

When is your next inspection?

Springboard Capstone Project

by Jonas Makonnen

# Problem and Client

- ▶ Problem: Prediction of the number of days until next restaurant inspection
- ▶ Potential Clients: Restaurant owners, that could use predictions to prepare for an upcoming inspection ahead of time, and thereby potentially increase their chances to receive a good grade.

## Data

- ▶ Open Data on restaurant inspections carried out by the Department of Mental Health and Hygiene (DOHMH) in New York City
- ▶ Information about inspection: date, name of the restaurant, cuisine type, violations, grade, actions taken by the DOHMH, etc.

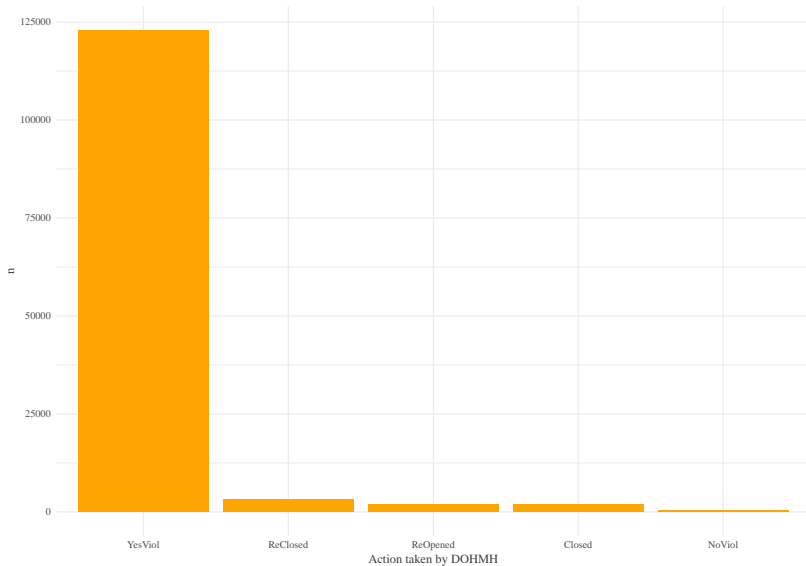
```
## # A tibble: 8 x 3
##   Var          Var          Var
##   <chr>        <chr>        <chr>
## 1 id          inspection_date record_date
## 2 rest_name   action          violation_group
## 3 boro        violation_code  viol_vermin
## 4 building    violation_descr viol_facility
## 5 street      critical_flag   viol_food
## 6 zipcode     score          viol_hygiene
## 7 phone       grade          viol_not_scored
## 8 cuisine_descr grade_date      inspection_type2
```

# EDA

Descriptive analysis of individual variables . . .

## EDA (Action taken)

Records of violations in the vast majority of cases.



## EDA (Violation type)

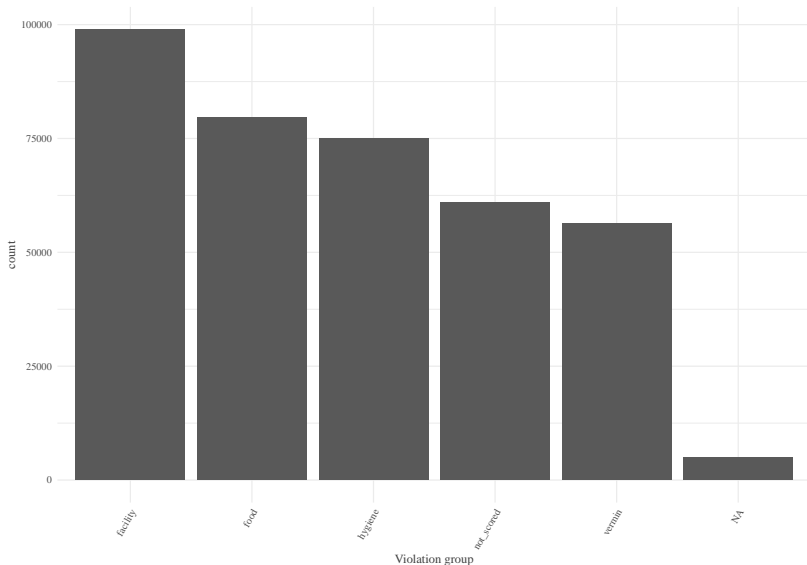
10F (general violation pertaining to non-food contact surfaces) is the most common violation type. followed by 08A (facility not vermin proof), 04L (Evidence of mice).

```
## # A tibble: 10 x 3
```

