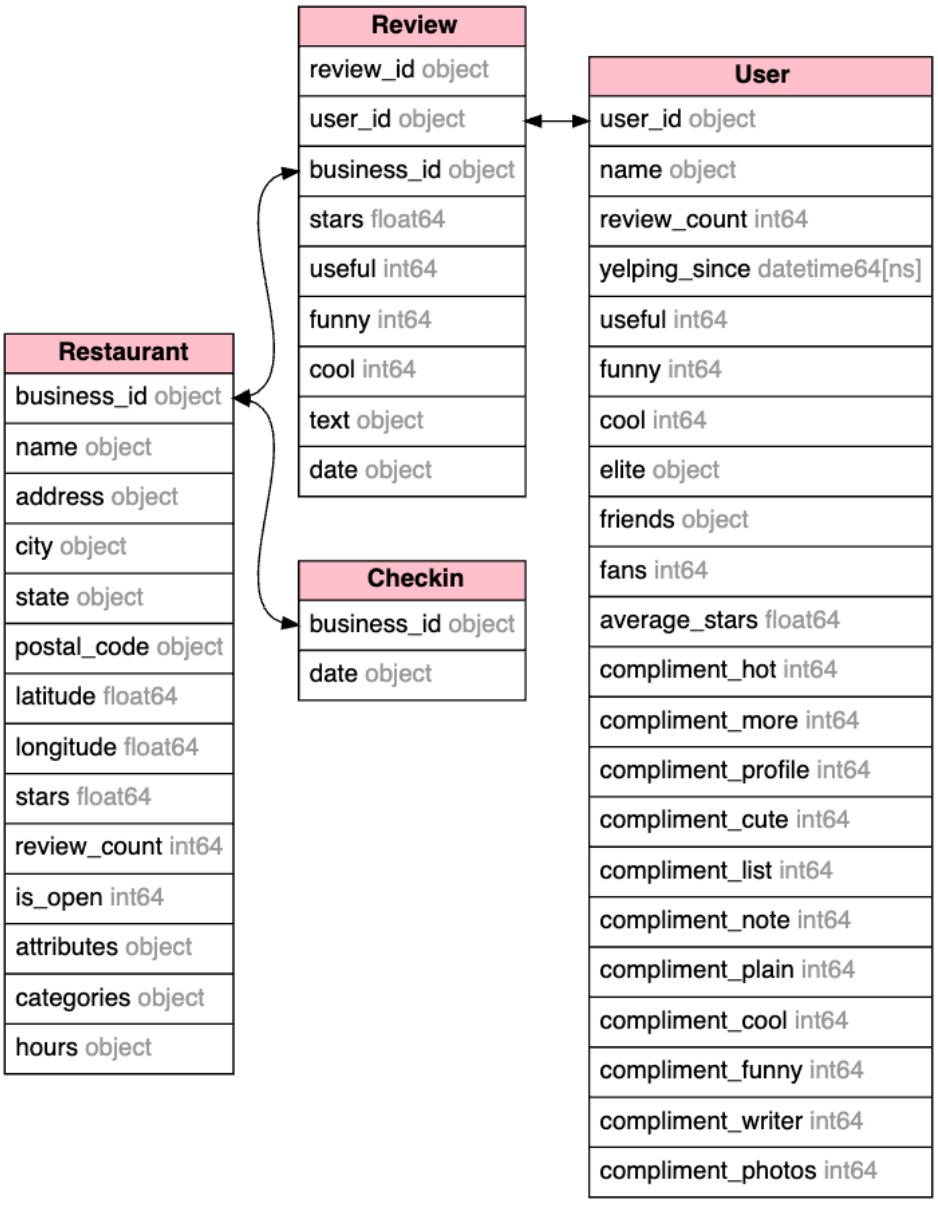
A tip dataset is also available for this challenge but was excluded from being implemented in modelling. This is because tip does not play a very significant role in determining whether a restaurant will be busy or not for a given day of week.

The primary keys and foreign keys from each dataset along with other unique factors like name, latitude, longitude, date will not be used in modelling, but these features will be used in feature engineering ton extract more information that will be useful in predicting how busy a restaurant will be.

6.1. Initial Data Cleansing or Preparation

The datasets are json files with respective key value pairs stored in a zip file. The datasets need to be read using json library in python and joined together after preprocessing and cleaning.

