Restaurant revenue prediction

Team 38

Anjali Shenoy Rehas Sachdeva Saumya Rawat

Background

Enterprises today, highly motivated in the art of finding **anomalies, patterns and correlations** within data-sets.

Want to improve their **online reviews to attract clientele** or seek to establish a new business that is mindful of what **drives good reviews**, particularly true for **restaurants and food establishments.**

Introduction

Several studies conducted to look at the **correlation between a restaurant's success and its reviews and ratings.**

TFI is behind the famous brands like Burger King, Sbarro, Popeyes etc, interested in **extrapolating their data across geographies and cultures.**

We will be working with a **TFI data set of about 1 lakh Turkish restaurants.**

Problem Statement

Supervised learning problem

Objective

To develop a model and a set of preprocessing procedures to accurately predict the annual restaurant sales of 100,000 regional locations using various parameters.

Problem Challenges

- The size of training dataset is 137 samples while that of test dataset is 1,00,000 samples. This is a **large disparity**.
- **We don't have the ground truth for our test data**. So we cannot use sophisticated performance measures like precision, recall, k-fold cross validation etc. **We only know RMSE** for the entire test data.
- Whether to predict **revenue or log(revenue)** as we see that training data follows **normal distribution** when taken with log(revenue). But we can't say the same about test dataset.
- **Parsing** the data types of various attributes.
- **Unaccounted problem:** the disparity between the features for the training set and test set as the test set contains more information than the training set.
- **Categorical vs continuous problem:** whether the obfuscated P-Variables should be treated as categorical or continuous.
- **Zero problem:** For certain P-Variables, a large number of samples contain zero values and are dependent among each other such that if one p-variable has zero on a certain row, the probability that other p-variables take on a zero value is high.

Dataset Description

- The dataset based on 1 lakh Turkish restaurants, is uploaded on **Kaggle**.
- Size of training dataset: 137 samples, Size of test dataset: 1,00,000 samples.
- The **43 data fields** provided are:

  ID: Restaurant ID

  Open Date: Date that the restaurant opened in the format M/D/Y

  City: The city name that the restaurant resides in

  City Group: The type of city can be either big cities or other

  Type: The type of the restaurant where FC - Food Court, IL - Inline, DT - Drive through and MB - Mobile.

  P-Variables (P1, P2, ... ,P37): Obfuscated variables within three categories: demographic data, e.g population, age, gender; real estate data e.g car park availability and front facade; commercial data e.g points of interest, other vendors, etc. It is unknown if each variable contains a combination of the three categories or are mutually exclusive.

  Revenue: Annual (transformed) revenue of a restaurant in a given year and is the target to be predicted.

Feature explanation

- Sales depend on the **location and type of city** it is in. If the city is a **metropolitan city,** it will have a **larger customer base** than a town, and hence revenue generated will be different in both these cities.
- On a similar note, revenues generated will be different for different restaurant types- **Drive through** will attract more customers in a remote area where as a **restaurant will attract more customers if it is placed in the heart of the city**.
- The open date attribute doesn't do much as such. But if processed to get **number of days a restaurant stays open, year of opening, month of opening**, we can get an idea of **cyclical or seasonal patterns** that affect revenue.
- Apart from these, we have **demographic attributes** e.g population, age, gender; **real estate data** e.g car park availability and front facade; and **commercial data** e.g points of interest, other vendors, etc. These also decide sales to varied extents. They can also be **correlated.**

Feature Extraction and Selection

- **Training and test dataset** - available as csv files containing 137 and 100,000 samples respectively. The input data is too large to be processed and majority of the data fields are **obfuscated variables** without giving any prior knowledge of each one. The data is pre processed by the following methods :
- We use **histograms to see that log(revenue)** follows an approximately **normal distribution**. So choosing target variable as log(revenue) instead of revenue improves performance of base models.
- **Opening date cannot simply be assumed to be a factor** so two additional features are created: month that they opened and the year that they opened. These two features can potentially help **proxy seasonality differences** since restaurant **revenues are highly cyclical**.
- Since the restaurant **'Type'- 'MB' is not present in the training but available in the Test**, the Test set is modified so each mobile type restaurant is matched with a non-mobile type restaurant through finding the most similar features, as measured by euclidean distance.
- **Similarly for 'City' since the number of cities in the test set is more than the training set**, using KNN all 137 cities are clustered on the basis of P Variables that best describe Geographical locations and then these clusters are assigned to the city column of the datasets. To know exactly what each p-variable represents, under the assumption of mutually exclusive categories, a change in the mean over each city should elicit a change in certain p-variables. And using box plot of mean p-variables over each city, P1, P2, P11, P19, P20, P23, and P30 are identified to be approximately a good proxy for geographical location.
- We also use PCA for the P-Variables because we aren't given exactly what these represent. **They can be correlated**. So PCA can represent them better.

