Restaurant revenue prediction

Team 38

Anjali Shenoy Rehas Sachdeva Saumya Rawat

Problem Statement

Supervised learning problem

Objective

To develop a model and a set of preprocessing procedures to accurately predict the annual restaurant sales of 1,00,000 regional locations using various parameters.

Dataset Description

Base of dataset	Around 1 Lakh Turkish Restaurants
Source of Dataset	TFI via Kaggle
Size of training dataset	137 samples
Size of test dataset	1,00,000 samples

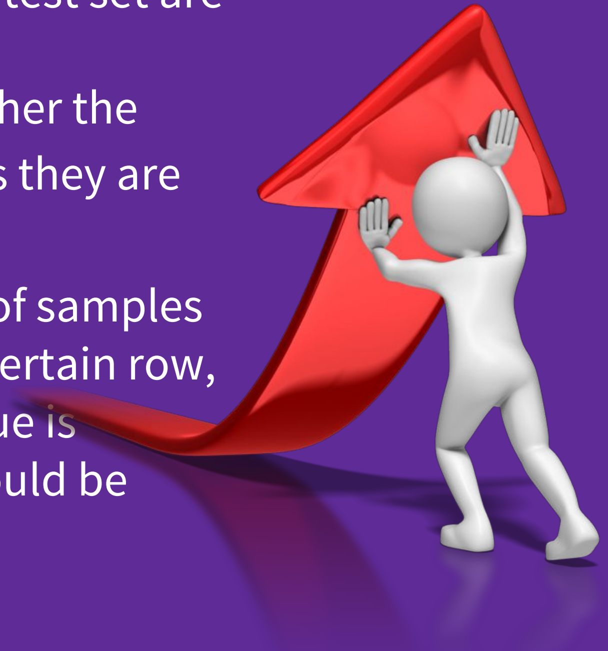
Data Fields

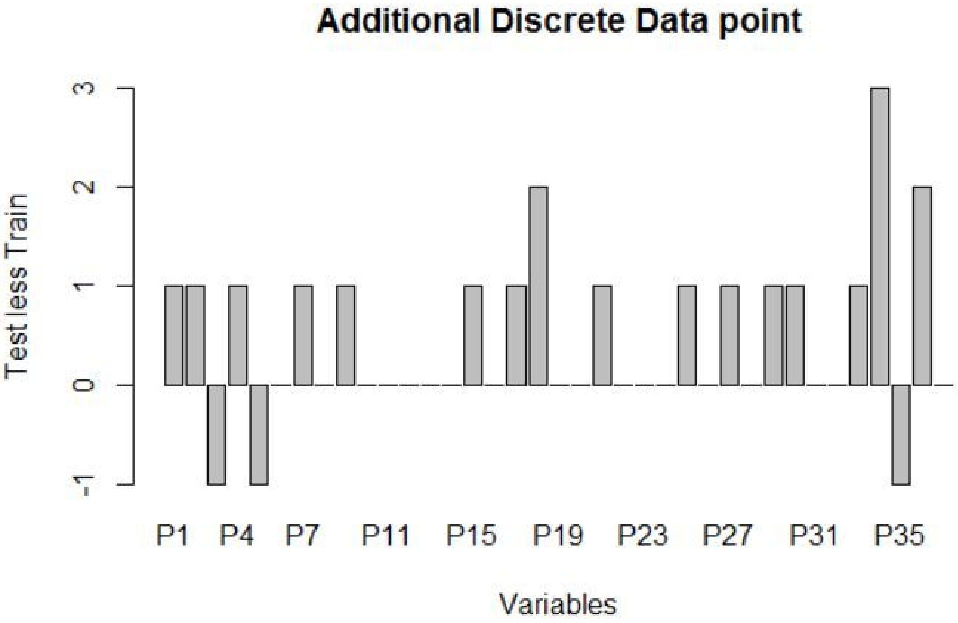
ID	Restaurant ID
Open Date	Date that the restaurant opened in the format M/D/Y.
City	The city name that the restaurant resides in.
City Group	The type of city can be either big cities or other.
Type	The type of the restaurant: FC - Food Court, IL - Inline, DT - Drive through & MB - Mobile.
P-Variables (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 revenue of a restaurant in a given year and is the target to be predicted.