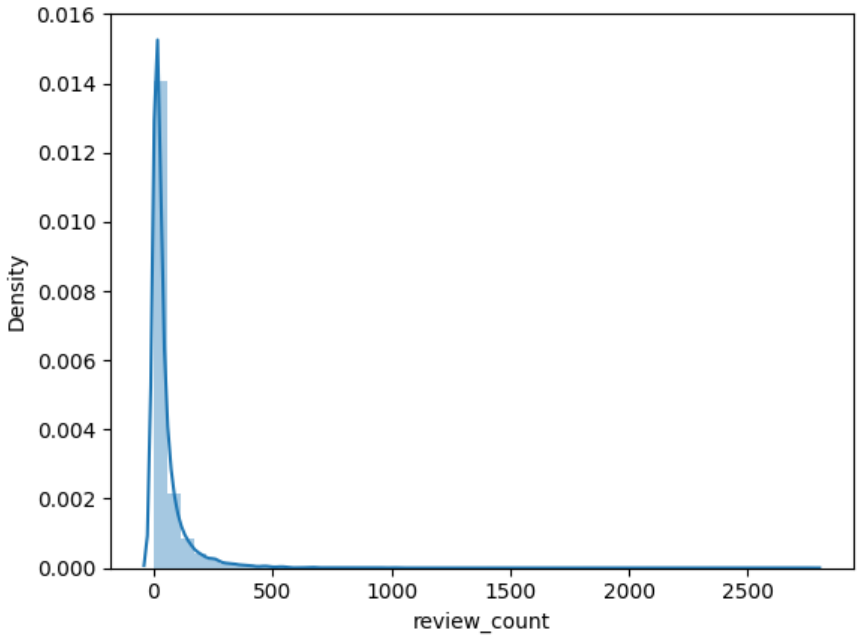
7.0. Data Dictionary

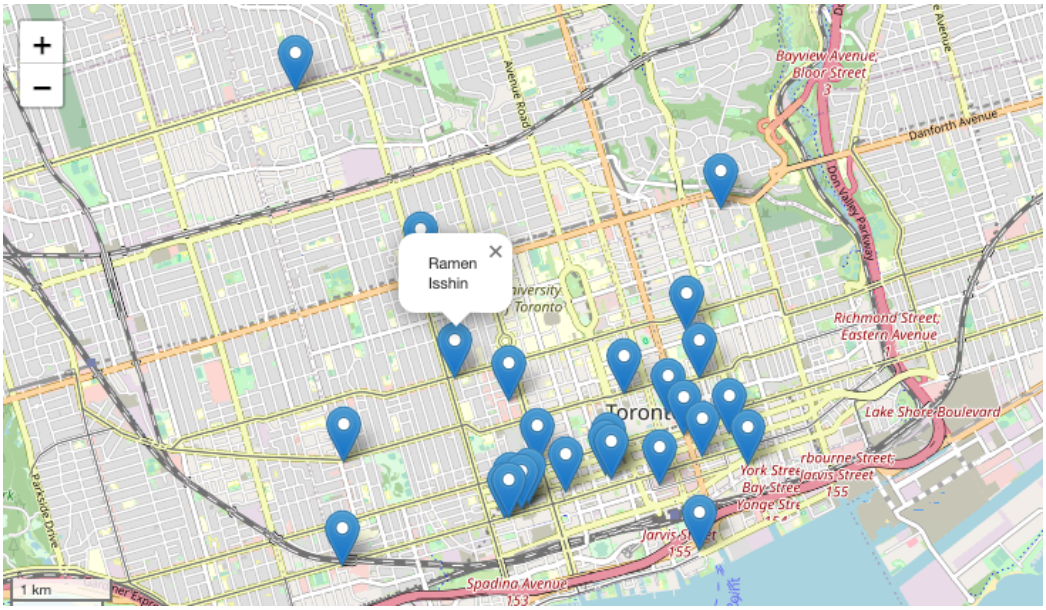


8.0. Data Exploration Techniques

Restaurants Data

Since the business goal is to create a model that can predict the flux of customers in Toronto, the dataset was filtered based on Toronto. We have decided to keep restaurants with ratings greater than 3 stars since the values for restaurants with lower ratings contained a lot of missing fields. Attributes and hours were two variables which contained a dictionary with multilevel hierarchy of values. This could be used to expand the key-value pairs into individual columns that can help in modelling. Categories were multiple for each restaurant that can be used to create more variables but since these inputs would create a very sparse matrix, and not impact the target significantly, manipulating this column will be considered on later stages. Review count is a highly positively skewed variable and appropriate techniques need to be taken to normalize it.





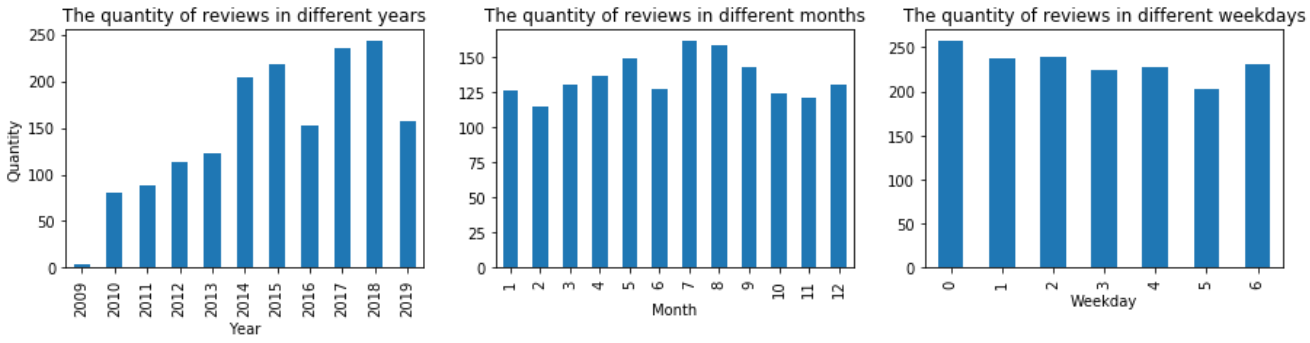
Restaurants that received highest number of reviews are near Yonge St and King St

Also the restaurants that received the highest number of reviews are near the university line

Subway - Line 1

Reviews Data

1. Number of reviews by Year, Month, Weekday



The review counts gradually increase through the years with drops in 2016 and 2019

respectively. There is no seasonality in the number of reviews received each month. But on

average, restaurants received 120-150 reviews each month. Reviews were usually given on Sundays.

2. Average Review rating over the years.

```
year
2009    3.250000
2010    3.975000
2011    4.193182
2012    4.175439
2013    4.065574
2014    4.098039
2015    4.174312
2016    4.117647
2017    4.131915
2018    4.279835
2019    4.006369
Name: stars, dtype: float64
```

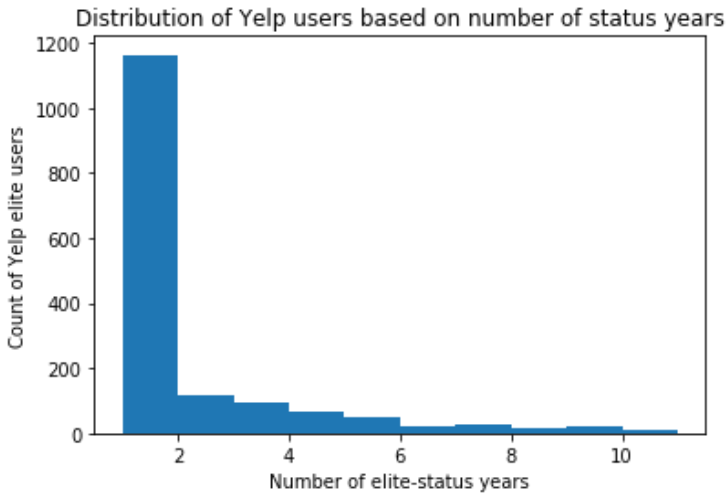
The review ratings have shown a relatively stable trend over the years, with a slight increase from 3.25 in 2009 to 4.28 in 2018, but then slightly dropped to 4.01 in 2019.

3. Reviews word cloud from elite users

The reviews provided by top elite users require greater consideration as they are more engaged than typical Yelp users and hold the potential to influence others significantly through their reputation by sharing posts or reviews about the client's business.

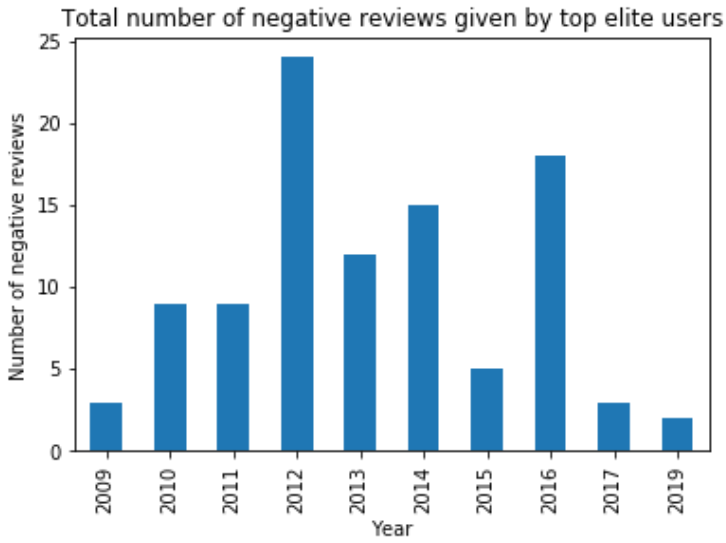
User Data

1. Elite users duration in the data



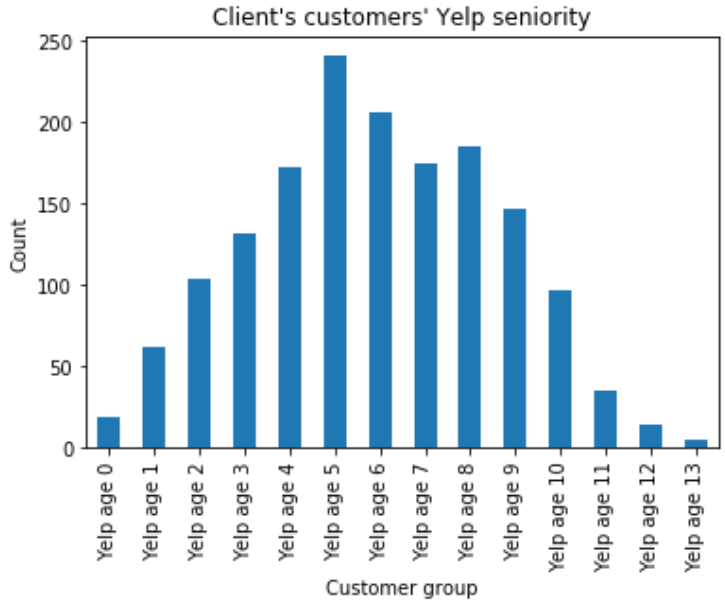
The data suggests that newly appointed elite users contribute the majority of elite-user reviews, while feedback from higher-seniority elite users decreases due to the larger proportion of non-elite users on Yelp. However, the restaurant's ability to attract high-seniority elite users in the past is positive as they can have significant influence on social media platforms and act as valuable word-of-mouth ambassadors.

