Parking Costs Survey

Final Report

Shanhai Liao, Zuoqi Zhang, Tianshu Qu

1. Project Description

There are many parking lots across Boston region. The parking costs may differ from regions to regions and the costs may change over time. The purpose of this project is to collect the parking costs in various Boston regions and see how the price changes on the time scale. And the ultimate goal of our project is to figure out the reasons behind these changes over time and difference across various regions. What components are driving up the parking costs in the last few years? For example, whether the parking prices will go up around a big event such as a Red Sox game or Christmas festival? What factors are making the parking costs differ from places to places? What algorithms should we utilize to do the analysis? And what methods can we use to evaluate our algorithms?

At first, we collect Boston parking prices from ParkWhiz.com via its API. The dataset contains the daily parking prices of last ten years. Unexpectedly, the parking costs just change from weekday to weekends regularly but almost do not change daily during the ten years. We try different time step selections to see whether it can make a difference such as 12:00AM-12:00PM and 9:00AM-5:00PM although this does not help. Then we select other data sources to extract the parking prices information. But they do not provide any API or past parking prices information. Therefore, we write a python script to scrape the website. And the data is exactly the same as what we get previously. The negative dataset is disappointing and the time component does not make any sense, which makes us mainly focus on the geographic information.

Then we apply K-means algorithm on our dataset based on the coordinates and parking prices of these parking lots. We find the result makes sense when K equals 5. The clustering results show the price rank across Boston region. After that, we use Google Places API to collect the number of places in different types, just like restaurant, stadium and bar. According to the dataset we get, we redefine our second problem as how does the number of places in different types nearby a parking lot determine its parking cost? And what factors are most significant? After cleaning the dataset, we choose Linear Regression as a base model. We split the dataset into two parts, the 90% training set and the 10% testing set to do the Cross Validation. We use Mean Squared Error between the true value and predicted value to estimate our algorithm. Apart from Linear Regression model, we apply regularized quadratic model, Neural Network model and K Nearest Neighbors models into our dataset. The regularized polynomial algorithm performs best and the MSE decreases by 90% from LR model.

2. Data Description

In order to do the clustering based on the parking prices and coordinates and predict how does the number of specific types of places affects the parking costs, we collect 2 datasets, the parking lot information and the nearby places' information. We will describe how we collect these data and what our data is.

2.1 Parking Lot Information

We obtain the parking lot from ParkWhiz.com via its API. The API needs five parameters: API key, coordinate , radius, begin_time and end_time. We use a point in Back Bay area, [42.358056,-71.063611] as the centroid , Jan 1st 2008 as the start time, Apr 1st 2018 as the end time and let the radius be 10 miles so that the API would return the parking lots information within the circle from Jan 1st 2008 to Apr 1st 2018. The returned results are in JSON format containing the parking lot information in the last ten years. And it's about 1.2GB in total. We only extract the parking lot address, coordinates and its hourly parking prices. For these Null values in the parking price column, we set them to 0. And after observing the dataset, we find that the price does not change from hour to hour. So we decide to just keep the daily price. After cleaning the data, there are 55 available parking lots in Boston and we store them in a 55*3713 table, the size of the dataset is 1MB. The dataset has the following attributes:

Address — String

Date — String

Price — float

2.2 Neighbors Information

We obtain the nearby places information from Google Places API. The API needs three parameters: API key, coordinate, radius. We use the coordinates of each parking lot which we collect in dataset parking lot information to search the nearby places around it, where the radius equals five hundred meters. The API returns at most 60 nearby places around one site. Each place nearby the parking site can have many types, such as restaurant, park and bar. The dataset is in JSON format and is about 1MB in total. There are 92 categories of places. We count the number of each type of places nearby a specific parking lot and store it in a 55*92 table. In the data analysis part, we use the number of each type of places nearby a parking site as the feature vector to do some regression. But all algorithms do not have a good result. So we drop some types, most of whose values are zero. And there are 43 types left finally. After cleaning the data, the size of dataset is 21KB. The dataset has the following attributes:

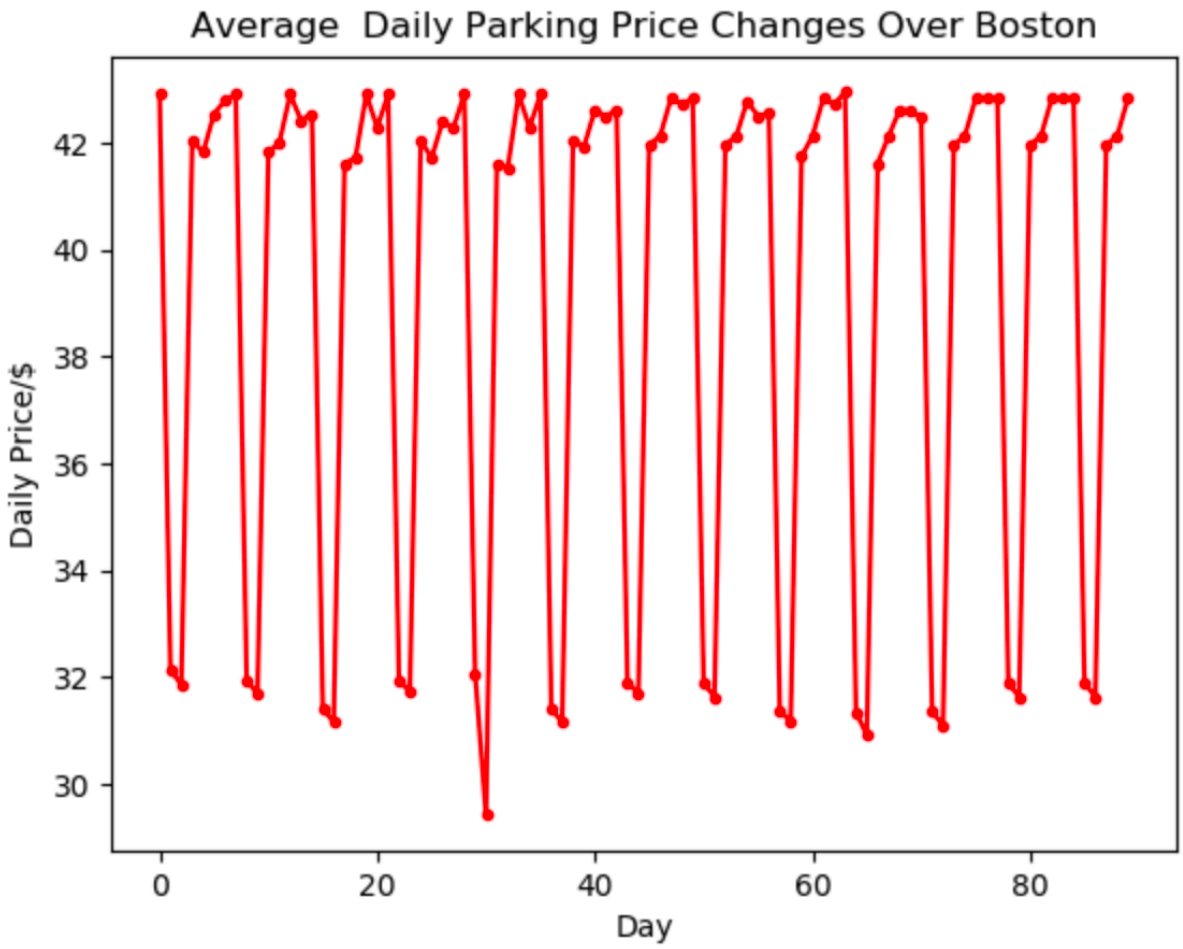
Address — String

Number of Each Types — Indicator Vector

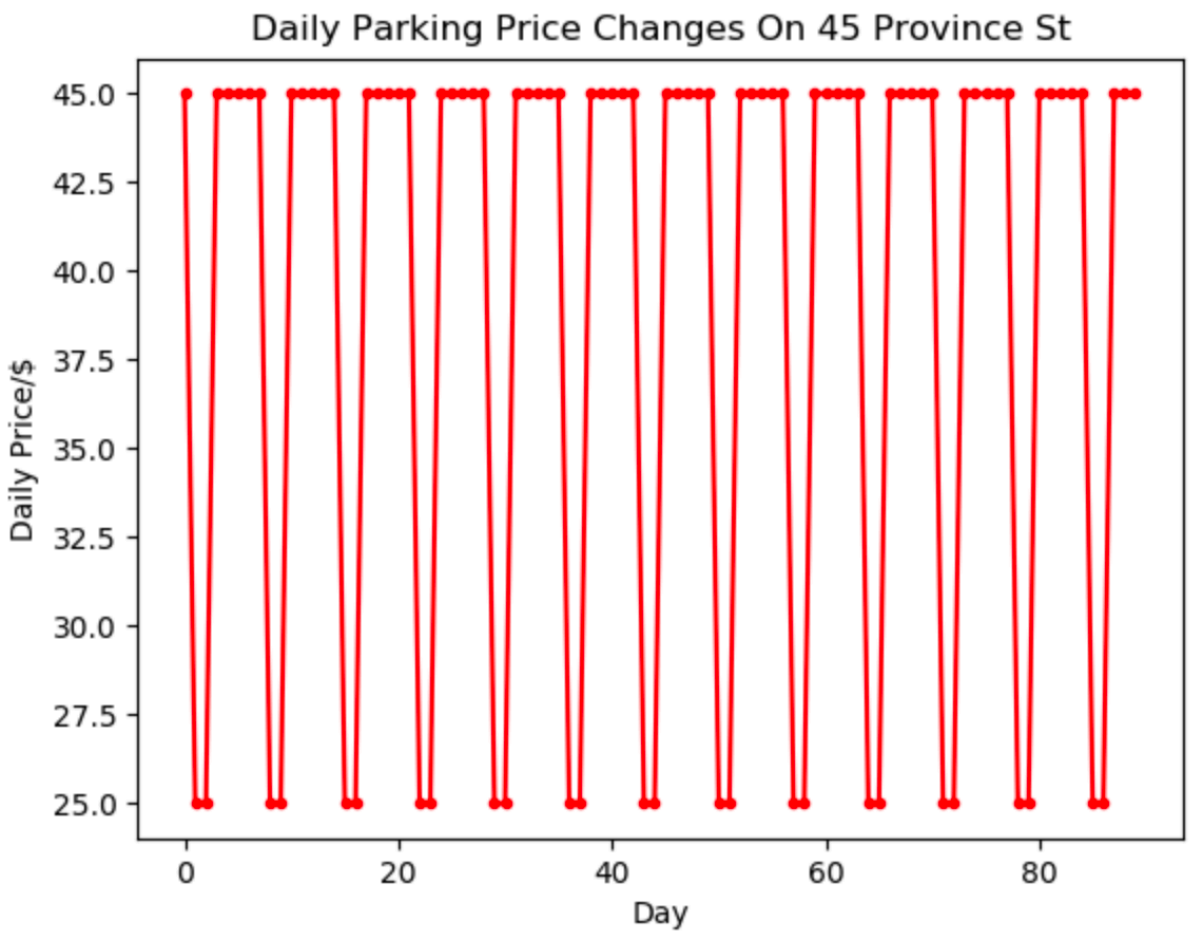
3. Data Analysis

3.1 Data Change Plot

Firstly, we try to get something interesting from the Parking Lot Information dataset. We get a set of data in the period between Dec 1st, 2017 to Mar 1st, 2018 as an example to show the plots as the graph below.