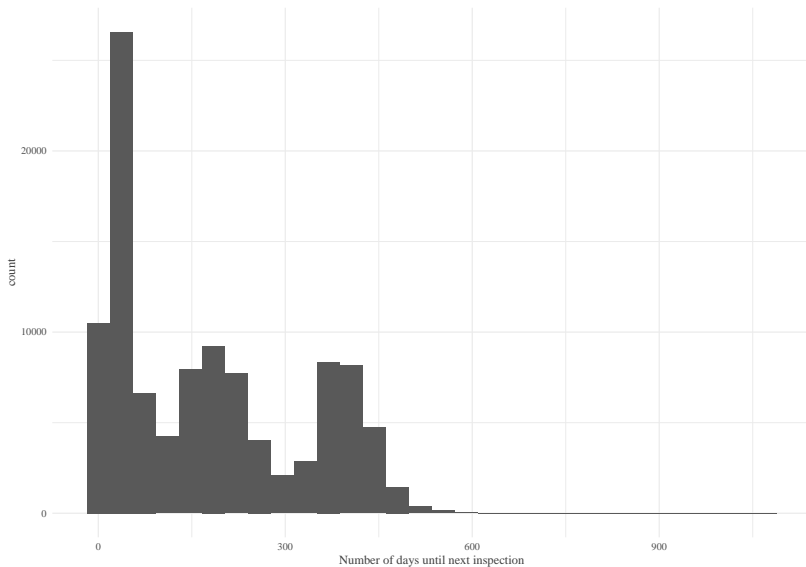
##		violation_code	number	rank
##		<chr>	<int>	<dbl>
##	1	10F	53154	1
##	2	08A	38706	2
##	3	04L	26768	3
##	4	06C	25521	4
##	5	06D	25215	5
##	6	02G	23866	6
##	7	10B	21819	7
##	8	02B	19023	8
##	9	04N	18882	9
##	10	04H	8174	10

## EDA (Violation type ... continued)

We will gather these codes into the broader categories: *facility*, *food*, *vermin*, *hygiene*, and *not\_scored*.



## Target: Number of days until next inspection





# Features

- ▶ **score** of the inspection
- ▶ **grade** of the inspection
- ▶ **critical flag**, which indicates whether the inspection received a critical flag
- ▶ **inspection type** for the given inspection
- ▶ **cuisine** of the restaurant under inspection
- ▶ **Number of violations** per inspection for a given restaurant for each of the categories *food*, *vermin*, etc.

## First models: Linear Regression and Regression tree

- ▶ A linear regression model with all predictors performs poorly in terms of fit to training data.

```
## # A tibble: 1 x 1
##   Rsquared
## *      <dbl>
## 1      0.477
```

The coefficient of determination is 0.477, i.e., the model explains 47 percent of the total variation in the target feature.

- ▶ A Regression tree model employing 10-fold cross-validation performs only slightly better:

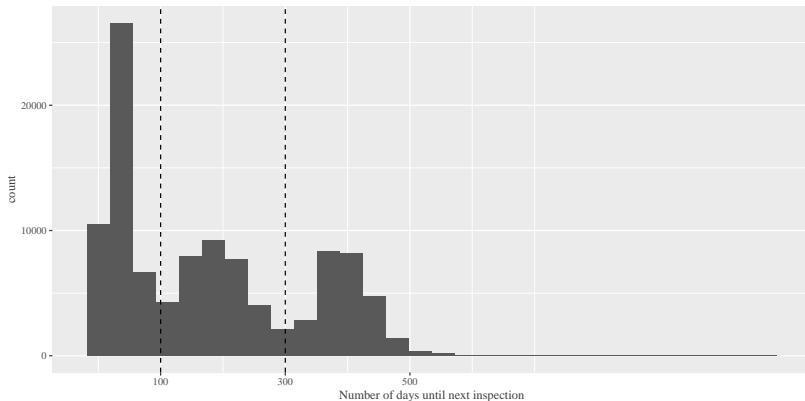
```
## # A tibble: 1 x 1
##   Rsquared
## *      <dbl>
## 1      0.512
```

# Reformulating the problem

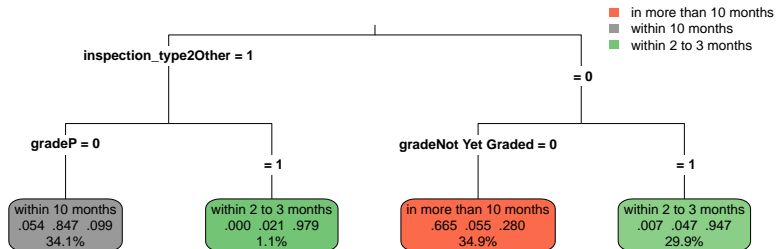
- ▶ It seems that the problem in its current formulation (prediction of number of days) is hard to solve.
- ▶ Instead try: prediction of time interval the next inspection is likely to fall into.
- ▶ Less ambitious. Nevertheless, potentially useful.

# Classification

- ▶ Frame the initial task as classification problem
- ▶ Distribution of the target feature: three humps
- ▶ Partition range accordingly: *within the next 3 months, within the next 10 months, more than 10 months*
- ▶ Method: Classification trees, Cross-Validation to choose a best model.



# Visualization of model



## Interpretation:

The model predicts the next inspection to occur *within 2 to 3 months* if the last inspection was an initial inspection and the restaurant did not receive a grade. If the last inspection was not initial and the restaurant did not receive a “grade-pending”-card, the next inspection occurs *within 10 months*. Finally, if the last inspection was an initial one, and no grade was awarded, the model predicts the next inspection to occur *in more than 10 months*.

## Performance:

The final model has an accuracy of 0.81.

```
## # A tibble: 1 x 1
##   Accuracy
## *      <dbl>
## 1      0.815
```

Accuracy on its own is not that informative, so we will have to consider other evaluation metrics as well:

- ▶ Specificity (proportion of actual positives that are correctly identified as such)
- ▶ Sensitivity (proportion of actual negatives that are correctly identified as such)

Knowing the proportions of the classes, we can use these measures to calculate the probability that the prediction of the classifier is correct (Positive Predictive value and Negative Predictive value) (using Bayes' Theorem)

```
## # A tibble: 3 x 3
##   Class          Sensitivity Specificity
##   <chr>          <dbl>         <dbl>
## 1 More_than_10    0.919         0.843
## 2 Within_10      0.896         0.923
## 3 within_2_to_3  0.691         0.972
```

```
## # A tibble: 3 x 3
##   Class          `Pos Pred Value` `Neg Pred Value`
##   <chr>          <dbl>         <dbl>
## 1 More_than_10    0.665         0.968
## 2 Within_10      0.847         0.949
## 3 within_2_to_3  0.948         0.809
```

With the exception of the class *More\_than\_10*, we see positive and predictive values of at least 0.8.

Performance on the test data is similar.



# Conclusion

- ▶ Initial goal (prediction of number of days until next inspection) could not be achieved, so we reframed the problem into one which is easier to handle.
- ▶ The solution of the reframed problem is useful for the potential client (restaurant in NYC), since it is valuable to have a time interval in which to expect the next inspection.
- ▶ The final model could be used in an application that takes the ID of a restaurant and outputs a predicted time interval for the next inspection.