2. Number of negative reviews provided by elite star users



3. How long users have been active on yelp

```
Yelp age 0      19
Yelp age 1      62
Yelp age 2     104
Yelp age 3     132
Yelp age 4     172
Yelp age 5     241
Yelp age 6     206
Yelp age 7     174
Yelp age 8     185
Yelp age 9     146
Yelp age 10     97
Yelp age 11     35
Yelp age 12     14
Yelp age 13      5
Name: yelp age, dtype: int64
```



The majority of the restaurant's customers have been using Yelp for a period of 5 to 8 years.

One notable observation is that there are more customers with lower Yelp seniority compared to those with higher Yelp seniority, indicating that the restaurant has been successful in attracting new Yelp users but has missed the opportunity to attract more seasoned users.

Most unfavorable reviews from restaurant clients are dated back more than three years.

Notably, in 2018, there were no negative reviews recorded for the client among this group of users.

4. Received Elite status quickly

name	yelping_since	review_count	elite	fans	average_stars	elite_since	yelp_elite_diff
Rachel	2010-11-26 23:13:07	1069	2010,2011,2012,2013,2014,2015,2016	448	3.97	2010-12-31	0.093240
Jenny	2009-11-20 16:02:35	737	2009,2010,2011,2012,2013,2014,2015,2016,2017,2018	53	3.66	2009-12-31	0.110497
Vincci	2009-11-20 02:39:38	880	2009,2010,2011,2012,2013,2014,2015,2016,2017,2018	93	3.89	2009-12-31	0.112025
Bryan	2018-11-18 23:23:49	57	2018	1	3.98	2018-12-31	0.115137
Seth	2010-11-12 06:04:38	321	2010,2011,2012,2016	21	3.78	2010-12-31	0.133553

There have been users who have become elite within months of starting to use YELP.

This is probably due to the high volume of reviews given and their fan following.

They seem to be influencers in deciding which restaurants are good or not.

5. Longest tenure of being Elite

name	yelping_since	review_count	elite	fans	average_stars	elite_tenure
Jessi	2006-08-23 19:56:06	381	2006,2007,2008,2009,2012,2013,2014,2015,2016,2...	45	3.87	12.0
Amy	2006-08-26 04:28:29	1839	2006,2007,2008,2009,2010,2011,2012,2013,2014,2...	109	3.87	12.0
Megan	2005-06-03 04:08:16	845	2006,2007,2008,2009,2010,2011,2012,2013,2014,2...	618	4.09	12.0
Leang	2006-06-18 04:18:27	2122	2006,2007,2008,2009,2010,2011,2012,2013,2014,2...	91	3.87	12.0
Ed	2006-07-22 01:22:26	4913	2006,2007,2008,2009,2010,2011,2012,2013,2014,2...	2034	3.66	12.0
Mimi	2005-04-14 23:50:38	1246	2006,2007,2008,2009,2010,2011,2012,2013,2014,2...	133	4.13	12.0
Mihir	2006-03-03 07:03:41	1387	2006,2007,2008,2009,2010,2017,2018	67	3.08	12.0

maximum tenure for users being elite is 12 years. The review counts and number of fans have. also been significantly high for these users.

6. Most reviews but not yet elite

name	yelping_since	review_count	elite	fans	average_stars
Kenneth	2011-06-10 03:52:07	6762		275	3.32
Inigo	2006-03-14 22:23:23	2552		63	3.55
Matt	2009-11-08 21:35:17	2255		167	4.64
Fancypants	2008-12-12 00:23:01	2162		166	3.69
Stefan	2013-05-01 15:07:23	1892		27	3.75

There have also been users who have a lot of reviews and fans but have not achieved elite status yet. For instance, Kenneth has the highest review count amount the top 4 elite users combined but still did not receive elite status.

9.0. Data Cleansing

Restaurant Data

1. Category was one column with a dictionary structure with multilevel heirarchy.

   Exploding this column to get the information of each attribute in a column format so that it can be parsed to machine learning models.

   Example input and output of the exploding the dictionary type columns

   Input -

```
{'RestaurantsPriceRange2': '2',
 'RestaurantsAttire': "u'casual'",
 'Alcohol': "u'none'",
 'Caters': 'False',
 'HasTV': 'False',
 'GoodForKids': 'True',
 'RestaurantsGoodForGroups': 'True',
 'RestaurantsTakeOut': 'True',
 'RestaurantsDelivery': 'False',
 'RestaurantsReservations': 'False',
 'BusinessParking': "{'garage': False, 'street': False, 'validated': False, 'lot': True, 'valet':
 'NoiseLevel': "u'average'",
 'WiFi': "'no'",
 'Ambience': "{'romantic': False, 'intimate': False, 'classy': False, 'hipster': False, 'divey':
 'OutdoorSeating': 'False'}
```


Output –

	Alcohol	HasTV	GoodForKids	RestaurantsGoodForGroups	RestaurantsTakeOut	RestaurantsDelivery	Restau
	eer_and_wine'	True	True	True	True	False	
	NaN	NaN	NaN	NaN	True	NaN	
	u'none'	False	NaN	NaN	True	False	

2. The weekday column needs to be label encoded and one-hot encoded so that the final check-in dataset can be used to join using the label encoded values. This will ensure that each business is getting their total check-in count for each day of the week.

Reviews Data

Reviews data is a text field which needed to address separately. The steps involved in cleaning text data were lowercase, removing whitespace, removing hyperlinks, removing emoticon (eg: ☺), removing emoji, removing chat words (eg: BRB: Be Right Back), removing punctuations, removing non-English words, removing stopwords (eg: a, an, and, are) and lemmatization (Keeping the root form of the word).

Input:

```
Had the most amazing time at 🍔FoodFusion🍕! The atmosphere was 🔥, and the staff
were super friendly 😀. The dishes were out of this world 🌟, especially the
"Gourmet Delight" — total flavor explosion! 🎉💥 I'm already planning my next visit,
and I'm bringing all my friends 🙌. Check out my full experience here: [https://www.yelp.com/] 🤔📸
```

Output

```
Output :

amazing time foodfusion atmosphere staff super friendly dish world especially
gourmet delight total flavor explosion im already planning next visit im
bringing friend check full experience
```

Check-in Data

The check-in data contains a comma separated field for the check-in into each business from 2010 to 2019. Manipulating this field to extract the actual count on each day needs to be done so that the target variable can be specified.

Input :

	business_id	date
0	--1UhMGODdWsrMastO9DZw	2016-04-26 19:49:16, 2016-08-30 18:36:57, 2016...
1	--6MefnULPED_I942VcFNA	2011-06-04 18:22:23, 2011-07-23 23:51:33, 2012...
2	--7zmmkVg-IMGaXbuVd0SQ	2014-12-29 19:25:50, 2015-01-17 01:49:14, 2015...

