Capstone Project

-

NYC restaurant inspection result

General Assembly DSI (NYC)
Chris Lee

February 2019

Problem Statement

New York City is the biggest city on the east coast of the United States and there are tons of different kinds of foods and restaurants in the melting pot.

With the huge food market, NYC restaurant inspection is one of the most important issue since there are various ways to cook, store, and serve different kinds of food.

I resolved to tap into this wealth of information, and use it to create a tool to provide NYC restaurant inspectors by applying statistical models with some recommendations.

Agenda

1. Data Collecting
2. Exploratory Data Analysis (EDA)
3. Preprocessing
4. Modeling & Evaluation
5. Conclusion & Recommendation

1. Data Collecting (dataset)

Which neighborhoods/area to identify?

- NYC 5 boroughs (Manhattan, Queens, Brooklyn, Bronx, Staten Island)

DOHMH NYC Inspection Results NYC OpenData

What does this NYC restaurant inspection result data include?

- CAMIS
- DBA
- Boro
- Address
- Phone

- Violation Code
- Violation Description
- Score
- Grade
- Grade date

- Inspection Date
- Inspection Type
- etc.

About 350,000 observation with 23,000 unique restaurants / Period : 2015 ~ 2019

1. Data Collecting (dataset)

NEW YORK STATE OF OPPORTUNITY. | Office of Information Technology Services

- **ZIP code dataset**
  - Extract a list of NYC zip codes to use it as a search term in Yelp API
  - Bronx - 005 / Kings (Brooklyn) - 047 / New York (Manhattan) - 061 / Queens - 081 / Richmond (Staten Island) - 085

yelp

- **Yelp API** (search term = list of zip codes)

- What does this NYC restaurant inspection result data include?

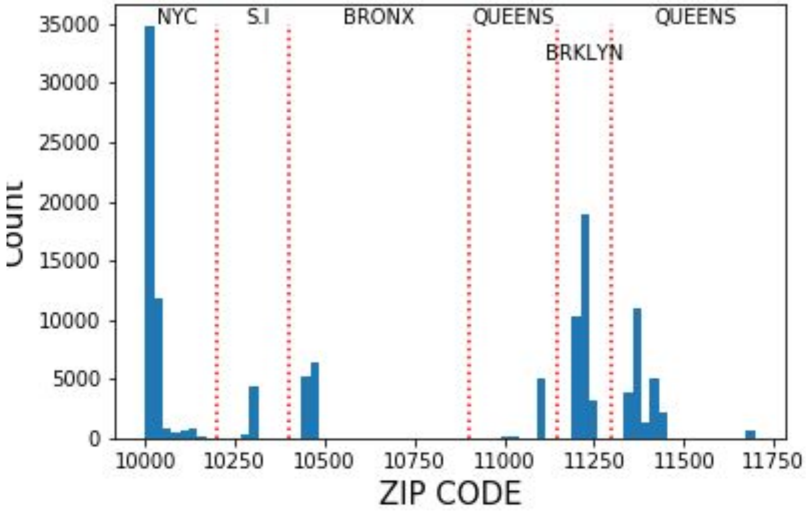
- name
- address
- zipcode
- rating
- price
- cuisine
- phone

- Scraped about 6,000 unique restaurants

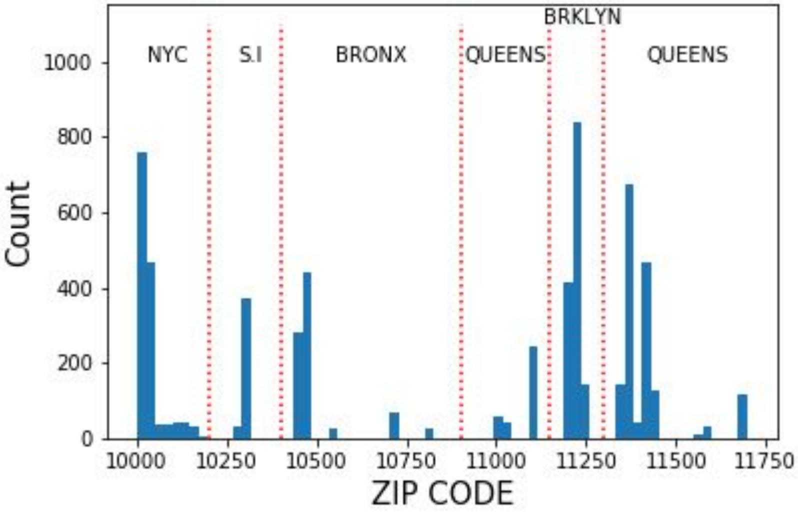
2. Exploratory Data Analysis (EDA)