Validation techniques used

- We use **histograms to see that log(revenue)** follows an approximately **normal distribution.** So choosing target variable as log(revenue) instead of revenue improves performance of base models.
- **Visualizing a box plot of mean** of **P-variables** for each city helps us identify which P-variables are majorly geographical.
- **Davies-Bouldin index for K-Means clustering on variables**: P1,P2,P11,P19,P20,P23,P30 helps us validate the **best K** to be used as number of clusters.

Performance metrics

Root Mean Squared Error (RMSE)

Submissions are scored on the root mean squared error. RMSE is very common and is a suitable general-purpose error metric. Compared to the Mean Absolute Error, RMSE punishes large errors:

$$\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2},$$

where y hat is the predicted value and y is the original value.

Algorithms Mentioned in the paper

- **KNN** to account for **types of restaurants** that are there in test data but not in training data. The idea is to match those **unknown types with known types** (from training data) using nearest neighbour method based on rest of the attributes. **KNN is also used for zero problem**, to get an appropriate value for missing values of various attributes.

- **KMeans to account for cities** of restaurants that are there in test data but not in training data. The idea is to match those **unknown cities with known cities** (from training data) using clusters of cities.

- **PCA to reduce the dimensionality** of the data, especially useful for the p variables, because we are given no information about what they may represent, and so could themselves be **linear combinations** of the "real" variables, or simply be **highly dependent**.

- **Random forest, SVM: Support Vector Machines with Regression, and an ensemble method** based on combination of these two models, enable us to predict the annual revenue of a restaurant.

Algorithms we additionally experimented on

- **Extra Trees Classifier:** Extension random forest, used for restaurant type classification

- **Ridge model**: as a part of the ensemble.

Analysis and Results on paper based algorithm

- Pre-processing on City and Type greatly improves performance over baseline models.
- Log transformation on revenue does not improve real test set performance although training set results are very promising.
- Treating zero problem with PCA or KNN doesn't improve performance probably due to large misspecification errors of the treatment models that introduce more noise rather than clarity.
- Apart from zero problem treatment, all solutions proposed in the paper together lead to least RMSE.

WHAT WE DID EXTRA

Our ideas

- For restaurant type such as T_MB, T_DT which were very rare we dropped it from the table and substituted those values with predicted type of **Extra Trees classifier** (a basic version of Random Forest)
- For the categorical and continuous problem we did a **one hot encoding** for "P" variables and took them as categorical.
- A certain set of columns are either mostly all zero or all non-zero. We added a feature to mark this, storing the count of zero columns. This also greatly improved accuracy.
- We tried an alternative treatment for City problem, replacing cities with their total counts. This also greatly improved accuracy.
- We also **scaled all input features** between 0 and 1. This greatly improved accuracy.
- We tried different ways to taking the revenue apart from log(revenue), like **sqrt of revenue**, etc.

We also looked at the ridge model and tried to make it as an ensemble with SVM and random forest.

What did we observe?

Ridge model as a standalone gave us even better results than the models mentioned in the paper and a Kaggle rank of 7. The corresponding RMSE score was 1,750,100.

Languages and Toolkits

- Python
- SKlearn toolkit

Scope

- The solution is **only applicable to Turkish restaurants** and locations on which the data is based. A different location based data may need a different kind of ensemble for accuracy.
- It is limited to only **annual and not seasonal** revenue analysis.

Conclusions

When training data is small in size, **the simplest model often gives the best results.** In our case, compared to SVM and Random Forest the ridge model gave us the best results and heavily reduced our RMSE values.

Timeline

Preprocessing
By Mid September

Ensemble models
By September end

Compare Models and try to improve results
By Mid October

Experiment with Ridge models and other methods
By October 3rd week

Visualize and Interpret Results
By October end