Output :

	business_id	weekday	checkin
0	--1UhMGODdWsrMastO9DZw	1	3
1	--1UhMGODdWsrMastO9DZw	2	1
2	--1UhMGODdWsrMastO9DZw	3	1

10.0. Summary

The analysis focuses on the Yelp Restaurant Reviews Challenge and provides valuable insights into restaurants, reviews, and user data. In the Restaurants Data section, the dataset is filtered for Toronto to predict customer influx, while restaurants with ratings above 3 stars are retained due to missing fields in lower-rated ones. Attributes and hours, stored as multilevel hierarchy dictionaries, offer potential for feature expansion. However, categories, though multiple, might not significantly impact the target due to their sparse nature. Additionally, review counts exhibit skewness, necessitating normalization techniques.

The Reviews Data segment highlights trends across years, months, and weekdays. Review counts steadily increase over the years, with average monthly counts ranging from 120 to 150, while Sundays attract the highest number of reviews. Furthermore, the average review rating displays a relatively stable trend over the years, except for a slight dip in 2019. Reviews from elite users hold influence due to their higher engagement. In User Data, insights cover elite users' contributions, their active duration on Yelp, and the potential impact of highly tenured elite users. Overall, the analysis underscores the relevance of data cleansing and preparation in uncovering meaningful patterns.

11.0. Data Preparation Needs

Expanding the attributes and hours variables within the restaurant dataset leads to an increase in the total features, now totaling 79. Among these, 74 are categorical variables, with 9 classified as dichotomous variables and 65 as polychotomous variables. An interesting aspect of

the polychotomous variables is their variation due to different encodings, resulting in multiple categories representing the same input. For instance, 'casual' and 'causal' are treated as distinct entities based on their encodings. To address this, it's crucial to process the various input encodings. By aggregating inputs with identical meanings but different encodings into one feature during the creation of dummy variables, we effectively increase the count of dichotomous variables. This manipulation step not only leads to a reduction in unique values but also contributes to improving the overall quality of the dataset.

It's worth mentioning that numerical variables such as latitude and longitude hold minimal significance in the current context and are considered for exclusion prior to commencing the model building process. Additionally, certain variables within the dataset contain 'None' as inputs, necessitating their replacement with NaN values to ensure the integrity and precision of subsequent analyses.

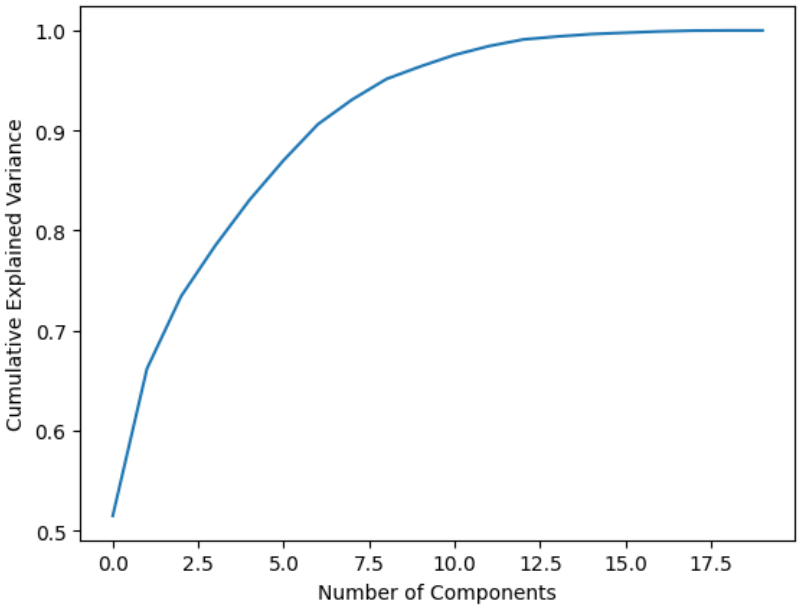
The preprocessing journey then goes into the transformation of categorical variables. For nominal categorical variables, the process of one-hot encoding is adopted, effectively transforming them into binary columns. On the other hand, ordinal categorical variables are subjected to label encoding, wherein they are assigned numerical representations based on their order or ranking. For example, an ordinal variable like 'Noise Level' with gradations from 'quiet' to 'very loud,' would be encoded numerically as 1 to 4.

```
Variable: (RestaurantsPriceRange2)
Unique Values: ['2' '1' '3' nan '4']

Variable: (RestaurantsAttire)
Unique Values: ["u'casual'" "'casual'" nan "u'dressy'" "'dressy'" "u'formal'" "'formal'"]

Variable: (Alcohol)
Unique Values: ["u'none'" "u'beer_and_wine'" "'none'" nan "u'full_bar'" "'beer_and_wine'"
 "'full_bar'"]

Variable: (NoiseLevel)
Unique Values: ["u'average'" nan "'loud'" "u'loud'" "'average'" "u'quiet'" "u'very_loud'"
 "'quiet'" "'very_loud'"]

Variable: (WiFi)
Unique Values: ["'no'" "u'no'" "u'free'" "'free'" nan "u'paid'" "'paid'"]

Variable: (Day)
Unique Values: ['Monday' 'Tuesday' 'Wednesday' 'Thursday' 'Friday' 'Saturday' 'Sunday']
```

A distinctive feature of the dataset pertains to the 'categories' column, which describes the diverse cuisines offered by individual restaurants. Although an initial inclination might be to generate distinct dummy variables for each specific cuisine, such an approach could result in an overly sparse matrix with limited informative value. Consequently, a shift in focus ensues, with consideration directed towards calculating the total count of different cuisines offered by each restaurant – an approach that provides a more effective representation of cuisine diversity. Furthermore, the 'Hours' variable captures the operational hours for each day. An interesting observation emerges: most restaurants adhere to uniform operating hours on weekdays while featuring varying schedules on weekends. Recognizing that information regarding operating hours doesn't substantially contribute to the modeling of the target variable, a pivot occurs. Attention shifts towards the duration of operating hours – specifically, the calculated difference between a restaurant's opening and closing times for each day.

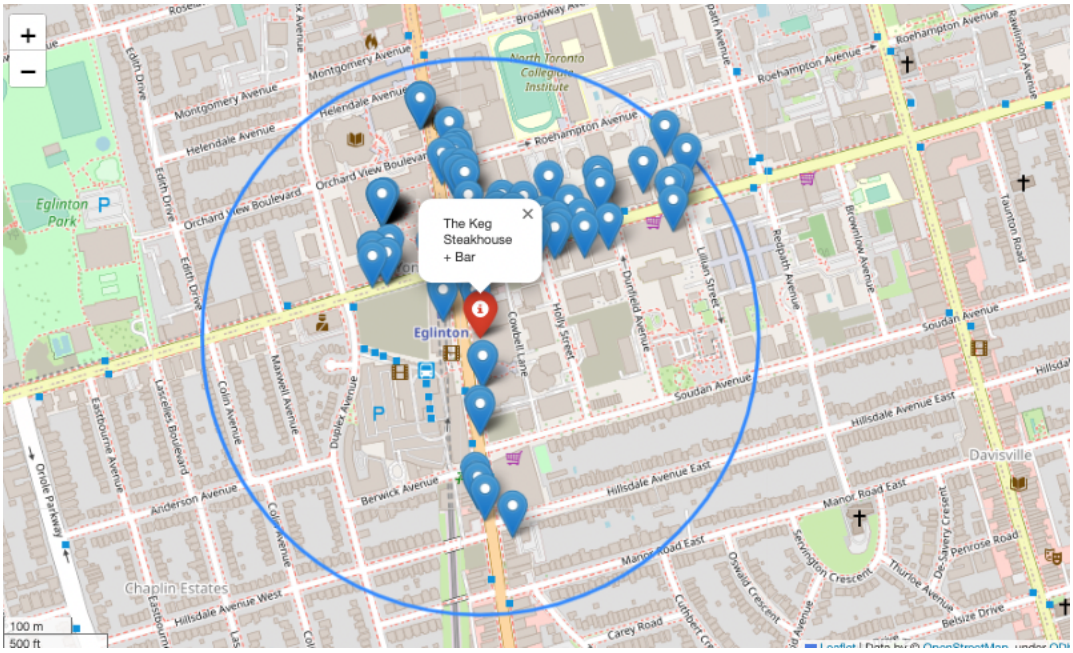
	operating_hours	start_time	end_time	duration
50325	8:0-21:0	1900-01-01 08:00:00	1900-01-01 21:00:00	13.0
50326	15:0-22:0	1900-01-01 15:00:00	1900-01-01 22:00:00	7.0
50327	12:0-22:0	1900-01-01 12:00:00	1900-01-01 22:00:00	10.0
50328	11:30-22:0	1900-01-01 11:30:00	1900-01-01 22:00:00	10.5