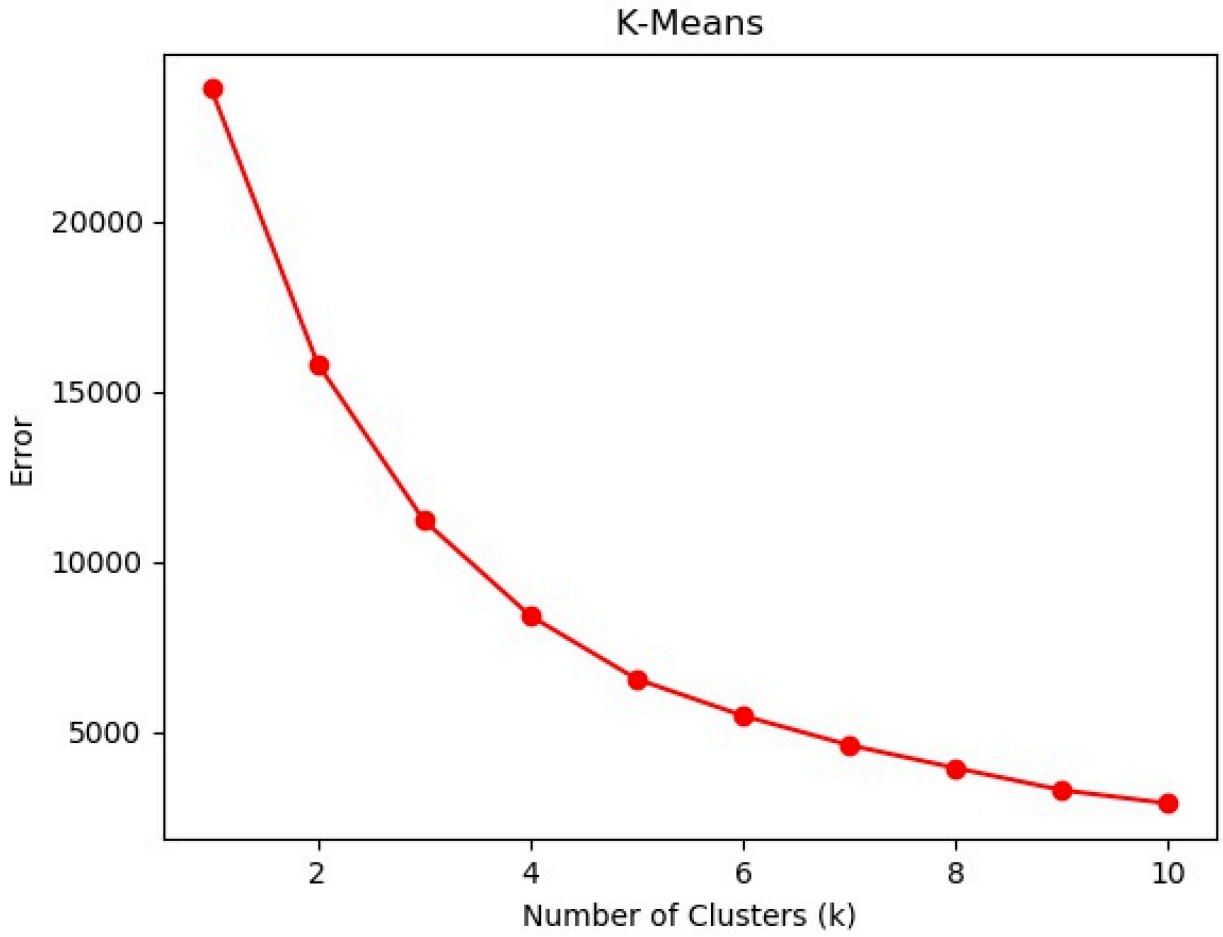
The value of daily price is an average daily rate of all parking lots in Boston region. We can see from the plot that the average daily prices rarely change from day to day except on weekends.The occasional absent parking prices from some sites make the periodical function look imperfect. Then we choose an address "45 Province St" to observe the change of its daily rate. The results are shown below.



The figure shows that the daily price just changes periodically from weekday to weekends.

