Predicting Building Permit Issuance Times

Project Report by

Aparna Shastry

Orange County,
California, USA

Content

1. Introduction

A building permit is an official approval document issued by a governmental agency that allows you or your contractor to proceed with a construction or remodeling project on one's property. For more details click here. Each city or county has its own office related to buildings, that can do multiple functions like issuing permits, inspecting buildings to enforce safety measures, modifying rules to accommodate needs of the growing population etc. The delays in permit issuance pose serious problems to construction industries and later on real estate agencies.Read this Trulia study and Vancouver city article. We are set out to conduct an analysis and modeling of certain data related building permits.

1.1 Possible Clients and the benefits

Most significant outcome of this study is a tool to have a better idea on within which window a certain building permit is expected to be issued. Thus this project can benefit builders, planners and real estate industry. Planners can have reduced uncertainty and more concrete dates for several phases of their construction projects, builders can accordingly streamline their various constructions and finally real estate industry can keep up better with the demand, due to reduced delays in projects.

1.2 Scope of this Project

Primary objective of this Data Science Project is design a machine learning model that learns using historical building permit data and predicts the time delay in days between permit application and permit issuance. Here, as an example, it is done for the data set obtained for the city of San Francisco, California, USA. The process is similar for other cities, although there might be slight differences in the attributes of data. For the city of San Francisco, permit issuing is taken care by Permit Services wing of Department of Building Inspection (henceforth called DBI). Since it is not possible to accurately predict the delay in resolution of days, the problem is limited to predicting if a permit will be issued in a week, or in 3 months or beyond 3 months.

Apart from this, a few insights are drawn from the data to answer a few questions that might interest the applicants, or those who want to apply.

2. Data Description / Data Wrangling

2.1 Data retrieval

Data used to get the results explained in next sections is available in San Francisco city open data portal. It is updated every Saturday.

Step by step process to download:

- Go to the link: SF data portal.

- Click on Filter and "Add a Filter Condition". A drop down menu appears.
- Select, "Filed Date" and "is after".
- Enter date as 12/31/2012, because I wanted to do analysis of last 4-5 years. I think most recent data is important in matters such as this, the city council policies could change, there might be new rules, new employers to expedite process etc. Old data may not be too useful in modeling.

CSV format is chosen because it is less than 100MB size and easy to load into notebook. There are other methods like downloading json, or using socrata. This is found to be more reliable and less dependent on any extra libraries.

Date of download for this analysis: The file as of Feb 25, 2018 (Sunday) has been downloaded and kept locally for easy access. Size is about 75MB. The results of this analysis can be reproduced only if one more filter is used in the second step above, to select “Filed Date” “is before” and put Feb 26th, 2018.

2.2 Data Attributes

The data downloaded for 5+ years has close to 198,900 records and 43 columns. Here is the table containing the column names.

Sl No	**Column name**	**Description**	**Number of unique values [Make a note if < 100 non-null entries]**
1	Permit Number	Number assigned while filing	198900
2	Permit Type	Type of the permit represented numerically.	8
3	Permit Type Definition	Description of the Permit type, for example new construction, alterations	8
4	Permit Creation Date	Date on which permit created, later than or same as filing date	N.A.
5	Block	Related to address	4896
6	Lot	Related to address	1055
7	Street Number	Related to address	5099
8	Street Number Suffix	Related to address	18

9	Street Name	Related to address	1704
10	Street Name Suffix	Related to address	21
11	Unit	Unit of a building	660
12	Unit suffix	Suffix if any, for the unit	164
13	Description	Details about purpose of the permit. Example: reroofing, bathroom renovation	134272
14	Current Status	Current status of the permit application. This can have “filed”, “issued”,”completed”, and also many more, like “withdrawn”, “plancheck”, “cancelled”	14
15	Current Status Date	Date at which current status was entered	N.A
16	Filed Date	Filed date for the permit	N.A
17	Issued Date	Issued date for the permit	N.A
18	Completed Date	The date on which project was completed, applicable if Current Status = “completed”	N.A
19	First Construction Document Date	Date on which construction was documented	N.A
20	Structural Notification	Notification to meet some legal need, given or not	1 (it is either Y or blank)
21	Number of Existing Stories	Number of existing stories in the building. Not applicable for certain permit types	64
22	Number of Proposed Stories	Number of proposed stories for the construction/alteration	64
23	Voluntary Soft-Story Retrofit	Soft story to meet earth quake regulations	1 unique entry (it is either Y or blank) 35 Y only, not useful

24	Fire Only Permit	Fire hazard prevention related permit	1 (it is either Y or blank)
25	Permit Expiration Date	Expiration date related to issued permit.	N.A
26	Estimated Cost	Initial estimation of the cost of the project	N.A
27	Revised Cost	Revised estimation of the cost of the project	N.A
28	Existing Use	Existing use of the building	93
29	Existing Units	Existing number of units	348
30	Proposed Use	Proposed use of the building	94
31	Proposed Units	Proposed number of units	368
32	Plansets	Plan representation indicating the general design intent of the foundation..	8
33	TIDF Compliance	TIDF compliant or not, this is a new legal requirement	2 unique types, but 2 non-null entries only
34	Existing Construction Type	Construction type, existing,as categories represented numerically	5
35	Existing Construction Type Description	Description of the above, for example, wood or other construction types	5
36	Proposed Construction Type	Construction type, proposed, as categories represented numerically	5
37	Proposed Construction Type Description	Description of the above	5
38	Site Permit	Permit for site	1, Y or blank