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, via submission on Kaggle.
- Whether to predict **revenue, log(revenue) or sqrt(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.

- **Unaccountability problem:** The test set contains more information than the training set.
  - Type ‘MB’ missing in training set.
  - 34 cities in the training set but 57 in the test set.
  - Number of unique values of each P Variables in the test set are almost always more than the training set.
- **Categorical vs continuous problem:** it is unclear whether the obfuscated P-Variables should be treated categorical as they are discrete in nature.
- **Zero Problem:** For certain P-Variables, a large number of samples contain zero values and if one p-variable has zero on a certain row, the probability that other p-variables take on a zero value is high. For the train dataset, this probability is 1. These could be missing values or simply be highly correlated.





Additional Discrete Data point

Language & Toolkit

- Python (Jupyter Notebook)
- SKlearn toolkit

Solution

Feature Extraction and Selection

- **Parsing Open Date:**
  - calculate the **number of days open** by taking the difference between the opening date and an arbitrary constant such that it is later than the latest opening date of all samples. The date chosen is January 1, 2015.
  - **two additional features, month that they opened and the year** that they opened, to potentially help proxy seasonality differences since restaurant revenues are highly cyclical.
- **Enhancing the ideas in the paper:**
  - **Log transform on Days open, month, year etc.** Doing this for Days open improve the results.
  - D**ropping month and year attributes**.

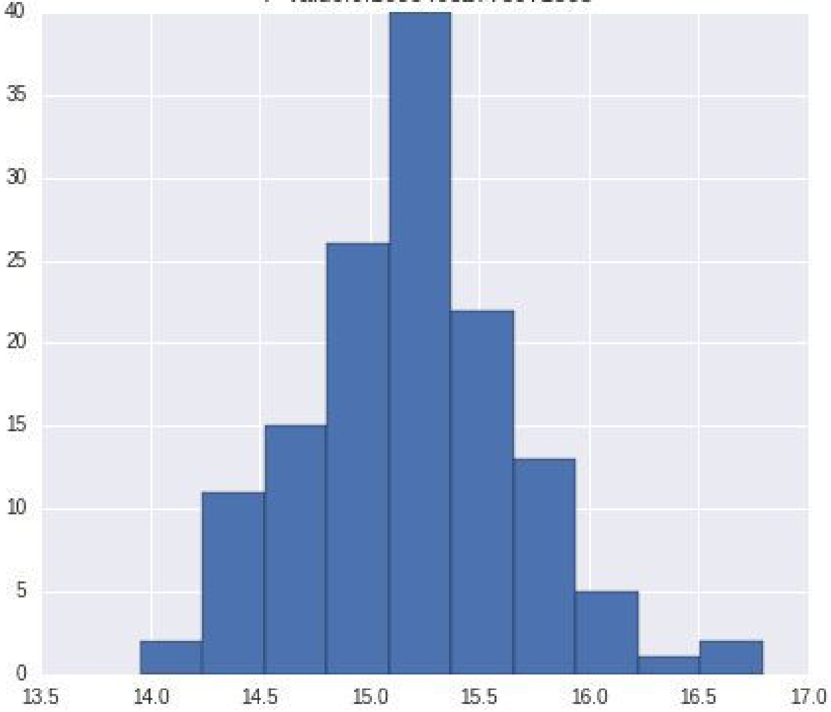
Date Parsing and Processing	**Submission RMSE**
Adding number of days open, month and year of opening	1755350
Log treatment on days open, month and year of opening	1750539
Log treatment on days open, dropping month and year	**1719022**