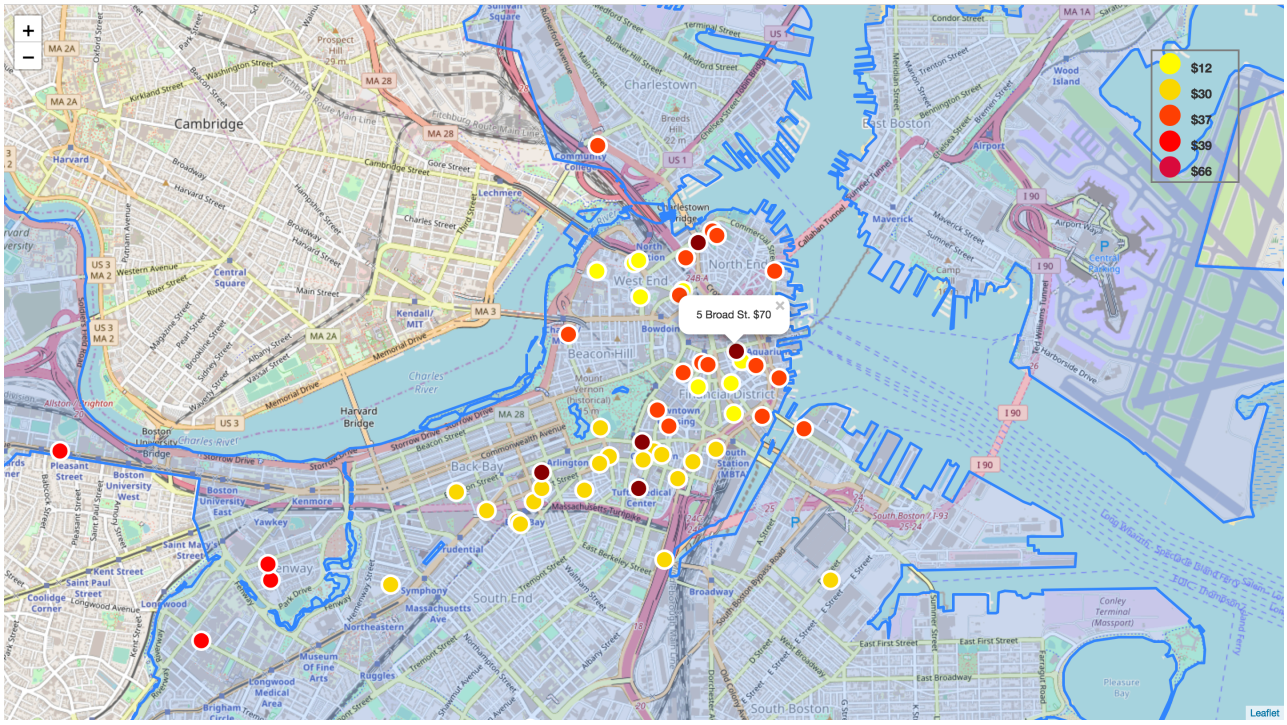
3.2 Clustering on Dataset

Then we mainly focus on the geographic information. We try to utilize K-means algorithm to categorize these parking lots based on their daily rates and coordinates. Note that, in this part, the spatial coordinates and restaurant categories have different units of scale. To make our results not skewed, we scale the latitude and longitude at a range [0,100] and after clustering we rescale the coordinates into the original units to show it on a geographic map. One of the challenges in this part is how to determine the number of clusters. The method here is to plot a function between the total error and the number of K.



By observing the K-means plot, when k equals 5 the error curve declines sharply, but after k equals 6 the slope of the curve gets much slower, where there is no need to add more clusters for the dataset. Therefore, the number of clusters should be 5.

The clustering results are shown below.



3.3 Regression on Base Model

Next, we use Linear Regression as the base model. We calculate an average daily price of a parking lot of the last ten years as its daily price. According to what we have shown in the 3.1 Data Change Plot section, we know the average daily rating can represent the current parking price of a parking lot. The regression summary is shown below.