39	Supervisor District	Supervisor District to which the building location belongs to	11
40	Neighborhoods - Analysis Boundaries	Neighborhood to which the building location belongs to	41
41	Zipcode	Zipcode of building address	27
42	Location	Location in latitude, longitude pair.	57604
43	Record ID	Some ID, not useful for this	As many as permit numbers

As obvious from the table, not all 43 attributes are useful for learning from the data. This leaves us with a lot of scope for data wrangling. The code is in BuildingPermitDataWrangling.ipynb

2.3 Cleaning up

Columns to Retain:

- A few columns have numeric and text versions both. Only numerics were retained.
- Location information is in many columns, like Block, lot, street number, name, unit, Zipcode, neighborhood, supervisor district and Location. Location is numerical and more precise. Hence retained only Location.
- Permit Number and Record ID are not useful to analysis and prediction, dropped.
- Permit Creation date, expiry date, First construction document date are irrelevant to the problem.
- TIDF Compliance, Voluntary soft-story retrofit suffer from lack of non-null entries, not even 100. Hence dropped.
- Estimated Cost is not necessary, as there is Revised cost, which is more meaningful and recent.
- Current Status and Current status date are eliminated after using them for a few row elimination
- Issue date is eliminated after creating the ‘time_taken’ column

Rows to Retain:

a) Current Status had 14 types, of which withdrawn, cancelled and disapproved status and not having issue dates are not relevant for further study. Hence rows corresponding to

these records are dropped. This was about 2k in total. After doing this Current Status column is eliminated.

b) Records corresponding to no location also had to be dropped, as it made no sense in the next stages.
c) Rows with file dates corresponding to beyond Sep 30th, 2017 are dropped, after computing the time taken variable. This is done at EDA after seeing the summary statistics

Cleaning the NaNs:

a) Fire Only Permit, Site Permit and Structural Notification had only Y and blank entries. Blanks are interpreted as N, and replaced with N.
b) Revised cost NaNs were filled with 0's initially and at EDA stage all zeros are filled with 10^(-5) to avoid underflow while taking logarithm.
c) Blanks in existing use and proposed use are filled with strings 'Unknown'
d) All other NaN are left as they are, because it means "Not applicable" and the categories will be handled as such by the models.

Invalid weekdays:

The DBI is open only from Monday to Friday. Saturday, Sundays are replaced by the nearer weekday to avoid anomalies in EDA.

We are left with the following subset to do the EDA:

1. Permit Type
2. Permit Type Definition (Duplicate, retained for meaningful visualizations)
3. Plansets
4. Fire Only Permit
5. Revised Cost
6. Current Status
7. Filed Date
8. Structural Notification
9. Number of Existing Stories
10. Number of proposed Stories
11. Existing use
12. Proposed Use
13. Existing Construction Type
14. Proposed Construction Type
15. Site permit
16. Location

2.4 Other potential Data set

The process followed in this project can be generalized with minor modifications, for any city's building permit data, provided that it has at least permit filing date and issue date attributes. It is not always guaranteed that both will be there in the database. For example, Los Angeles city data or Chicago did not have application filing date.
Another dataset that has similar attributes to that of SFO is New York city building permit data. This can be used for studying and coming up with model for permit issue delays.

3. Exploratory Data Analysis (EDA)

The project notebook BuildingPermit-EDA.ipynb explains all the details of exploratory data analysis. We will highlight some key findings here, with assumptions made.

3.1 Assumptions made on "time taken" variable:

Firstly, the variable of interest, *"time_taken"-that is the time difference between issue date and filed date in number of days,* revealed a few insights. As some of the permits were not assigned during the download time, there was need to make approximations for the EDA and inferential statistics to work more realistic. Without doing those, the time taken had the following statistics: Note that, the count doesn't include rows without issue dates, as that is how Python pandas works.

Count	183960
Mean	26.05
Std	91.06
Min	0
50%	0
75%	6
Max	1740

Table 3.1 Descriptive statistics of Time taken variable without filling the blank issue dates

1. The options were, dropping the rows with no issue date, or put a hypothetical date. Dropping would introduce bias and make the mean wait time appear smaller than it is.

Not only the mean, but also the median, 75% percentile and the max wait time appear smaller number than they would be, compared to filling the blanks in issue date with at least the download date. Hence issue date was filled as download date. This imputation was done in EDA part, after looking at the difference it would make.

2. We dropped the records corresponding to file date after September 30th 2017, so that whatever records with hypothetical issue dates had surely wait time of at least 150 days. This was a good approximation, looking at the outliers of the data. Later on while modeling, this will be brought up again.