Feature Extraction and Selection

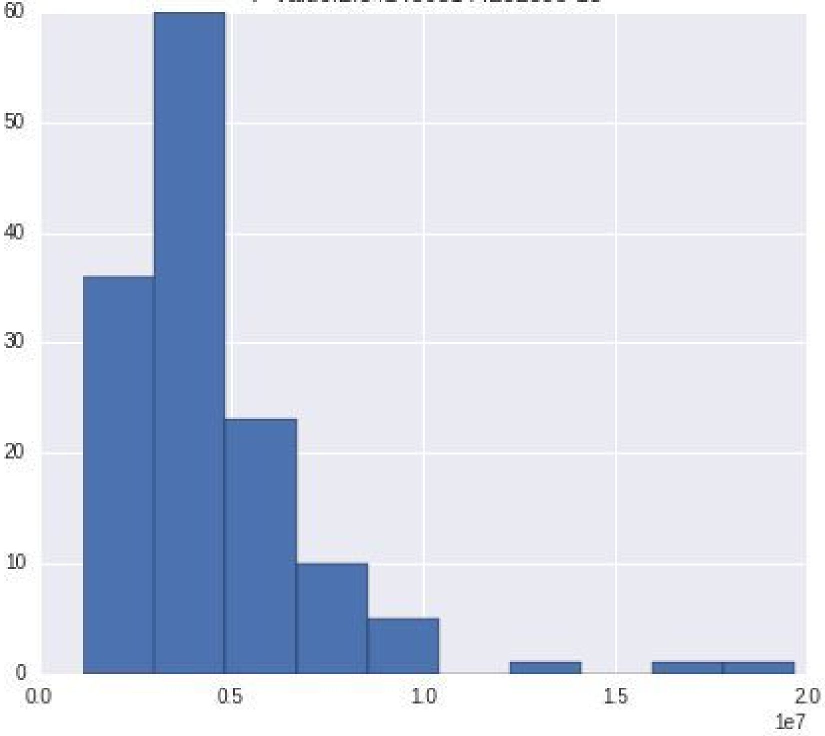
- **Modelling distribution in Revenue:**
  - H**istograms** show that log(revenue) follows an approximately **normal distribution**.
- Reduces **the skewness in the distributions of data** and improves performance of **models.**
- **Enhancing the ideas in the paper:**
  - other **tricks like Sqrt treatment** on Revenue. **The maximum and minimum predictions made on test data were more extreme in case of Sqrt treatment**, so it probably gave better results for extreme points in test data.

Transform on Revenue	**Submission RMSE**
No transform	1762259
Log transform	1773825
Sqrt transform	**1719022**