- **Cleaned data has 7,707 inspection results with 1388 restaurants**

ZIP code distribution (original)

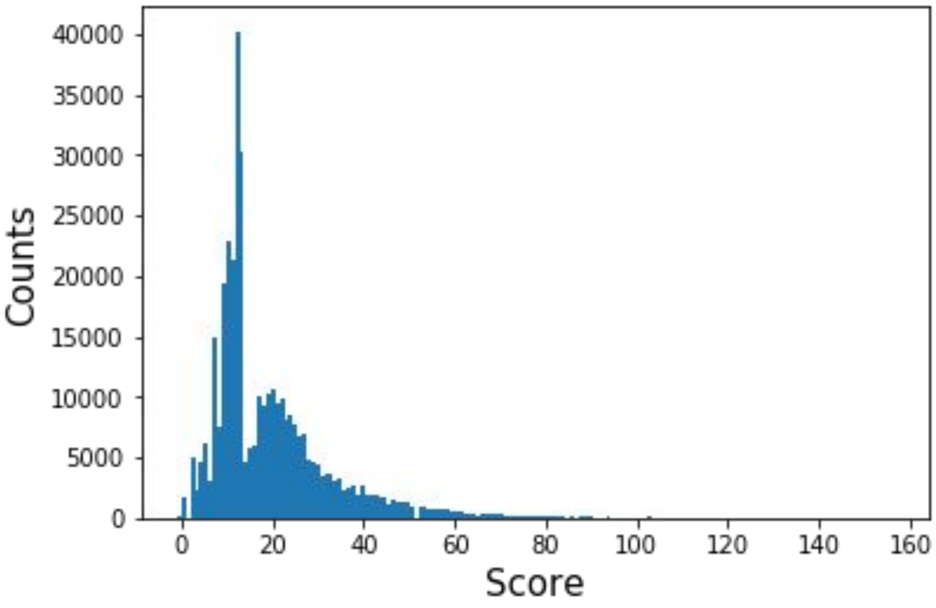


ZIP code distribution (after combining)