Leveraging the meticulously preprocessed and cleaned review data, the forthcoming analysis will pivot towards constructing a topic modeling model. This endeavor is anticipated to yield valuable insights that contribute to informed decision-making concerning customer footfall in diverse restaurant settings. Furthermore, the user dataset introduces a distinctive challenge, characterized by the presence of multiple numerical fields that have the potential to exert computational strain. In response, the approach of Principal Component Analysis (PCA) is harnessed. This technique enables the reduction of dataset dimensionality by encapsulating variance across numerical fields. The resulting n-principal components, which encapsulate the most critical information, are thoughtfully integrated into the modeling process. This fusion ensures that the approach remains not only efficient but also impactful. 5 principal components explain more than 90% of the variance and as a result, these 5 principal components will be used in modelling.



12.0. Feature Engineering

Restaurants in the vicinity can provide useful information to divide the customer flux and create relative point of interest locations. The latitude and longitude data will be used to explore restaurants in the vicinity for a 300-meter radius. Since downtown Toronto has a lot of restaurants the threshold is selected, and output variable will be used in modelling.



Even after dropping records which contained more than 40% missing values the number of records and features remained significantly high. As a result, the records which contained missing values have been dropped and no imputation technique has been used. This has not disturbed the distribution or proportion of each variable since the number of records are more than ~245K after merging all datasets.

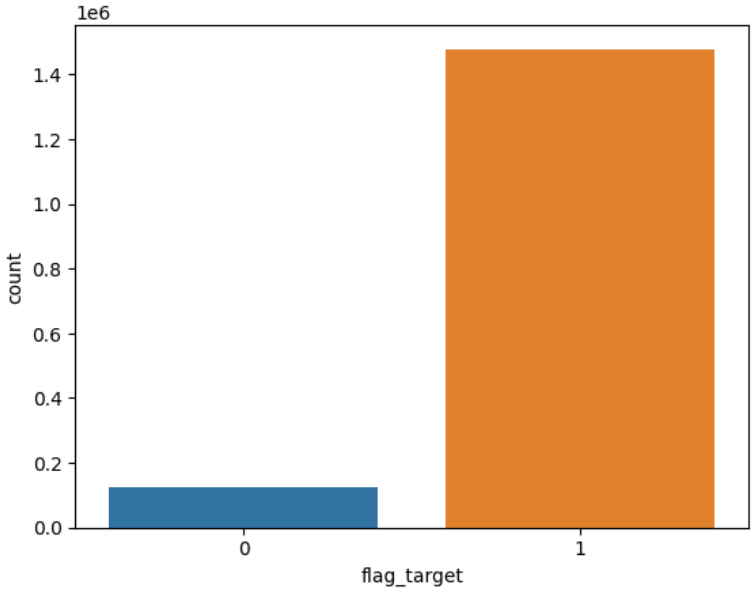
The next step, is using method called LDA (Latent Dirichlet Allocation) to help us understand our restaurant reviews better. Think of it like a detective tool that helps us find out what people are

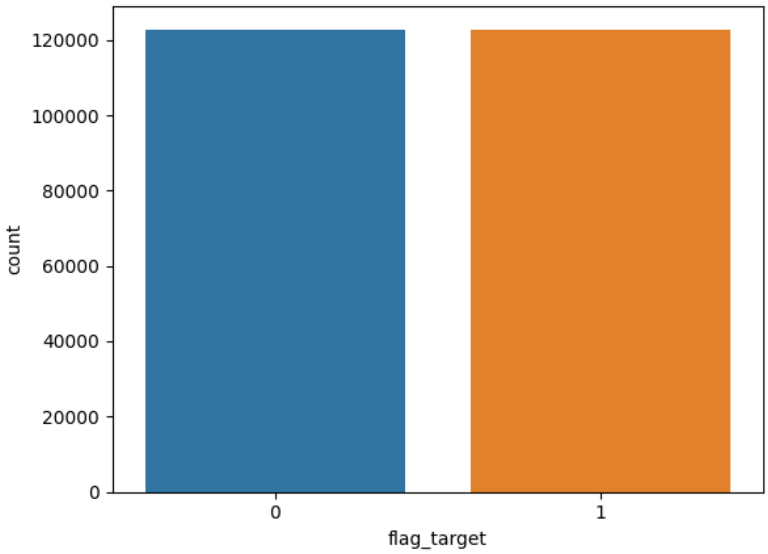
talking about the most in their reviews. We want to know the main things they like or don't like. This will help us improve our restaurants based on what our customers want.

LDA is like a magic wand that turns lots of reviews into small groups, each talking about a specific topic. For example, one group might talk about burgers, another about friendly staff, and so on. This will help us see the important things in the reviews without reading all of them. With these groups, we can make smarter decisions about what changes or improvements we need to make in our restaurants to make our customers even happier.

```
[(0,
  '0.012*"dish" + 0.011*"dessert" + 0.010*"lobster" + 0.009*"menu" + '
  '0.007*"delicious" + 0.007*"wine" + 0.007*"cream" + 0.007*"pasta" + '
  '0.006*"well" + 0.006*"dinner"'),
 (1,
  '0.023*"best" + 0.020*"always" + 0.020*"ive" + 0.019*"amazing" + '
  '0.018*"delicious" + 0.018*"love" + 0.014*"toronto" + 0.014*"friendly" + '
  '0.013*"staff" + 0.011*"every"'),
 (2,
  '0.015*"table" + 0.013*"order" + 0.010*"came" + 0.009*"didnt" + '
  '0.009*"server" + 0.009*"minute" + 0.008*"ordered" + 0.008*"even" + '
  '0.007*"asked" + 0.007*"friend"'),
 (3,
  '0.020*"sushi" + 0.019*"price" + 0.015*"roll" + 0.010*"quality" + '
  '0.010*"lunch" + 0.009*"portion" + 0.008*"pretty" + 0.007*"small" + '
  '0.007*"better" + 0.007*"come"'),
 (4,
  '0.031*"burger" + 0.024*"fry" + 0.015*"sandwich" + 0.015*"chicken" + '
  '0.014*"cheese" + 0.011*"sauce" + 0.010*"meat" + 0.008*"salad" + '
  '0.007*"side" + 0.007*"ordered"'),
 (5,
  '0.017*"pancake" + 0.017*"nice" + 0.015*"egg" + 0.014*"coffee" + '
  '0.014*"brunch" + 0.013*"breakfast" + 0.011*"drink" + 0.010*"friendly" + '
  '0.009*"definitely" + 0.009*"patio"'),
 (6,
  '0.029*"chicken" + 0.023*"dish" + 0.020*"rice" + 0.018*"fried" + '
  '0.014*"sauce" + 0.014*"pork" + 0.013*"beef" + 0.012*"soup" + 0.012*"spicy" '
  '+ 0.011*"noodle"'),
 (7,
  '0.085*"pizza" + 0.018*"dumpling" + 0.017*"chinese" + 0.016*"dim" + '
  '0.015*"sum" + 0.014*"crust" + 0.011*"pie" + 0.011*"slice" + 0.009*"italian" '
  '+ 0.008*"congee"'),
 (8,
  '0.035*"taco" + 0.019*"fish" + 0.015*"chip" + 0.014*"bowl" + 0.009*"ramen" + '
  '0.009*"try" + 0.008*"flavour" + 0.008*"mexican" + 0.008*"sauce" + '
  '0.008*"burrito"'),
 (9,
  '0.011*"beer" + 0.009*"bar" + 0.007*"dont" + 0.006*"im" + 0.006*"youre" + '
  '0.006*"people" + 0.005*"make" + 0.005*"know" + 0.005*"drink" + '
  '0.005*"toronto"')]
```