Feature Extraction and Selection

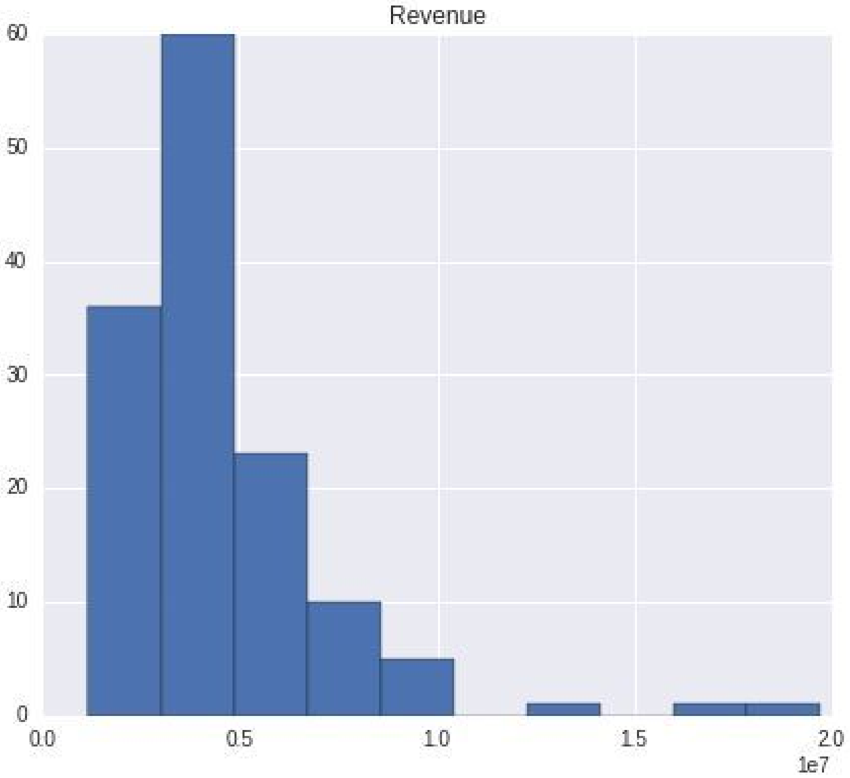


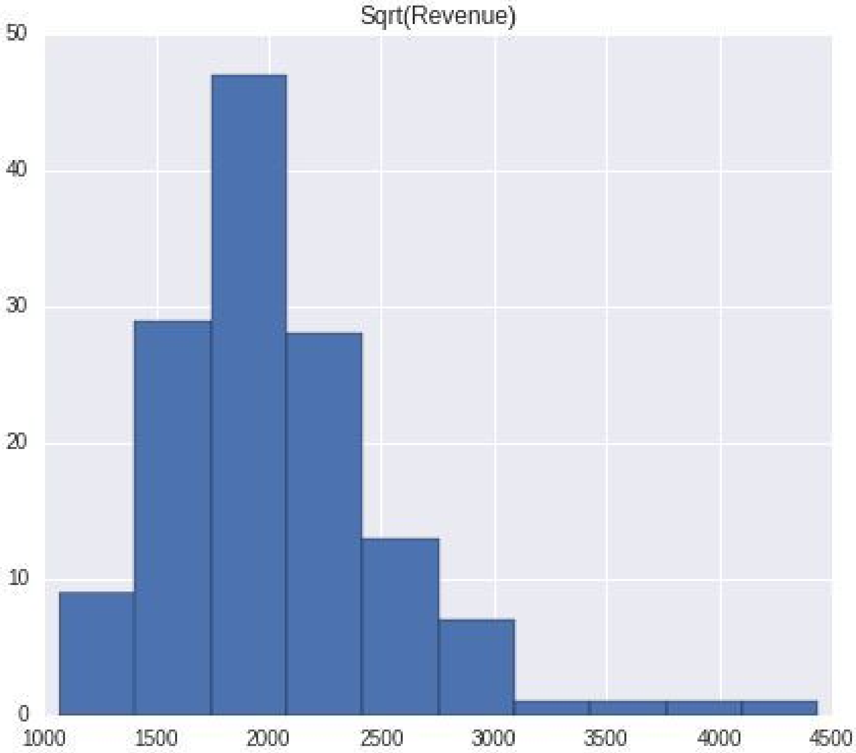
Log(Revenue)
P-value:0.16584032773971558



Revenue
P-value:2.341430814425255e-13

Feature Extraction and Selection





Feature Extraction and Selection

- **Dealing with Type categorical variable:**
  - **query matrix** - test set where each row is a mobile type restaurant.
  - **search matrix** - test set where each row is not a mobile type restaurant.
  - **KNN algorithm to classify query matrix data as one of IL, FC an DT.**
- **Enhancing the ideas in the paper:**
  - other classifiers like **Extra Trees Classifier**: “Extremely randomized trees” where at each step the entire sample is used and decision boundaries are picked at random, rather than the best one.
  - But **KNN** performed slightly better.