The modified statistics after these two approximations:

Count	180811
Mean	67.46
Std	221.42
Min	0
50%	0
75%	13
Max	1903

Table 3.2 Descriptive statistics of Time taken variable after filling the blank issue dates as download date

Now the problem became slightly more interesting with at least 75 percentile having a value of 13. The following table gives a better split on the percentage permits issued with dropping NaTs and without dropping NaT's, records for file dates Jan 2013 - Sept 2017:

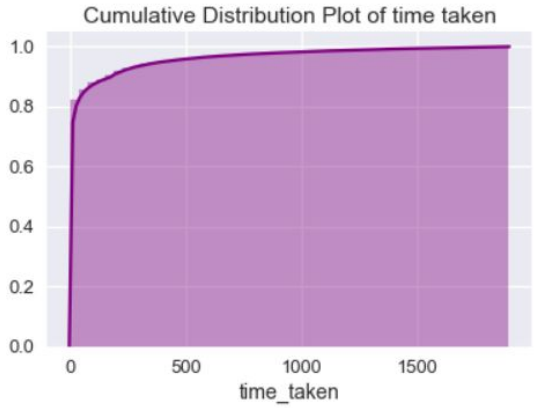
```
Number of records: 170832 for first column and 180811 for second column
```

	Percentage with dropping	Percentage with hypothetical issue date
same day	62.21	58.77
less than 15 days	80.74	76.29
less than 3 months	91.97	86.90
less than 6 months	95.19	90.02
less than a year	98.26	94.55

Table 3.3 Summary of difference in time taken percentages

There are 8 types of permits which are explained in section 3.4. The type 8 is over the counter (OTC) alterations, and we are interested to see also the distribution of time taken without it.

Hence there are figures 3.1a and b. These are the cumulative distribution function of time_taken variable, after filling NaT in issue dates and taking the difference



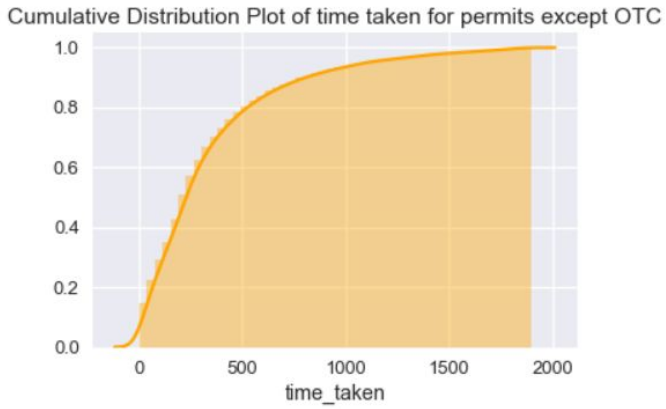


Fig 3.1a CDF of the time taken(days) 3.1b CDF of the time taken excluding OTC permits

3.2 What is the best day of the week to visit DBI?

General belief is that Wednesday being the middle of the week is least crowded. Is that true?

It is found that Monday is the least crowded day and also on Mondays mean wait times are lower than other days, and also the probability of permits getting processed same day is highest.

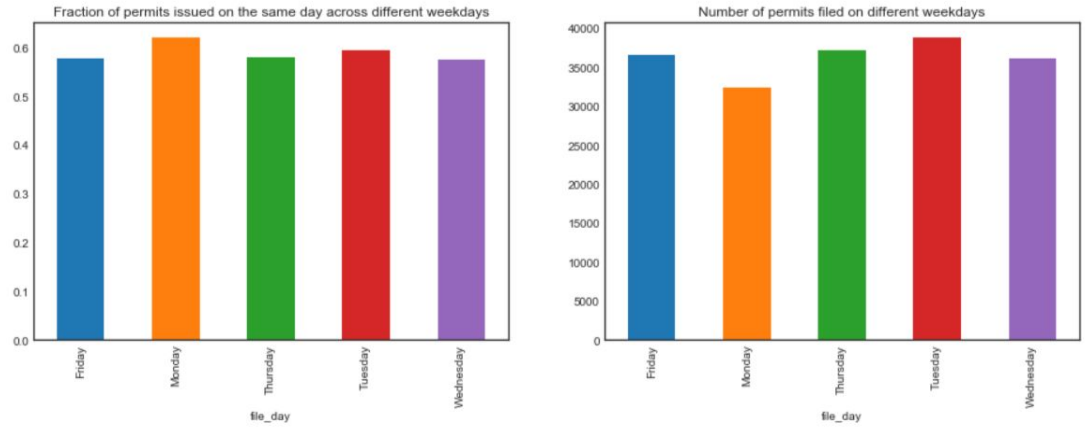
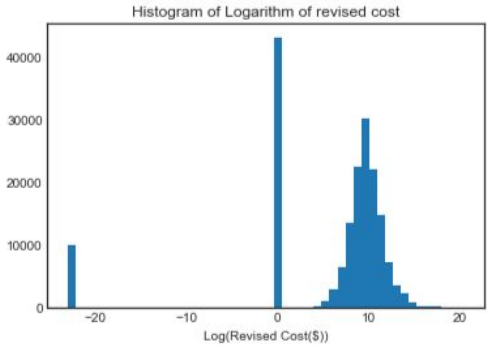


Fig 3.2 Bar Charts showing time taken Vs Day of the weeks

3.3 How does the histogram of Revised Cost look? How is it related to “time_taken”?

The plain scatter plot of Revised cost is rather messy. Hence we took logarithm. Many applicants do not prefer to reveal the cost. There are about 28-29% entries which are less than 10$. This can not be accident, purposely the builders do not fill the cost field. The following plots clearly show it: First one is histogram of logarithm of revised cost, second is scatter plot against time_taken.