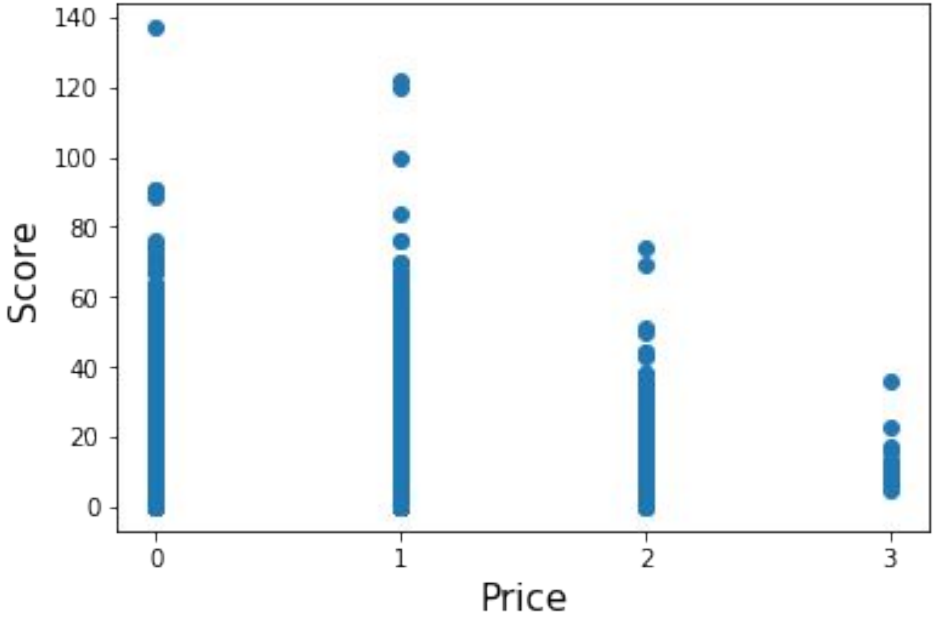


2. Exploratory Data Analysis (EDA)

Inspection score distribution



Price vs Inspection score



2. EDA - Interesting fact

- McDonald's average inspection score **12.2**
- Burger King average inspection score **14.9**
- Wendy's average inspection score **11.3**

3. Preprocessing

Difficulties matching **NYC restaurant inspection result** and **Yelp data**

1. Restaurant name is different
2. Address format is different
3. Phone number is different
4. and more ...

example)

NYC restaurant inspection result			Yelp API		
name	address	phone	name	address	phone
carvel	36-10A 47th ave	7180000000	carvel ice cream	3610 47 avenue	+17180000000
chris sushi II	237 w 7th street	6460000000	chris II	237 west 7th st	9170000000

3. Preprocessing

Features (numerized)

- Borough - one-hot encoded
- inspection date - month and year
- inspection type - initial inspection or not
- rating - 1.0 ~ 5.0
- price - $$ to number
- Cuisine - one-hot encoded

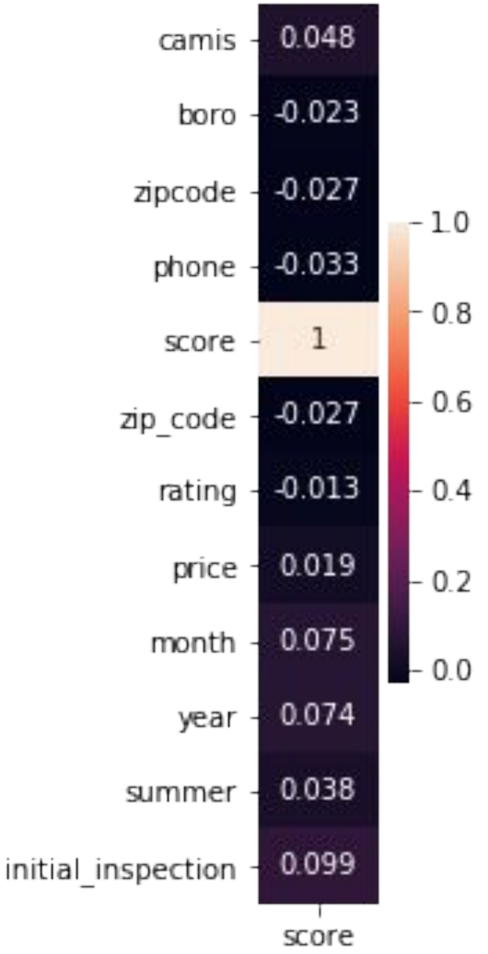
Target

- Inspection score

Grade A = 0 < score < 13

Grade B = 14 < score < 27

Grade C = 28 < score



4. Modeling & Evaluation

Regression Models - predict the inspection score

- ***Linear***
  - Linear regression
  - Lasso
  - Ridge
  - Elastic net

- ***tree-based***
  - Decision Tree
  - Bagged Tree
  - Random Forest
  - Extra Tree
  - Ada Boost
  - Gradient Boost

- ***Neural Network***
  - Neural Network

4. Modeling & Evaluation

Classification Models - predict if the inspection score is within certain range

- **Option A)** Class 0 - inspection score 0 ~ 13
  Class 1 - inspection score 14 ~ 27
  Class 2 - inspection score 28+