The target variable is highly skewed since it considers all the check-ins for a particular restaurant on a Monday between 2010 and 2019. Although this is not the actual flux of customers on a given day, the gist of whether a restaurant is busy or not can be extrapolated. Since yelp users account to a very small proportion of customers visiting a restaurant, the target variable has been binned into a binary target (Busy or Not Busy) based on the median value of the number of check-ins. This ensures that the skewed distribution is not impacting the target as median is not sensitive to outliers. Utilizing down-sampling technique in an imbalanced dataset with 1.6 million records enhances model performance by mitigating class imbalance, improving classification accuracy, and preventing bias towards the majority class. We are still left with 245K records which is decent for modelling.




120000
100000
80000
60000
40000
20000
0
count
0
1
flag_target


Review
business_id object
user_id object
review_id object
text object
date object
text_len int64
lda_topic int64
User
user_id object
PC1 float64
PC2 float64
PC3 float64
PC4 float64
PC5 float64
Restaurant
business_id object
address object
latitude float64
longitude float64
stars float64
review_count int64
is_open int64
categories int64
HasTV float64
GoodForKids float64
RestaurantsGoodForGroups float64
RestaurantsTakeOut float64
RestaurantsDelivery float64
RestaurantsReservations float64
NoiseLevel int64
OutdoorSeating float64
garage float64
street float64
validated float64
lot float64
valet float64
romantic float64
intimate float64
classy float64
hipster float64
touristy float64
trendy float64
upscale float64
casual float64
weekday int64
duration float64
RestaurantsPriceRange2_1 bool
RestaurantsPriceRange2_2 bool
RestaurantsPriceRange2_3 bool
RestaurantsPriceRange2_4 bool
RestaurantsAttire_casual bool
RestaurantsAttire_dressy bool
RestaurantsAttire_formal bool
Alcohol_beer_and_wine bool
Alcohol_full_bar bool
WiFi_free bool
WiFi_no bool
WiFi_paid bool
Day_Friday bool
Day_Monday bool
Day_Saturday bool
Day_Sunday bool
Day_Thursday bool
Day_Tuesday bool
Day_Wednesday bool
competitors_count int64
Checkin
business_id object
weekday int64
checkin int64
flag_target int64

13.0. Modeling Approach/Introduction

The initial step involved combining datasets using shared identifiers like business ID, user ID, and review ID to create a cohesive dataset. The primary focus was on the "flag_target" variable, denoting restaurant occupancy. Less relevant columns such as user and review details, location, and text were excluded, resulting in a dataset containing 52 predictors and 1 target variable. Within these, 42 predictors are categorical, with 40 being binary and 2 having multiple categories. Additionally, there are 12 numerical predictors, where 'categories' holds distinct values, while the rest are continuous variables. To maintain a good dataset balance given its size, a 70:30 training and testing split was chosen. To optimize model performance efficiently, we used a randomized search CV for hyperparameter tuning, avoiding the complexity of a grid search CV. Various models including Logistic Regression, Random Forest Classifier, Support Vector Machine, Decision Tree, and Naive Bayes were assessed. The final model was selected based on metrics like Accuracy, Precision, and Recall. This approach ensures a dependable model for forecasting restaurant occupancy with confidence and accuracy.

14.0. Model Technique #1 – Random Forest

The Random Forest Classifier is an ensemble machine learning algorithm used for both classification and regression tasks. It creates multiple decision tree models during training and combines their predictions to make final predictions. Each tree in the forest is constructed on a randomly selected subset of the data, and the final output is determined by the majority vote (classification) or average (regression) of the individual tree outputs.

Hyperparameter Optimization:

Hyperparameter optimization for the Random Forest Classifier is done using Randomized Search Cross-Validation Optimizing these hyperparameters is crucial to find the best configuration that results in improved model performance. By tuning these hyperparameters, the model can strike a balance between underfitting and overfitting, leading to better generalization on unseen data. The hyperparameters being optimized are:

- **n_estimators**: Number of decision trees in the forest.
- **max_depth**: Maximum depth of each decision tree.
- **min_samples_split**: Minimum number of samples required to split an internal node.
- **min_samples_leaf**: Minimum number of samples required to be at a leaf node.

Hyperparameter optimization is essential to ensure that the Random Forest Classifier performs at its best by finding the combination of hyperparameters that maximizes model performance. An unoptimized model might lead to suboptimal predictions due to underfitting or overfitting. Optimizing hyperparameters helps achieve better accuracy, generalization, and overall model effectiveness.

Computational Resources:

The Randomized Search Cross-Validation tests various combinations of the specified settings (10 times in this case) and evaluates each one through cross-validation (splitting the data into 10 portions). Consequently, the process of optimizing these settings can be computationally demanding, particularly when using more iterations and cross-validation portions.

Feature Importance:

The Random Forest Classifier calculates feature importance by evaluating how much each feature contributes to reducing uncertainty when making decisions in individual trees. The importance scores of features are averaged over all trees in the forest. Features that consistently lead to a higher reduction in uncertainty across various trees are deemed more significant. Feature importance scores provide insights into which aspects have the most predictive strength in the model.

Output

The hyperparameter tuning process for the Random Forest model yielded insightful results. Among the key predictors, 'competitors_count' demonstrated a significant impact with a score of 0.10251, indicating its strong influence on the model's performance. Additionally, 'review_count' held the highest score of 0.442355, underscoring its importance in predicting restaurant occupancy.

After experimenting with various parameter combinations, the best configuration was identified: 'n_estimators' set to 100, 'min_samples_split' at 5, 'min_samples_leaf' as 1, and 'max_depth' as None. This optimal parameter set led to an impressive outcome, with the model achieving an accuracy score of 90%. This robust performance reinforces the model's ability to accurately predict whether a restaurant will be busy or not based on the given features.

15.0. Model Technique #2 – Decision Tree Classifier

The Decision Tree Classifier is a fundamental machine learning algorithm used for both classification and regression tasks. It creates a hierarchical tree structure during training to make

predictions based on feature values. The tree structure consists of internal nodes representing feature conditions and leaf nodes representing class labels.

Hyperparameter Optimization:

Optimizing hyperparameters for the Decision Tree Classifier is crucial to finding the best configuration that results in improved model performance. By tuning these hyperparameters, the model can achieve the right balance between underfitting and overfitting, leading to better generalization on unseen data.

The hyperparameters being optimized are:

- max_depth: Maximum depth of the decision tree.
- min_samples_split: Minimum number of samples required to split an internal node.
- min_samples_leaf: Minimum number of samples required to be at a leaf node.

Hyperparameter optimization ensures that the Decision Tree Classifier performs at its best by finding the combination of hyperparameters that maximizes model performance. Suboptimal hyperparameters might lead to poor predictions due to overfitting or underfitting. Optimizing these hyperparameters results in improved accuracy, generalization, and overall model effectiveness.