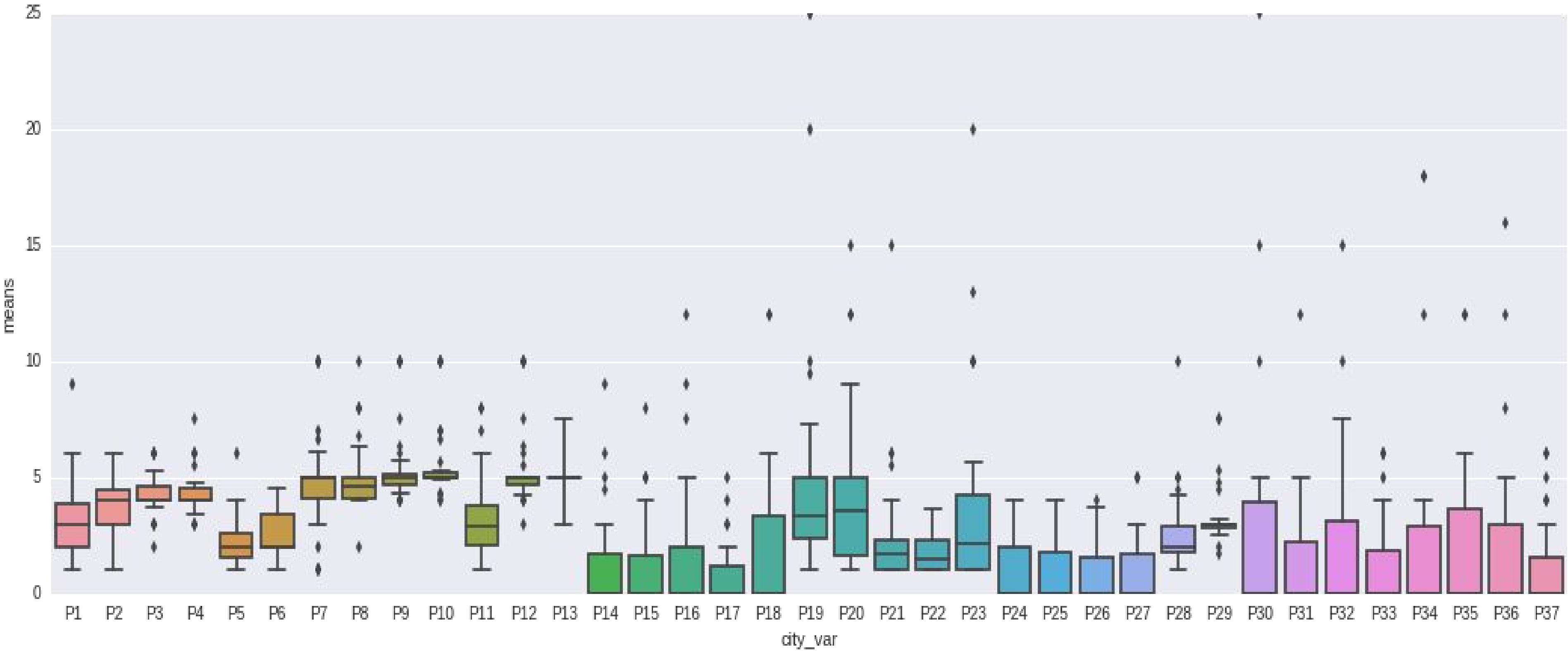
Type Treatment	**Submission RMSE**
Extra Trees Classifier to replace rary types ‘MB’ and ‘DT’	1750100
KNN Treatment to replace ‘MB’ which is unaccounted in train data	**1719022**

Feature Extraction and Selection

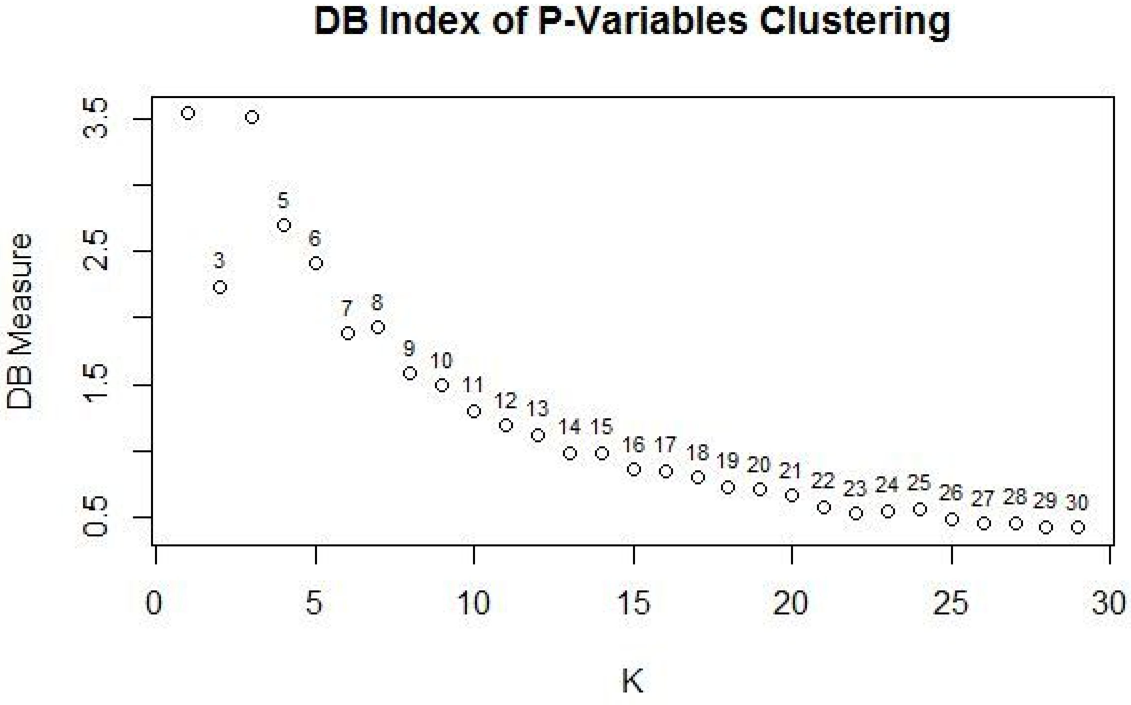
- **Dealing with City categorical variable:**
  - **K-Means clustering**
- **Choosing P variables for K-Means:**
  - Some P-variables are geographical.
  - average P-Variable for each subset of cities and plot the deviation over all cities.
  - P1, P2, P11, P19, P20, P23, and P30 are good proxy.
- **Choosing best K:**
  - Davies and Bouldin index: ratio of within-cluster distance and between-cluster distance.
  - DB index minimized for K=20 to K=25.
- **Enhancing the ideas in the paper:**
  - replacing cities with their total counts.
  - unaccounted cities have lower counts.