- **Option B)** Class 0 - inspection score 0 ~ 13
  Class 1 - inspection score 14+

- **Option C)** Class 0 - inspection score 0 ~ 27
  Class 1 - inspection score 28+

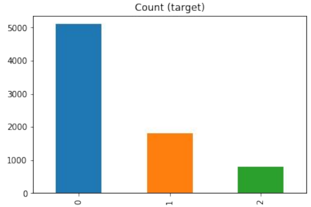
4. Modeling & Evaluation

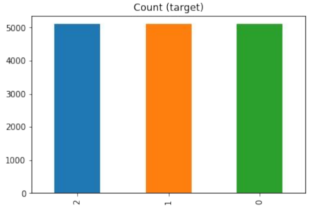
Classification Models - predict if the inspection score is within certain range

- ***Linear***
  - Logistic regression
- ***tree-based***
  - Decision Tree
  - Bagged Tree
  - Random Forest
  - Extra Tree
  - Ada Boost
- ***Neural Network***
  - Neural Network

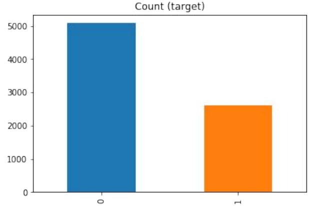
4. Modeling & Evaluation

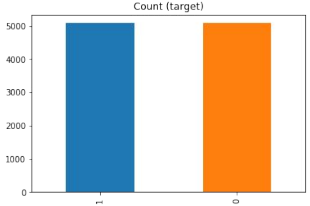
Imbalanced class (over-sampling)



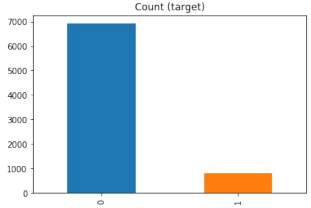


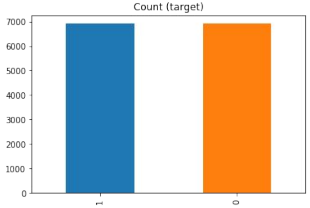
Option A) baseline accuracy - 66.17%





Option B) baseline accuracy - 66.13%





Option C) baseline accuracy - 89.61%

4. Modeling & Evaluation

Option A) multi-class classification

	model	original			polynomial feature + pca		
	model	cross-val_x	train_x	test_x	cross-val_y	train_y	test_y
0	RandomForestClassifier(bootstrap=True, class_w...	0.610898	0.889792	0.601453	0.620245	0.898097	0.598858
1	ExtraTreeClassifier(class_weight=None, criteri...	0.566783	0.908478	0.574987	0.541181	0.908478	0.550597
2	DecisionTreeClassifier(class_weight=None, crit...	0.573182	0.908478	0.562013	0.542047	0.908478	0.542813
3	BaggingClassifier(base_estimator=None, bootstr...	0.616094	0.887370	0.602491	0.611423	0.894637	0.578620
4	LogisticRegression(C=1.0, class_weight=None, d...	0.652250	0.668858	0.655423	0.632873	0.719377	0.630514
5	KNeighborsClassifier(algorithm='auto', leaf_si...	0.619893	0.714014	0.609237	0.625776	0.715398	0.618578

4. Modeling & Evaluation

Option B) binary classification

		original			polynomial feature + pca		
	model	**cross-val_x**	**train_x**	**test_x**	**cross-val_y**	**train_y**	**test_y**
0	RandomForestClassifier(bootstrap=True, class_w...	0.746390	0.963639	0.792520	0.780650	0.961801	0.834252
1	ExtraTreeClassifier(class_weight=None, criteri...	0.714621	0.969808	0.770866	0.723810	0.969808	0.789370
2	DecisionTreeClassifier(class_weight=None, crit...	0.721186	0.969808	0.775197	0.738248	0.969808	0.780709
3	BaggingClassifier(base_estimator=None, bootstr...	0.742711	0.960226	0.775984	0.790759	0.962457	0.848819
4	LogisticRegression(C=1.0, class_weight=None, d...	0.628774	0.674455	0.630315	0.706878	0.870832	0.730315
5	KNeighborsClassifier(algorithm='auto', leaf_si...	0.611445	0.766868	0.635433	0.632972	0.772381	0.644094