OLS Regression Results

Dep. Variable:	y	R-squared:	0.867
Model:	OLS	Adj. R-squared:	0.426
Method:	Least Squares	F-statistic:	1.964
Date:	Sun, 29 Apr 2018	Prob (F-statistic):	0.104
Time:	21:04:04	Log-Likelihood:	-166.96
No. Observations:	53	AIC:	415.9
Df Residuals:	12	BIC:	496.7
Df Model:	40		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	-53.7338	53.382	-1.007	0.334	-170.042	62.575
x1	32.1701	14.935	2.154	0.052	-0.371	64.711
x2	-8.4504	9.481	-0.891	0.390	-29.108	12.207
x3	13.8498	8.219	1.685	0.118	-4.059	31.758
x4	18.0639	7.650	2.361	0.036	1.395	34.733
x5	3.6211	3.766	0.961	0.355	-4.585	11.827
x6	4.3940	3.109	1.413	0.183	-2.380	11.168
x7	-14.8589	5.150	-2.885	0.014	-26.079	-3.638
x8	28.4284	10.187	2.791	0.016	6.232	50.625
x9	23.5172	11.336	2.074	0.060	-1.183	48.217
x10	4.2833	5.289	0.810	0.434	-7.240	15.806
x11	-47.8275	18.879	-2.533	0.026	-88.960	-6.695
x12	15.8554	11.866	1.336	0.206	-9.999	41.710
x13	-9.5857	7.867	-1.218	0.246	-26.727	7.555
x14	1.9300	0.911	2.120	0.056	-0.054	3.914
x15	1.4712	9.990	0.147	0.885	-20.296	23.238
x16	14.3662	16.255	0.884	0.394	-21.050	49.782
x17	-2.2169	4.012	-0.553	0.591	-10.959	6.525
x18	8.1556	12.123	0.673	0.514	-18.258	34.569
x19	-2.8515	9.450	-0.302	0.768	-23.441	17.738
x20	-10.4562	13.662	-0.765	0.459	-40.223	19.310
x21	43.9236	17.099	2.569	0.025	6.667	81.180
x22	-7.8694	4.174	-1.885	0.084	-16.964	1.225
x23	0.9269	2.179	0.425	0.678	-3.821	5.675
x24	-2.6254	2.904	-0.904	0.384	-8.952	3.702
x25	-50.6062	19.673	-2.572	0.024	-93.470	-7.742
x26	16.6759	10.159	1.642	0.127	-5.458	38.810
x27	48.7234	15.521	3.139	0.009	14.906	82.540
x28	8.1362	9.411	0.865	0.404	-12.369	28.642
x29	1.7326	7.173	0.242	0.813	-13.896	17.361
x30	20.3070	11.967	1.697	0.115	-5.766	46.380
x31	-16.1501	6.701	-2.410	0.033	-30.751	-1.549
x32	14.1085	7.643	1.846	0.090	-2.544	30.761
x33	0.0838	1.477	0.057	0.956	-3.134	3.302
x34	7.4935	9.606	0.780	0.450	-13.437	28.424
x35	-31.9069	12.570	-2.538	0.026	-59.294	-4.519
x36	32.6488	10.330	3.161	0.008	10.141	55.156
x37	62.0938	22.299	2.785	0.017	13.509	110.679
x38	-3.6979	2.674	-1.383	0.192	-9.524	2.128
x39	4.3940	3.109	1.413	0.183	-2.380	11.168
x40	4.3940	3.109	1.413	0.183	-2.380	11.168
x41	4.3940	3.109	1.413	0.183	-2.380	11.168
x42	1.0973	7.412	0.148	0.885	-15.053	17.247
x43	8.8760	6.725	1.320	0.212	-5.777	23.529

Omnibus:	3.516	Durbin-Watson:	1.778
Prob(Omnibus):	0.172	Jarque-Bera (JB):	3.090
Skew:	-0.181	Prob(JB):	0.213
Kurtosis:	4.126	Cond. No.	1.03e+16

The dependent variable here denotes the parking price of a parking site. And X1~X43 represents the number of these different types of places nearby the parking lot. There are 43 types in total. By observing the summary, the P value of X4, X7, X8, X25, X27, X31, X35, X36, X37 are less than 0.05, which indicates these independent variables are the most significant factors in the regression model. They represent bank, cafe, car_rental, meal_delivery, movie_theater, parking, shoe_store,

spa and stadium. What's more, from the coefficient of these independent variables, it's clear that as the number of cafe, meal_delivery, parking, shoe_store nearby a parking lot increases, the daily rate of a parking lot will go down. On the other hand, as the number of bank, car_rental, movie_theater, spa and stadium nearby a parking lot increases, the daily rate of a parking lot will go up.

4. Algorithms

We want to get a quantitative relation between the parking price and the number of different types of places around it. We have utilized four regression algorithms to find which model fits our dataset best and has the best results. In this section, we will describe what these algorithms are and how to address them into our problem.