City treatment	**Submission RMSE**
Replace with city counts	**1719022**
K Means (K=20)	1750803

Choosing P variables for K-Means



Selecting optimal K for K-Means



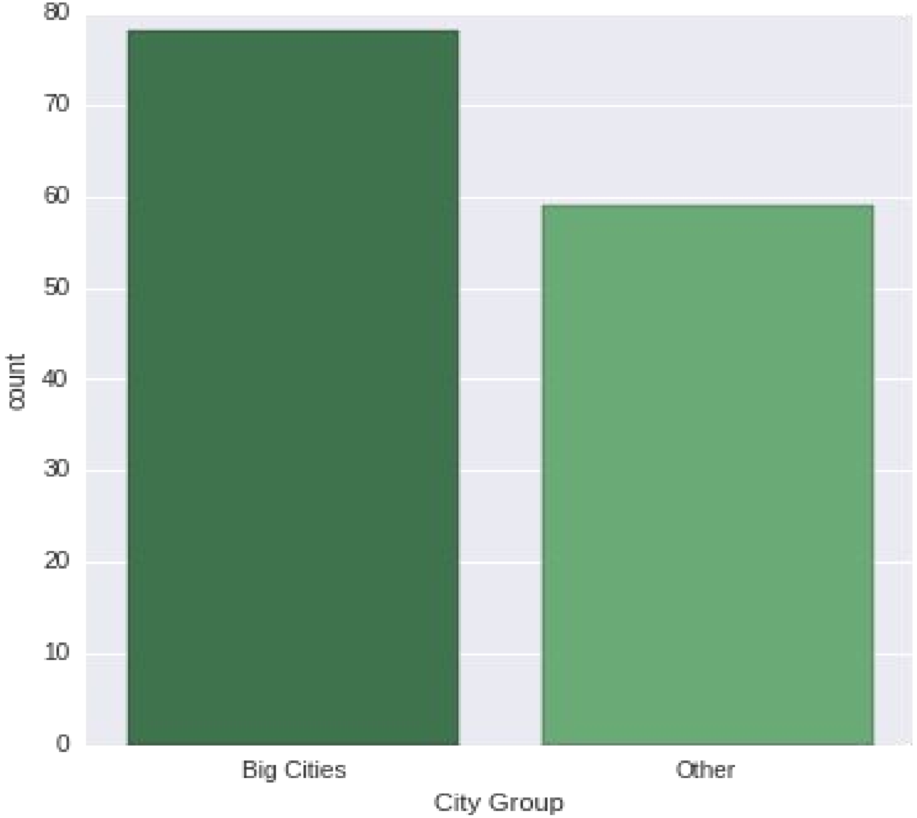
Feature Extraction and Selection

- **Dealing with City Group categorical variable:** one hot encoding as the two categories of City Group both appear very frequently.
- **Dealing with P-Variables**: **PCA** doesn't improve accuracy according to the paper.
- **Enhancing the ideas in the paper, for P-variables:**
  - 'zeros' - count of the columns which have the value 0, among the columns with this zero problem.
  - **One-hot encoding** (taking categorical).
- **Additional Ideas:** scaling all the numerical attributes between 0 and 1 (**normalization**) - important for models like SVR, Ridge, Lasso.