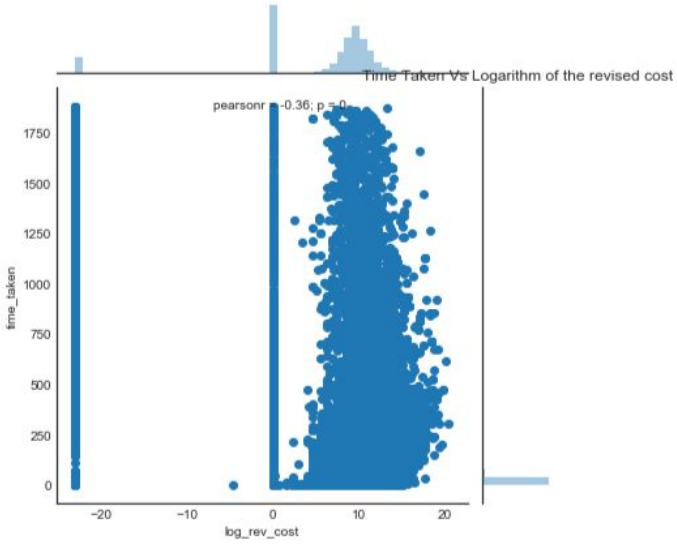


Fig 3.3a Histogram of (Log Revised Cost)

Fig 3.3b Scatter plot of time taken Vs Log (Revised cost)

It is also found that when revised cost is put as 0 or NaN, except for permit type OTC alterations, none of them get issued the same day, and the minimum delay is around 5 months for all except demolitions. The demolitions without cost entry has minimum delay of 39 days. **It is recommended to put a realistic number in revised cost field of the application.**

3.4 How does the time taken vary across permit types?

This is the summary statistics table for time taken Vs permit types. Notice that the plain new construction applications take minimum 2 months, although they are very small portion of the total applications. OTC alterations permits dominate with more than 80% representation and as the name OTC (Over the counter) suggests, they are supposed to be issued the same day. But some are not!

perm_typ_def	count	mean	std	min	25%	50%	75%	max
new construction	301.0	570.810631	344.639562	60.0	329.00	460.0	762.00	1745.0
new construction wood frame	873.0	507.321879	380.214589	2.0	221.00	409.0	763.00	1837.0
demolitions	516.0	463.164729	387.444375	0.0	159.00	368.0	706.50	1824.0
additions alterations or repairs	12597.0	345.122728	329.004391	0.0	135.00	240.0	436.00	1875.0
wall or painted sign	433.0	253.092379	439.792013	0.0	3.00	30.0	231.00	1859.0
grade or quarry or fill or excavate	88.0	203.784091	387.183833	0.0	44.75	81.0	156.25	1803.0
sign - erect	2587.0	153.627754	334.840111	0.0	2.00	15.0	126.00	1878.0
otc alterations permit	163416.0	38.200904	176.576709	0.0	0.00	0.0	5.00	1880.0