4.1 Linear Regression

Linear regression is a good basic regression model and we make an assumption that our models depend linearly on the unknown parameters. We apply R squared and Mean Squared Error (MSE) to evaluate this algorithm. The R squared is 0.867 and MSE is 1732. Obviously, the results are not satisfying and we improve it in the Polynomial Regression part.

4.2 Polynomial Regression with Lasso Regularization

We select eight most significant elements according to 3.3 Regression on base model and add eight quadratic term on the base model. What's more, to avoid overfitting, we apply Lasso Regularization on our model with the punishment parameter being 10. The R squared is 0.971 and MSE is 175, which improves a lot from the base model.

4.3 Neural Network

We create a three-layer neural network with two hidden layers and use gradient descent algorithm, back propagation algorithm to update our unknown parameters. After thousands of iterations, our algorithm converges. The R squared is 0.610 and the MSE is 599. In order to make some improvements, we try different learning rates of the neural network. Because the algorithm may skip the global optimization point in the gradient descent process, we begin the learning rate from a very small value. After several try, we find 0.001 has the best performance. And the R squared is 0.64 and the MSE is 256.

4.4 K Nearest Neighbors

We use Euclidean distance to measure the distance between the different data samples. For each testing data sample, we find the first K neighbors of it and use the mean value of the first K neighbors as the predicting value. After trying different K, we find the algorithm has the best performance. The R squared is 0.47 and the MSE is 170.

5. Experimental Results

We use Mean Squared Error and R squared to evaluate the performance of regression models. We use 90% data as the training dataset and 10% data as the testing data to do the cross validation. For each test, we shuffle the dataset and compute the average MSE and R squared. In our experiments, we shuffle the dataset for five times.

Algorithm	Mean Squared Error	R squared
Linear Regression	1732.484	0.867
Regularized Polynomial Regression	175.871	0.971
Neural Network	256.438	0.64
K Nearest Neighbors	169.199	0.47

The R squared is defined as (1 - u/v), where u is the residual sum of squares ((y_true - y_pred) ** 2).sum() and v is the total sum of squares ((y_true - y_true.mean()) ** 2).sum(). The best possible score is 1.0 and it can be negative (because the model can be arbitrarily worse). A constant model that always predicts the expected value of y, disregarding the input features, would get a R^2 score of 0.0. (We get this from scikit-learn)

The definition of MSE is shown below.

If $\hat{Y}$ is a vector of n predictions, and Y is the vector of observed values of the variable being predicted, then the within-sample MSE of the predictor is computed as

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2.$$

I.e., the MSE is the *mean* $\left(\frac{1}{n}\sum_{i=1}^{n}\right)$ of the *squares of the errors* $(Y_i - \hat{Y}_i)^2$. This is an easily computable quantity for a particular sample (and hence is sample-dependent).

(We get this from Wikipedia)

6. Conclusions

6.1 What components are driving up the parking costs in the last few years?

In the past ten years, the parking costs just change from weekday to weekends regularly but almost do not change from day to day even in a big event or extreme weather.

6.2 What factors are making the parking costs differ from places to places?

The number of different types of places around a parking site affect its parking prices. There are 43 independent variables in total. Especially eight variables of them are most significant. They are bank, cafe, car_rental, meal_delivery, movie_theater, parking, shoe_store, spa and stadium. As the number of cafe, meal_delivery, parking, store_shoe nearby a parking lot increases, the daily rate of a parking lot will go down. On the other hand, as the number of bank, car_rental, movie_theater, spa and stadium nearby a parking lot increases, the daily rate of a parking lot will go up.

6.3 What algorithms should we utilize to do the analysis?

We have applied Linear Regression model, Regularized Polynomial Regression model, Neural Network model and K Nearest Neighbors models into our dataset. And the Regularized Polynomial Regression stands out.

6.4 What methods can we use to evaluate our algorithms?

We use R squared and Mean Squared Error to evaluate our algorithms.

7. Resources

7.1 Work Repository

All dataset, code, graph and plot can be found on our GitHub.

7.2 Data Source

We obtain the parking lot form ParkWhiz.com via its API.

We obtain the nearby places information from Google Places API.

7.3 Algorithm Implementation

Our code is written in Python. The machine learning algorithm is implemented by python library statsmodels and sklearn.