P-Variable treatment	**Submission RMSE**
Taking them as continuous	1821659
PCA Processing with 20 Principal Components	1934188
Taking them as categorical	1749479
Adding 'Zeros' feature	**1719022**

Additional Preprocessing	**Submission RMSE**
Without normalization of numerical attributes	1780995
After normalization of numerical attributes	**1719022**

City Group Categorical Variable



Models Used

- Random Forest
- Support Vector machine with regression
- Ridge
- Lasso
- Ensemble models

Random Forest

A random forest is an **ensemble of decision trees** created using **random variable selection and bootstrap aggregating** (bagging).

Parameters: n_iterators, criteria of split

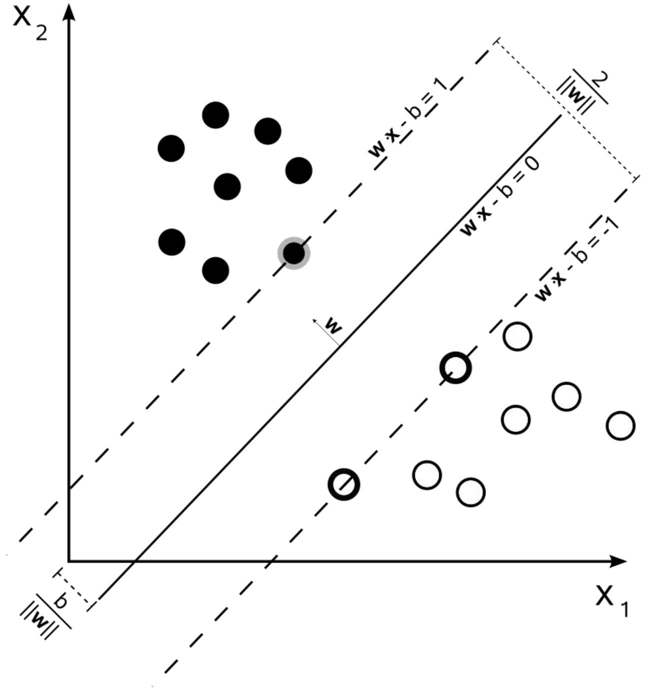
- Different regression trees can be run in **parallel**.
- Individual trees **tend to grow deep, learn highly irregular patterns: they overfit their training sets, i.e. have low bias, but very high variance.**
- Bias–variance tradeoff: Randomization reduces variance.
- Not sensitive to outliers.
- Ability to **rank variable importance.**

Support Vector Regression

An SVM training algorithm builds a model that assigns new examples to one category or the other, making it a non-probabilistic binary linear classifier.
Here we use multiclass SVR.

1.Produce very accurate classifiers but computationally expensive for large data.
2.Less overfitting, robust to noise.

Parameters:
1.Penalty (If you have a lot of noisy observations you should decrease it. It corresponds to regularize more the estimation.)
2. type of kernel used (rbf, poly, linear)



Ridge & Lasso Model

- **Ridge Regression:**
  - Performs L2 regularization, i.e. adds penalty equivalent to **square of the magnitude** of coefficients
  - Minimization objective = LS Obj + α * (sum of square of coefficients)
- **Lasso Regression:**
  - Performs L1 regularization, i.e. adds penalty equivalent to **absolute value of the magnitude** of coefficients
  - Minimization objective = LS Obj + α * (sum of absolute value of coefficients)

Parameters: learning rate, initial coefficients

Ensemble Models

- Goal is to combine the predictions of several base estimators to improve generalizability / robustness over a single estimator.
- Ensembling reduces variance and bias.
- **Used 0.5*rf + 0.5*svr**
- Enhancing the ensembles in the paper:
    - **0.5*ridge + 0.3*rf + 0.2*svr**
    - **0.5*ridge + 0.5*lasso**
    - **0.3*results_ridge + 0.3*results_lasso + 0.25*results_rf + 0.15*results_svm**

Results

Model	**Submission RMSE**
Random Forest	1895861
SVR	1909604
Ensemble of Random Forest and SVR	1782767
Ridge	1749468
Ensemble of Random Forest, Ridge and SVR	**1719022**
Lasso	1775850
Ensemble of Lasso and Ridge	1744109
Ensemble of Lasso, Ridge, SVR and Random Forest	1725979

Performance metrics

Since we **do not have the ground truth for the validation set**, we mainly tested out performance **by submitting our code on kaggle** to see what rank we got. This also gave us the **RMSE** values for the dataset and hence we used this as a **parameter for minimisation.**

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

Where $\hat{y}_i$ is the predicted revenue of the ith restaurant and y_i is the actual revenue of the ith restaurant.