Table 3.4 Time Taken Statistics by Permit types

Below is the box plot of the table 3.4

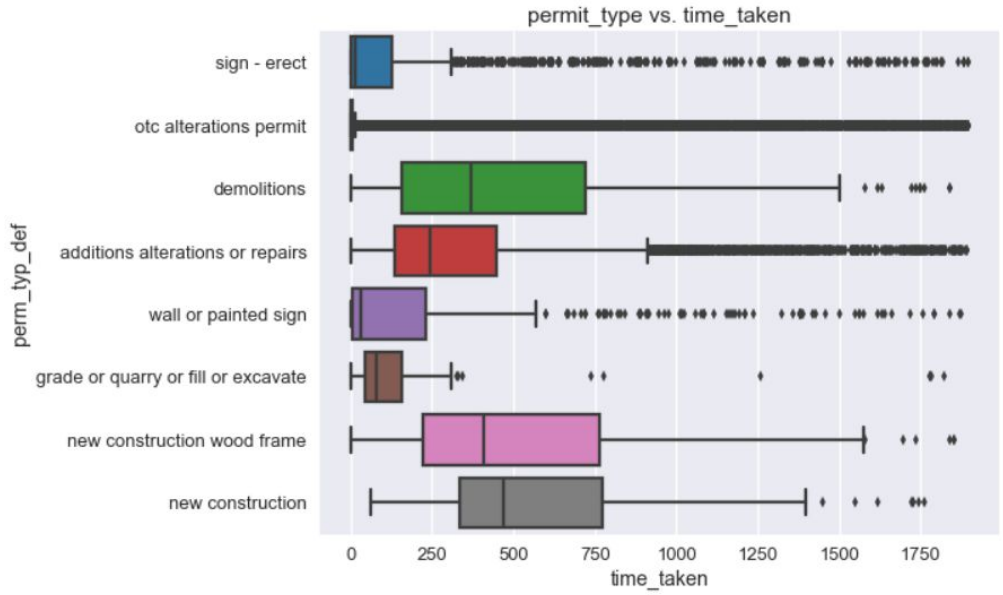


Fig 3.4 Box plot of time taken across Permit type

3.5 How does the time taken vary across Plansets?

	count	mean	std	min	25%	50%	75%	max
plansets								
0.0	58456.0	12.55	112.58	0.0	0.0	0.0	0.0	1879.0
1.0	2.0	276.00	0.00	276.0	276.0	276.0	276.0	276.0
2.0	88297.0	122.42	279.08	0.0	1.0	11.0	90.0	1880.0
3.0	246.0	231.10	70.51	72.0	225.0	225.0	225.0	1178.0
4.0	3.0	178.33	163.66	55.0	85.5	116.0	240.0	364.0
6.0	2.0	327.00	0.00	327.0	327.0	327.0	327.0	327.0
20.0	1.0	1241.00	NaN	1241.0	1241.0	1241.0	1241.0	1241.0
9000.0	1.0	2.00	NaN	2.0	2.0	2.0	2.0	2.0

Table 3.5 Variation across Plansets

The data set mainly has Plansets 0, 2 and 3. Rest are insignificant.

These are the ones that matter the most for modeling. Rest can be referred to in notebook.

3.6 How does Revised cost vary across permit types

perm_typ_def	count	mean	std	min	25%	50%	75%	max
additions alterations or repairs	12560.0	4.265321	13.875115	-23.025851	6.516193e+00	11.002100	12.429216	19.399238
demolitions	514.0	2.722766	14.051440	-23.025851	8.334367e+00	9.615805	11.002100	15.656060
grade or quarry or fill or excavate	88.0	10.951905	6.436913	-23.025851	1.057805e+01	12.899220	14.191652	16.510138
new construction	300.0	3.840977	18.668313	-23.025851	-2.302585e+01	15.795917	17.496244	20.475445
new construction wood frame	873.0	3.442886	16.474418	-23.025851	-2.302585e+01	13.167337	13.652992	18.627695
otc alterations permit	161936.0	5.755213	7.388122	-23.025851	1.000000e-10	8.764053	10.126631	17.034386
sign - erect	2542.0	4.718625	9.472716	-23.025851	6.907755e+00	7.937375	8.517193	11.918391
wall or painted sign	429.0	0.739596	11.826582	-23.025851	4.605170e+00	6.214608	7.090077	10.819778

Table 3.6 Variation of Log of Revised Cost across permit types

The summary statistics is given for reference. The value -23.x is corresponding to no entries substituted by very small values.

4. Inferential Statistics

A few statistical tests were conducted to check some of the assumptions/(hypothetical) claims and below is a summary. Details are available in the Inferential Statistics Report.

4.1. Is the DBI's (hypothetical) claim that on an average 65% of the applicants receive the permits same day true?

There is no sufficient statistical evidence to believe the DBI's claim that on an average it processes 65% of the permit applications the same day. A one sample population proportion test with a significance level of 0.01, on a randomly drawn sample of size 7500 records resulted in alternate hypothesis to be accepted in place of null (default) hypothesis.

1.a What is the mean wait time observed? Give the 95% confidence interval.

A randomly drawn sample of size 7500 records revealed that one can expect DBI to process 58.83% applications the same day, with 95% confidence interval being [57.71, 59.94]

4.2. Is there any difference in mean wait times of fire only permit or non fire only permits? Is this statistically significant?

There is enough statistical evidence to believe that average wait times for fire only permit and the normal permits are not the same. A 2 sample z-test for mean wait times for a significance level 0.01 on sample sizes of 700+ (fire only), 6700+ (not fire only) revealed statistically significant results to believe that mean wait times are different.

4.3. Was there a scope to conduct ANOVA test to compare statistics across various populations like mean time across weekdays or permit types?

Conditions required for ANOVA test were not satisfied. Hence these tests are not done. The variance itself is not at comparable levels.

5. Modeling and Predicting

5.01 Machine Learning Problem in the context of Building business:

The problem is defined as classifying time taken variable into one of the three classes using the independent variables that influence it.

Other possible definitions :

1. Why not Regression? Recall from Table 3.2 that median time taken is 0 and even 75% is just 13 days. There are a few applications which take too long to be issued but their percentage is low, as seen clearly from Table 3.3. It won't be possible to train a good regression model when it is so uneven to the extent that valid response looks like outlier and valid predictors look like high leverage points.
2. Why not binary classification? There are several categories of permits/permit applicants. Some of them will be interested to know if permit will be issued the same day, some will be interested to know if it will be within a week, or within 2 weeks or within 3 months or beyond 3 months. Recall that new building permits take minimum 2 months. Hence the binary classification of within a week or after a week, or within 2 weeks or after 2 weeks, is not of any value to this category, where in fact stakes are very high.