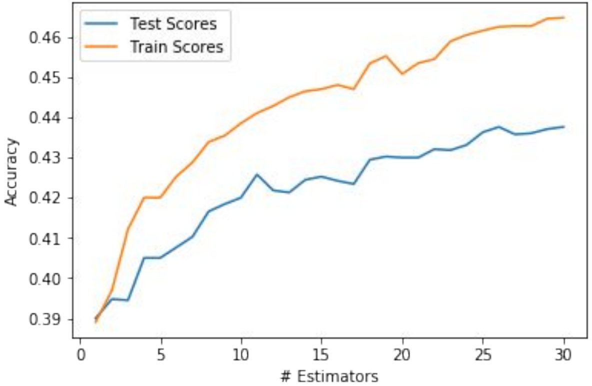
4. Modeling & Evaluation

Option C) binary classification

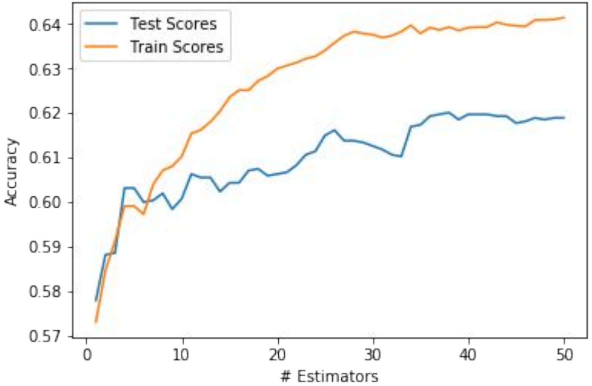
		original			polynomial feature + pca		
	model	cross-val_x	train_x	test_x	cross-val_y	train_y	test_y
0	RandomForestClassifier(bootstrap=True, class_w...	0.928107	0.984207	0.956699	0.971418	0.983916	0.982854
1	ExtraTreeClassifier(class_weight=None, criteri...	0.893906	0.984498	0.927928	0.942157	0.984498	0.966579
2	DecisionTreeClassifier(class_weight=None, crit...	0.890514	0.984498	0.929090	0.946517	0.984498	0.967742
3	BaggingClassifier(base_estimator=None, bootstr...	0.903595	0.982560	0.931706	0.972193	0.984207	0.983435
4	LogisticRegression(C=1.0, class_weight=None, d...	0.676291	0.705455	0.673641	0.855537	0.932177	0.881720
5	KNeighborsClassifier(algorithm='auto', leaf_si...	0.760875	0.877047	0.820110	0.758163	0.879760	0.826795