Scope

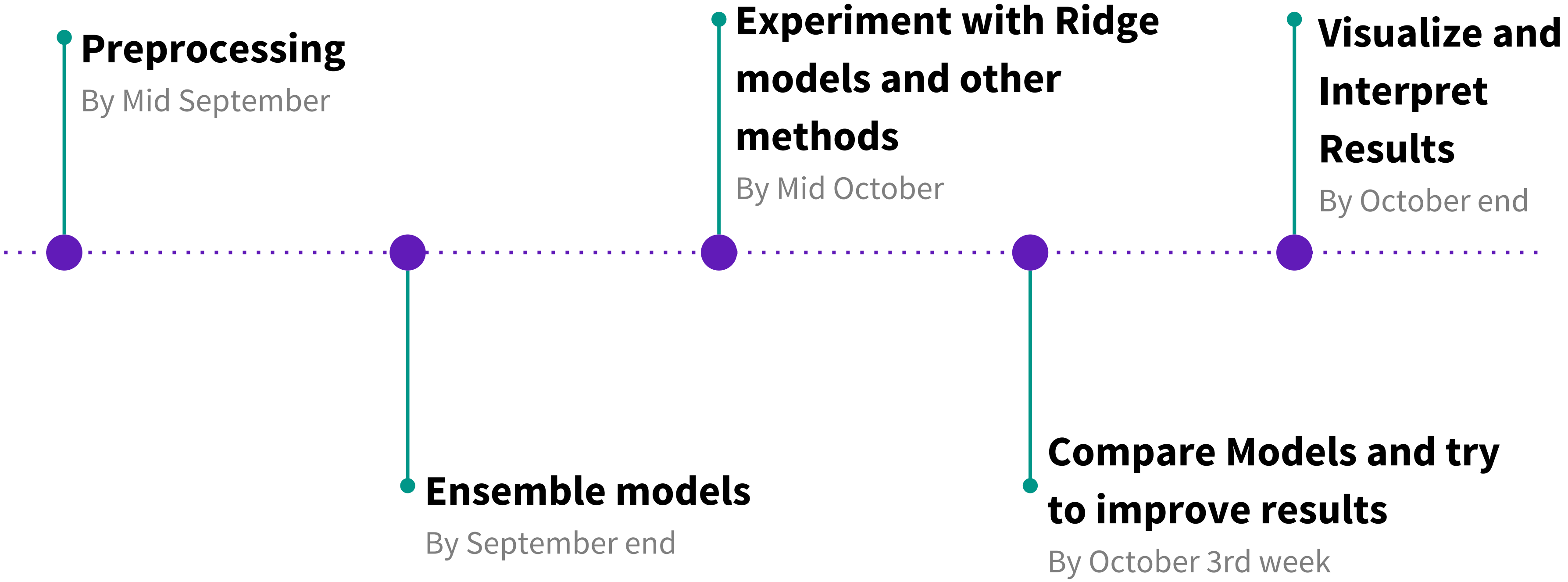
- The solution is **only applicable to Turkish restaurants** and locations on which the data is based. A different location based data may need a different kind of ensemble for accuracy.
- It is limited to only **annual and not seasonal** revenue analysis.
- The accuracy is **limited by the models**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

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.

- **Ensembling** helps to **average out the error** due to different models and noise in different models. An ensemble of Random Forest, SVR and Ridge gave us the best results among all our attempts, securing **Rank 1 on Kaggle Leaderboard** and an **RMSE value of 1719022.**

Timeline



Preprocessing

By Mid September

Ensemble models

By September end

Experiment with Ridge models and other methods

By Mid October

Compare Models and try to improve results

By October 3rd week

Visualize and Interpret Results

By October end

References

- https://en.wikipedia.org/wiki/Tikhonov_regularization
- http://www.saedsayad.com/support_vector_machine_reg.htm
- http://kpei.me/blog/wp-content/uploads/2015/05/TFIKaggleReport.pdf
- http://stats.stackexchange.com/questions/17251/what-is-the-lasso-in-regression-analysis
- http://statweb.stanford.edu/~tibs/lasso/simple.html