The code for Modeling can be found in notebook BuildingPermitSFOModeling.ipynb

5.02 Target and Predictor Variables

Let the target variable be y and define,

y = 0, if time_taken < 8 days
= 1, if time_taken > 7 days, < 92 days
= 2 else

	time_taken
0	0.718
1	0.151
2	0.130

Fig 5.1 Distribution of Target Variable Y

Predictors are,

1. Revised Cost
2. Latitude
3. Longitude
4. Permit Type
5. Plansets
6. Fire Only Permit
7. Site Permit
8. Structural Notification
9. Existing Use
10. Proposed Use
11. Existing Construction Type
12. Proposed Construction Type
13. Existing number of stories
14. Proposed number of stories
15. File day (Day of the week extracted from file date)
16. File month (Month in which application was filed)

Out of 43 columns in the data downloaded, there are only 16 useful features! Welcome to the real data science world!

5.03: Train/Dev/Test set split:

There were 180000+ clean records, which is quite a lot. This was split as Train: Development (dev): Test set in the ratio 60:20:20. Further the train set was used in K-fold cross validation to tune hyperparameters.

5.04: Metrics:

Before diving into models, a note on metrics used:

The process followed is, fitting the data into several machine learning models and evaluating their performance on the dev data. In binary classification problem, normally comparison of models is done using Area Under their respective Receiver Operating Characteristics (ROC) curve, the AUC score. In multi-class classification problem, however, it is not a good metric, as each class has to be treated as one Vs rest and there are as many number of AUC scores as there are classes. There could be inconsistencies, for example, AUC score of class k is better with Logistic Regression, but that for class k is better with Decision Tree.

For this problem, weighted f1-score as given by the classification_report function of scikit-learn is chosen as the metric to compare different models, and also in 5-fold cross validation.

5.05: Feature Engineering:

a) Predictors like Existing use and proposed use have too many categories to be really useful. Also, it is likely that many of them are equal and hence correlated, because as we saw, most permit types are alteration permits. Instead of dealing with numerous categorical variables, generated new binary predictors, dff_use = existing_use != proposed_use, and diff_story = existing_use != proposed_story

b) For revised cost predictor, Logarithm of revised cost is taken and used in lieu of original predictor, as Log of the cost is better correlated to time_taken.
c) Also experimented with square and cube of log of revised cost and these became 2 more features.
d) Week day of the filing and month of filing were not explicitly in data set, they were extracted from the filing date.
e) Location was opened up into Latitude and Longitude.

5.06 Feature Selection:

This is done at the modeling stage. With Random Forests, iteratively added a few features and removed to see the difference in performance.

5.06: One hot encoding: One hot encoding was done for all categorical predictor variables, before feeding the data to Logistic Regression model. The correlation was examined and redundant ones were dropped. Fig 5.2 shows the heatmap after dropping redundant categories.