4. Modeling & Evaluation

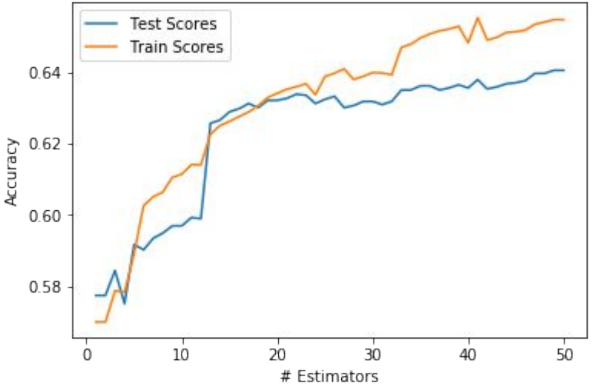
Ada Boost



option A



option B

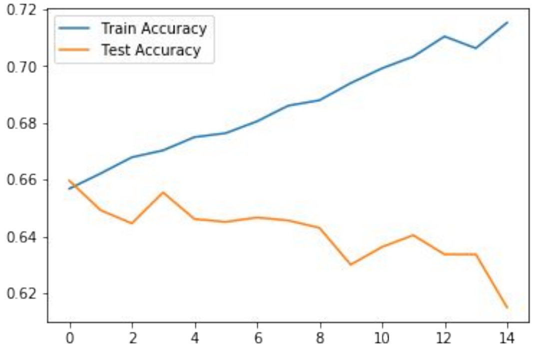


option C

4. Modeling & Evaluation

Neural Network

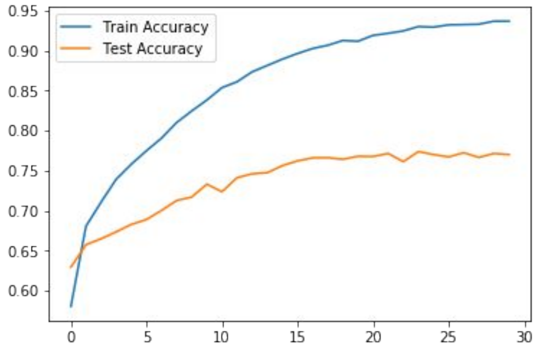
Accuracy



epochs

option A

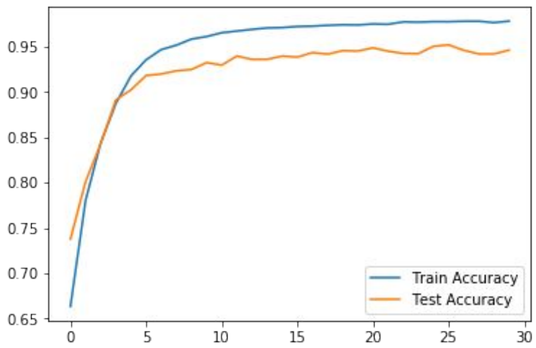
Accuracy



epochs

option B

Accuracy



epochs

option C

4. Modeling & Evaluation

Confusion matrix of tree based models:

Decision Tree

	Pred Neg	Pred Pos
Act Neg	6229	677
Act Pos	77	6829

Random Forest

	Pred Neg	Pred Pos
Act Neg	6293	613
Act Pos	59	6847

Bagged Tree

	Pred Neg	Pred Pos
Act Neg	6212	694
Act Pos	81	6825

Extra Tree

	Pred Neg	Pred Pos
Act Neg	6227	679
Act Pos	77	6829

4. Modeling & Evaluation

Random Forest
Feature importance

	feat
rating	0.177868
price	0.063179
summer	0.037286
Brooklyn = **2**	0.026463
Manhattan = **0**	0.026441
Queens = **3**	0.026191
Cycle Inspection / Re-inspection	0.022963

...

+

Logistic regression Coefficient

Feature	coef	exp(coef)
0 = Manhattan	0.2162	1.24
1 = Bronx	0.0036	1.00
2 = Brooklyn	0.0693	1.07
3 = Queens	-0.0686	0.93
4 = Staten Island	-0.0781	0.92
rating	-0.2713	0.76
price	0.0197	1.02
summer	0.4227	1.53
...	...	...
Cycle Inspection...	-0.2552	0.77
pizza	0.1774	1.19
mexican	0.1924	1.21
coffee	-0.2314	0.79
chinese	0.3018	1.35
hotdogs	-0.2570	0.77

Conclusion & Recommendation

- If a restaurant have one-unit higher rating, the chance to get inspection grade of C decrease by 23%.
- During the summer season, the probability to get 28 inspection score or more increase by 50%.
- Manhattan area tends to have poor inspection score.
- A restaurant being re-inspected would have lower chance to get the inspection grade of C.

Next step...

- If we could match each restaurants name in inspection data and yelp data, we may be able to build stronger and more accurate model with better accuracy.
- If we could collect more data such as size of the restaurant, number of visitors, etc.

Q & A