Computational Resources:

Tuning the hyperparameters of the Decision Tree Classifier involves evaluating various combinations of settings. This process can be computationally demanding, especially when considering a wide range of hyperparameter values. As the depth of the tree and the number of splits increase, the complexity of the model grows, potentially requiring more computational resources.

Feature Importance:

The Decision Tree Classifier assesses feature importance by measuring how much each feature contributes to making accurate predictions. Features that lead to more significant reductions in impurity or entropy when splitting nodes are considered more important. The cumulative importance scores of features help in understanding which attributes have the most predictive power in the model.

Output:

After fine-tuning the model's parameters, the Decision Tree algorithm yielded impressive outcomes. The feature 'review_count' emerged as a highly influential factor with an importance score of **0.725609**. This underscores its pivotal role in predicting the target variable. Additionally, the significance of the 'competitors_count' feature was reaffirmed, though to a lesser extent, with an importance score of **0.036082**. The feature 'duration' also contributed meaningfully with a score of **0.038715**, enhancing the model's overall predictive prowess. Lastly, the feature 'text_len' had a relatively modest importance of **0.028032**, yet it remained an important contributor to the classification process. These insights into feature importance offer valuable guidance for comprehending the model's inner workings and the weightage assigned to different attributes in making predictions.

Through experimentation with various parameter combinations, the optimal configuration was determined: 'max_depth' set to 10, 'min_samples_split' at 5, and 'min_samples_leaf' set to 2. This parameter set led to a satisfying outcome, with the model attaining an accuracy score of 87.2%. This performance underscores the model's ability to classify instances accurately based on the given features.

16.0. Model Technique #3 - Logistic Regression

The Logistic Regression Classifier is a foundational tool in machine learning used for classifying items. It estimates the likelihood of an event occurring based on input variables. In this context, it helps us gauge how alterations in inputs influence the likelihood of a specific outcome.

Hyperparameter Optimization:

To enhance the performance of the Logistic Regression Classifier, we fine-tuned a key parameter known as 'C.' This parameter assists the model in finding the optimal balance between simplicity and complexity. After experimenting with various values, we determined that setting 'C' to 1 yielded the best results.

Computational Resources

The complexity of Logistic Regression primarily depends on the number of features and samples in the dataset. The training process involves iterative optimization to determine the optimal parameters. While Logistic Regression is generally less resource-intensive compared to more complex algorithms, as the number of features and samples increases, the computational demand can escalate.

Feature Importance:

We focused on three main factors: 'RestaurantsAttire_dressy,' 'romantic,' and 'validated.' For 'RestaurantsAttire_dressy,' a slight alteration could increase the likelihood of an event by around 70%. 'Romantic' had a more substantial impact, raising the likelihood by about 396%. Similarly, 'Validated' also showed significance, leading to an increase in the likelihood by approximately 200%.

Output:

The Logistic Regression Classifier demonstrated commendable performance. With 'C' set at 1, the model achieved an accuracy of roughly 89.79%. This outcome underscores the classifier's ability to accurately distinguish between items using the provided information. The systematic adjustments made to settings, insightful analysis of important factors, and thorough evaluation of model performance collectively resulted in a robust Logistic Regression Classifier that holds the capability to make informed predictions.

17.0. Model Technique #4 - Support Vector Machine

The Support Vector Machine (SVM) Classifier is a robust algorithm employed in both classification and regression tasks. It aims to find a decision boundary that maximizes the separation between different classes, enhancing predictive accuracy.

Hyperparameter optimization

The SVM model utilizes kernels such as 'linear' and 'rbf' to transform data into higher-dimensional spaces for improved classification. To optimize its performance, hyperparameters such as 'C' for regularization strength and the choice of kernel are crucial. Through rigorous tuning, the SVM model with the best parameters - 'kernel: linear' and 'C: 0.1' - achieved an impressive accuracy score of 89.78%, affirming its proficiency in classifying instances accurately.

Computational Resources

Considering computation cost, the SVM algorithm's complexity is influenced by the choice of kernel and the size of the dataset. While linear kernels generally require less computation, non-linear kernels like 'rbf' demand more processing power due to the transformation of data into a

higher-dimensional space. Therefore, it's essential to consider computational resources, especially when dealing with extensive datasets, to ensure efficient model training and validation.

Feature importance

In terms of feature importance, the SVM Classifier's interpretability lies in its coefficients assigned to each feature. These coefficients determine the influence of features on predicting outcomes. In this context, the 'Day_Sunday' feature exhibited the highest positive coefficient of 0.488655, followed by the 'Day_Tuesday' feature with a magnitude of 0.437867. Conversely, the 'Day_Monday' feature displayed a negative coefficient of 0.420262. These coefficients underscore the impact of different features on the SVM's decision-making process.

Output

The SVM Classifier, fine-tuned to optimal parameters, yielded an impressive accuracy of 89.78%, confirming its capability in precise instance classification. This outcome highlights the model's effectiveness and potential for accurate predictions in real-world scenarios.

18.0. Model Technique #5 - Naïve Bayes Classifier

The Naive Bayes Classifier is a probabilistic machine learning algorithm commonly used for classification tasks. It's based on the Bayes theorem and assumes that features are conditionally independent given the class label. Despite its simplifying assumptions, Naive Bayes can perform remarkably well on a wide range of datasets.

Computation Cost:

Compared to some other more complex algorithms, the Naive Bayes Classifier tends to have lower computational requirements. This is due to its simplified assumption of feature independence and its probabilistic nature, which makes it efficient for both training and making predictions.

Output:

After training and evaluating the Naive Bayes Classifier, it achieved an accuracy score of 76.33%. This score indicates the proportion of correctly classified instances, showcasing the model's ability to make accurate predictions based on the provided features. While not as complex as some other algorithms, the Naive Bayes Classifier's straightforward approach can still yield competitive results in various classification scenarios.

18.0. Model Comparison

Comparing the performance of different models in this classification scenario, it's evident that the Random Forest algorithm stands out as the best performer. The evaluation metrics, including precision and recall, along with the accuracy score, emphasize the superiority of the Random Forest model.

Precision and recall are essential metrics in classification tasks. Precision measures the proportion of correctly predicted positive instances among all predicted positive instances, indicating the model's ability to avoid false positives. Recall, on the other hand, assesses the model's ability to correctly identify positive instances among all actual positive instances, thus addressing false negatives. The Random Forest model achieves high precision and recall scores of 0.95 and 0.82, respectively, for the positive class, as indicated in the classification report.

Furthermore, the overall accuracy score of 90% underscores the model's proficiency in accurately classifying instances. The Random Forest algorithm's strengths lie in its ensemble nature, where it combines multiple decision trees to produce more robust predictions. This allows it to mitigate overfitting while maintaining strong predictive power.

Additionally, considering computation cost, the Random Forest model strikes a balance between complexity and performance. While it involves training multiple decision trees, its parallel processing capability and feature subsampling techniques contribute to efficient use of computational resources. This makes the Random Forest algorithm a favorable choice for classification tasks, especially when high accuracy, precision, and recall are crucial.

19.0 Model Selection

Among the array of machine learning algorithms, the Random Forest Classifier stands out as the optimal choice for addressing the key objectives of the use case. This model selection is based on its ability to seamlessly integrate a diverse range of data sources, handle both continuous and categorical targets, and provide interpretable insights into the factors influencing restaurant customer visits.