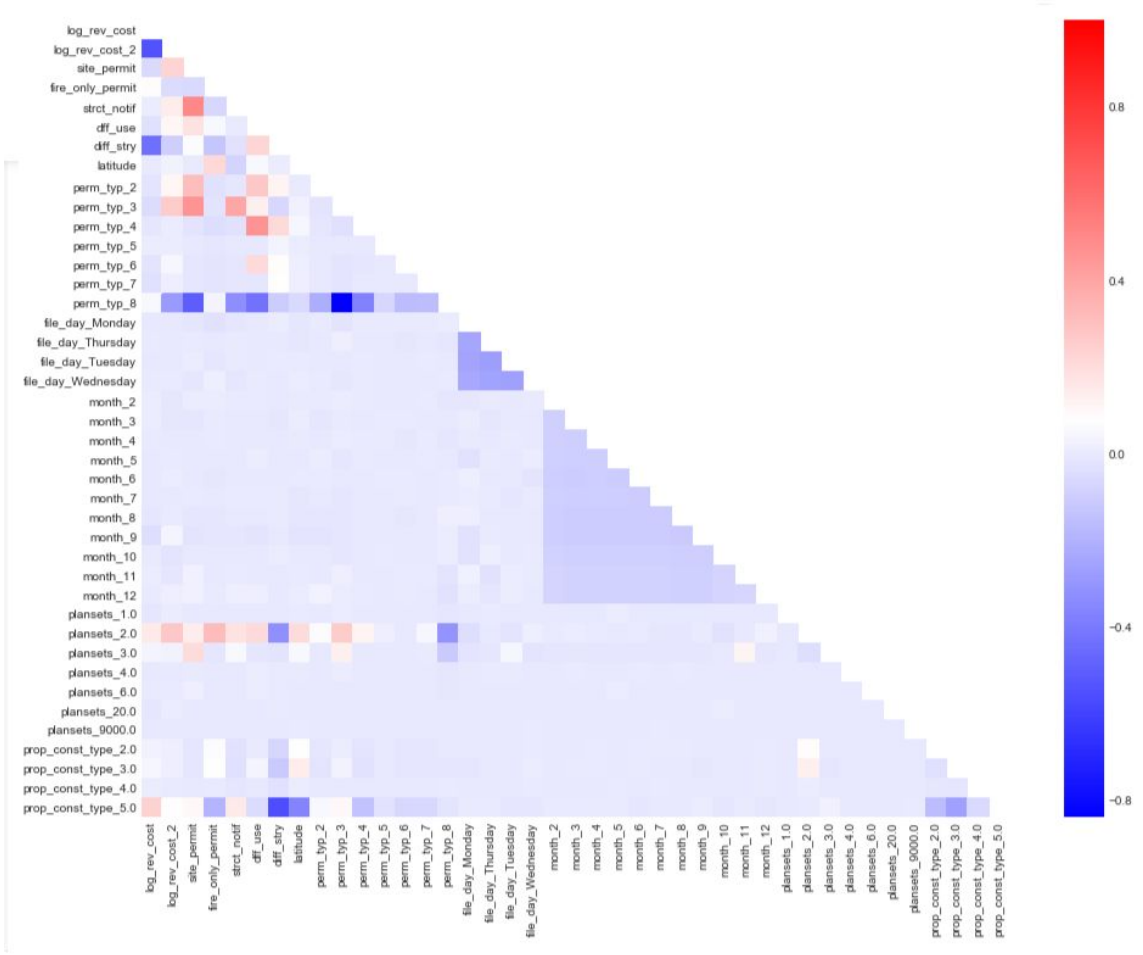


Fig 5.2 Heat Map of predictors for Logistic Regression

From the above correlation heatmap, it is observed that predictors do not suffer from multicollinearity. There are 41 predictors.

5.07: Checking for bias in Data set

To examine the bias in the samples, a logistic regression model was trained and tested with varying number of training samples. The training samples were shuffled to introduce random ordering. Number of training samples increased from a small number like total samples / 100, in steps of total samples / 100. Very soon, the train and development sample error curves met and reached steady state. This shows that there's bias. One way to overcome is by adding more features, or creating/engineering more features from the existing. Already explained in section 5.05

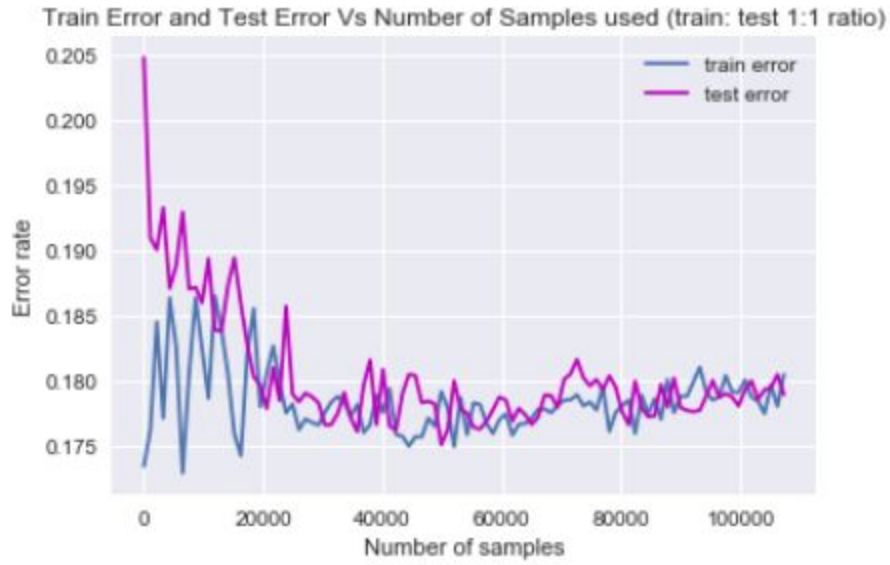


Fig 5.3 Train and Test Error Rate curve for different sample sizes

Now let us look at results of fitting and predicting the class y using a few well known machine learning models. Tried Logistic Regression, Decision Trees, Bagging Classifier, Random Forests and Gradient Boosting methods. A 5-fold cross validation on training set was used to tune hyperparameters, and the model was tested against dev set.

5.1 Comparison of Results with Various Models:

Model	Hyperparameters	Accuracy , f1 -score on dev data
Logistic Regression	C:[10,1000,100000] Chosen: 10000	78.82% , 0.81
Decision Tree	max_depth: [6,8,12,14], min_samples_leaf: [1,2,4,6] Chosen:12,1 respectively	82.54% , 0.82

	Works the same without any parameters as well	
Bagging Classifier	n_estimators:[40,50,100,200,300] Chosen: 200	84.59%,0.84
Random Forest	n_estimators: [40,50,100,200,300] Chosen: 200	82.68%,, 0.83
Gradient Boosting Classifier	n_estimators: 50 (Hyperparameter tuning was tried and the code is removed after choosing)	82.63%,0.0.82

Table 5.1 Comparison of Results

5.2 Comparison of Feature importance:

Feature Importance (Normalized to make max = 1)

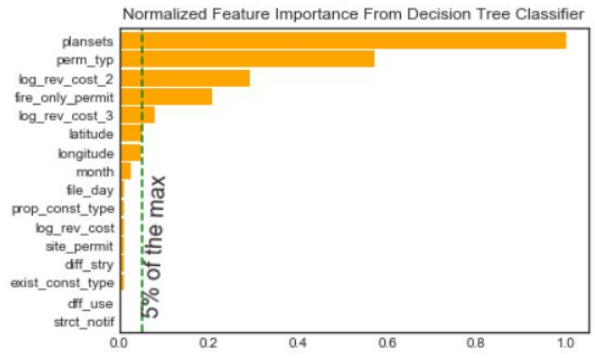


Fig 5.4a Feature Importance in Decision Trees

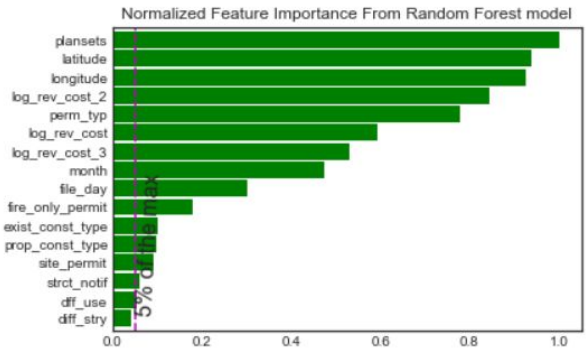


Fig 5.4b Feature Importance in Random Forest

5.3 Evaluation of Effectiveness of Feature Selection/Engineering:

Decision Tree and Random Forest methods were evaluated by incrementally adding features by their importance. It is found that Decision Tree saturates in performance with just about 4 features and Random Forest does better with addition of each, until all are added to reach the highest dev set accuracy. But mainly around 5 features matter the most for Random Forest.

Features	**Accuracy and f1 score**
Plansets alone	63.73%,0.65

Plansets, Permit type	69%, 0.73
Plansets, Permit type, log_rev_cost_2	73.65%,0.77
Plansets, Permit type, log_rev_cost_2, longitude	80%,0.81
Plansets, Permit type, log_rev_cost_2, longitude, latitude	**82%,0.83**
Plansets, Permit type, log_rev_cost_2, longitude, latitude, log_rev_cost, log_rev_cost_3, month, file_day	83%, 0.83

5.4 Additional models tried/parameters tuned:

1. Tried Logistic Regression with multinomial multiclass and other solvers like 'newton-cg' and 'sag'. They did not converge. Resorted back to default 'ovr'
2. Explicitly did a One Vs Rest Multiclass classifier using Logistic Regression as base model. This did not give any predictions for certain test samples. This could happen.
3. Support Vector Classifier was tried: It was too slow, did not even finish the run after 2 hours
4. All models were tried with cross validation of default score function (accuracy) as well. Did not see any difference in overall accuracy

6. Conclusion / Work Remaining

Did data cleaning, EDA and inferential statistics on the data. Defined the time taken variable as a 3 - class classification problem, fitted a few models, did hyperparameter tuning and evaluated the performance.

6.1 Final Choice of Model

Random Forest is chosen. From the table, it might appear that bagging classifier gives better performance, but we are interested in accuracy of label 1 as well, and Random Forest with class weight balanced gives a better recall rate on label 1.

```
BEST {'n_estimators': 200}
############ Bagging ################
Accuracy on training data: 99.92%
Accuracy on test data:     84.59%
confusion_matrix on dev data
[[24406  1460   115]
 [ 2375  2763   335]
 [  693   594  3422]]
classification report on dev data
             precision    recall  f1-score   support

          0       0.89      0.94      0.91     25981
          1       0.57      0.50      0.54      5473
          2       0.88      0.73      0.80      4709

avg / total       0.84      0.85      0.84     36163

##################################################
Wall time: 10min 4s
```

```
BEST {'class_weight': 'balanced', 'max_depth': None, 'min_samples_leaf': 2,
############ RandomForest ################    'n_estimators': 200}
Accuracy on training data: 95.35%
Accuracy on test data:     82.68%
confusion_matrix on dev data
[[22723  3125   133]
 [ 1344  3757   372]
 [  427   861  3421]]
classification report on dev data
             precision    recall  f1-score   support

          0       0.93      0.87      0.90     25981
          1       0.49      0.69      0.57      5473
          2       0.87      0.73      0.79      4709

avg / total       0.85      0.83      0.84     36163

##################################################
Wall time: 1h 7min 23s
```

6.1.1 Evaluation of Predictive power on the data never seen before: The hold out set gave a prediction accuracy of 83% (Highlighted below Recall column)

```
In [27]: # The final Score using the chosen model and held out test set
         ypred = rf.predict(Xte)
         print(confusion_matrix(yte,ypred))
         print(classification_report(yte,ypred))
```

```
[[22828  3033   119]
 [ 1277  3837   359]
 [  418   850  3441]]
             precision    recall  f1-score   support

          0       0.93      0.88      0.90     25980
          1       0.50      0.70      0.58      5473
          2       0.88      0.73      0.80      4709

avg / total       0.86      0.83      0.84     36162
```

Accuracy

6.2 Work Remaining

1. Gradient Boosting is not fully explored yet. Perhaps not needed at this point.
2. There is a scope for correlating with housing price data set and improving accuracy.

7. References

1. Introduction to Statistical Learning (Book)
2. Machine Learning Course by Andrew Ng
3. Inferential Statistics Course by Mine Çetinkaya-Rundel
4. Datacamp lectures, Springboard exercises

Code is available here

Author	Revision History Version / Date	Comments
Aparna Shastry	0.1/03.05.2018	Initial Draft
	0.2/03.06.2018	Modified after incorporating Mentor's feedback
	1.0/03.24.2018	Final release for Completion