20.0 Model Theory

The Random Forest Classifier operates on the principle of ensemble learning, constructing a multitude of decision trees during training. Each tree is trained on a subset of the data, and their predictions are aggregated to form the final output. This ensemble approach mitigates overfitting and enhances predictive accuracy, making it particularly well-suited for the complex and noisy restaurant data.

20.1 Model Assumptions and Limitations

While the Random Forest model doesn't impose stringent assumptions on the data distribution, it assumes that each feature contributes to the target independently, which may not hold true in all scenarios. Additionally, the model's performance can be influenced by the quality and quantity of the training data. Despite these limitations, the model's robustness and interpretability outweigh its assumptions.

21.0 Model Sensitivity to Key Drivers

The Random Forest Classifier excels in determining the sensitivity of predictions to key drivers, as highlighted by the hyperparameter tuning process. Among the predictors identified, 'competitors_count' exhibited a significant impact,with an importance score of 0.10251. This underscores the significant role competitors play in the restaurant’s performance. 'review_count' emerged as a pivotal factor with the highest importance score of 0.442355. This emphasizes the critical influence of customer reviews on predicting restaurant occupancy, reflecting the profound impact of customer sentiments on business outcomes.

22.0 Additional Models to Address Business Objectives

While the Random Forest Classifier serves as a strong foundation for addressing your business objectives, considering supplementary models like ensemble methods such as Voting Classifier or boosting algorithms can provide a comprehensive view. These models can harness the collective strengths of diverse algorithms to further refine predictions and glean deeper insights into the factors influencing restaurant performance.

Incorporating the Random Forest Classifier into analysis enables a data-driven approach to optimize inventory levels, staff allocation, and sales targets. Its adaptability, robustness, and capacity to provide actionable insights align seamlessly with the objectives of enhancing restaurant operations and making informed decisions for growth.

23.0 Impacts on Business Problem

The recommended predictive models offer a comprehensive solution to enhance various aspects of restaurant operations and strategic decision-making. By leveraging these models, businesses can identify high-demand regions and optimal restaurant locations, considering factors like cuisine preferences and operating hours. Moreover, aligning promotional efforts with forecasted demand surges can strategically maximize their impact, driving customer turnout. The models enable businesses to anticipate busy periods, optimizing staff scheduling to avoid overstaffing during slower times and ensuring resource efficiency. Additionally, by accurately forecasting demand, the models facilitate inventory optimization, leading to reduced waste and operational costs. Implementing just-in-time supply strategies based on demand forecasts helps avoid stockouts, ensuring seamless operations.

24.0 Recommended Next Steps

Following the successful implementation of the models, the next steps involve leveraging their insights to drive actionable improvements and optimizations. Specifically, businesses can focus on the following initiatives:

1. **Strategic Expansion:** Utilize the predictive models to identify regions with high demand potential and optimal restaurant locations. Consider factors like cuisine preferences and

operating hours to strategically expand the business footprint, capitalizing on customer preferences.

2. **Promotion Planning:** Align promotional efforts with forecasted demand surges to achieve maximum impact. By targeting high-demand periods, businesses can attract a larger customer base and enhance brand engagement.
3. **Resource Optimization:** Anticipate busy periods using the predictive models to optimize staff scheduling. This ensures efficient resource allocation, preventing overstaffing during low-demand periods and ensuring exceptional service during peak times.
4. **Inventory Management:** Optimize inventory levels based on demand forecasts, minimizing waste and costs. Implement just-in-time supply strategies to avoid stockouts and maintain efficient inventory turnover.
5. **Customer Experience Enhancement:** Leverage insights from the models to enhance customer experiences. Adapt offerings, service levels, and staffing based on predicted customer influx, leading to improved satisfaction and loyalty.
6. **Continual Model Refinement:** Regularly update and refine the models based on real-world performance data. Incorporate new variables, trends, or external factors to enhance predictive accuracy and relevance.
7. **Feedback and Adaptation:** Gather feedback from restaurant owners, managers, and stakeholders to fine-tune model predictions and recommendations. Continuously adapt strategies based on insights and business needs.

25.0 References

(2023). Retrieved from https://www.mordorintelligence.com/industry-reports/canada-foodservice-market

Banerjee, A. (2022, November 2). *Hyperparameter tuning using randomized search*. Analytics Vidhya. https://www.analyticsvidhya.com/blog/2022/11/hyperparameter-tuning-using-randomized-search/

CR, A. (2020, July 26). *Topic modeling using Gensim-LDA in python*. Medium. https://medium.com/analytics-vidhya/topic-modeling-using-gensim-lda-in-python-48eaa2344920

Kapadia, S. (2019a, April 14). *Topic modeling in Python: Latent dirichlet allocation (LDA)*. Medium. https://towardsdatascience.com/end-to-end-topic-modeling-in-python-latent-dirichlet-allocation-lda-35ce4ed6b3e0

Kapadia, S. (2019b, August 19). *Evaluate topic models: Latent Dirichlet allocation (LDA)*. Medium. https://towardsdatascience.com/evaluate-topic-model-in-python-latent-dirichlet-allocation-lda-7d57484bb5d0

Khachatryan, S. (2023). Retrieved from https://orders.co/blog/how-to-use-data-and- analytics-to-improve-your-restaurants-marketing-performance/#:~:text=Analyzing%20data%20such%20as%20sales,insights%20gained%20from%20data%20analysis.

kumar, S. (2019, August 20). *Getting started with text preprocessing*. Kaggle. https://www.kaggle.com/code/sudalairajkumar/getting-started-with-text-preprocessing

Lasek, A., Cercone, N., & Saunders, J. (2016). Restaurant Sales and Customer Demand Forecasting: Literature Survey and Categorization of Methods.

Liu, L., Bhattacharayya, S., Sclove, S., Chen, R., & Lattyak, W. (2001). Data mining on time series: an illustration using fast-food restaurant franchise data. Elsevier, 37, 455–476. doi:22 Jan , 2001

Mok, T. (2020). Retrieved from https://www.blogto.com/eat_drink/2020/09/new-study- shows-torontos-restaurant-reservations-have-been-climbing-but-still-drastically- lower-than-last-years/

Pandian, S. (2023, July 14). *K-fold cross validation technique and its essentials*. Analytics Vidhya. https://www.analyticsvidhya.com/blog/2022/02/k-fold-cross-validation-technique-and-its-essentials/

R, S. E. (2023a, July 5). *Understand random forest algorithms with examples (updated 2023)*. Analytics Vidhya. https://www.analyticsvidhya.com/blog/2021/06/understanding-random-forest/

R, S. E. (2023b, July 5). *Understand random forest algorithms with examples (updated 2023)*. Analytics Vidhya. https://www.analyticsvidhya.com/blog/2021/06/understanding-random-forest/

Saini, A. (2021, August 26). *Conceptual understanding of logistic regression for data science beginners*. Analytics Vidhya. https://www.analyticsvidhya.com/blog/2021/08/conceptual-understanding-of-logistic-regression-for-data-science-beginners/

Shah, R. (2023, July 24). *How to build word cloud in python?*. Analytics Vidhya. https://www.analyticsvidhya.com/blog/2021/05/how-to-build-word-cloud-in-python/

Tran, K. (2023, August 1). *Pyldavis: Topic Modelling Exploration Tool that every NLP data scientist should know*. neptune.ai. https://neptune.ai/blog/pyldavis-topic-modelling-exploration-tool-that-every-nlp-data-scientist-should-know#:~:text=Topic%20modeling%20involves%20counting%20words,can%20group%20different%20words%20together.

Vu, D. (2023, February 23). *Python word clouds tutorial: How to create a word cloud*. DataCamp. https://www.datacamp.com/tutorial/wordcloud-python