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 In case of categories, Also mention 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 (it is either Y or blank) 35 Y only.
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 set type for the construction.	8
33	TIDF Compliance	TIDF compliant or not, this is a new legal requirement	2 types, 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 munging.

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, so dropped. Permit Creation date, current status 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.
We are left with the following subset to do the EDA:

1. Permit Type
2. Permit Type Definition (Duplicate, but retained for meaning)
3. Plansets
4. Fire Only Permit
5. Revised Cost
6. Current Status
7. Filed Date
8. Issued Date
9. Structural Notification

10. Number of Existing Stories
11. Number of proposed Stories
12. Existing use
13. Proposed Use
14. Existing Construction Type
15. Proposed Construction Type
16. Site permit
17. Location

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.

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.

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 BuildingPermitSFO.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 main variable of interest, *"time_taken"-that is the time difference between issue date and filed date in number of days,* revealed some interesting insights. As some of the permits were not assigned during the download time, we had to do some approximations for the EDA and inferential statistics to work more realistic. Without doing any approximations, the time taken had the following statistics: Note that, the count doesn't include rows without issue dates

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. We had the option of 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 mean, the median, 75% percentile and the max wait time also be smaller number than it would be compared to assuming an issue date at least as late as the download date. Hence issue date was fixed at 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	66.20
Std	217.54
Min	0
50%	0
75%	13
Max	1880

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	90.61
less than a year	98.26	94.64

Table 3.3 Summary of difference in time taken percentages

This is the cumulative distribution function of time_taken variable, after filling NaT in issue dates and taking the difference

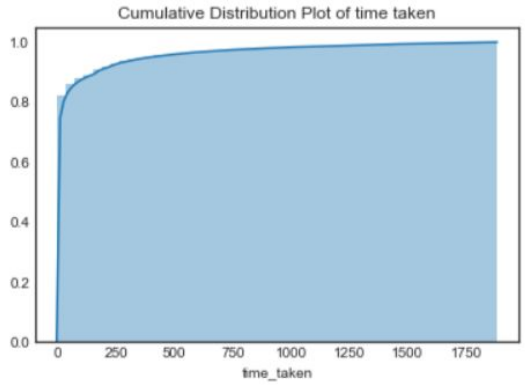


Fig 3.1 CDF of the time taken(days)

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.

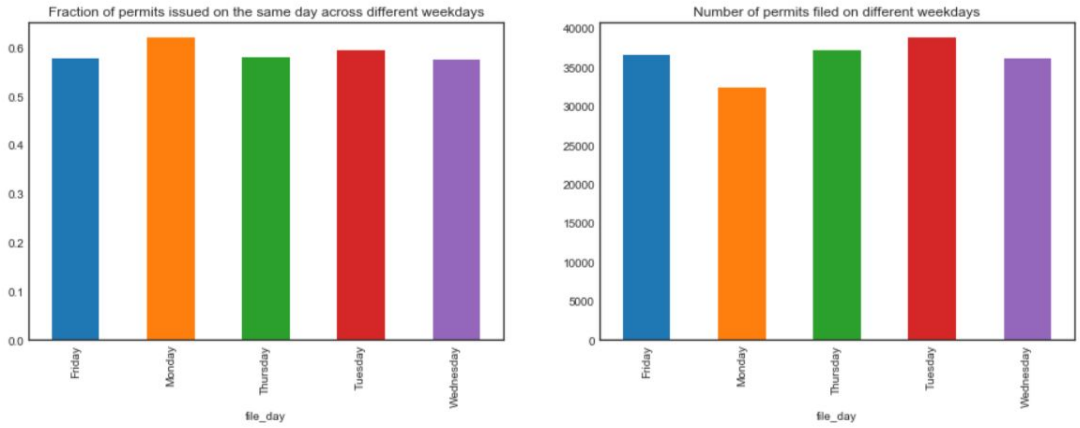


Fig 3.2 Bar Charts showing time taken Vs Day of the weeks

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. The following plots clearly show it: First one is histogram of logarithm of revised cost, second is scatter plot against time_taken.

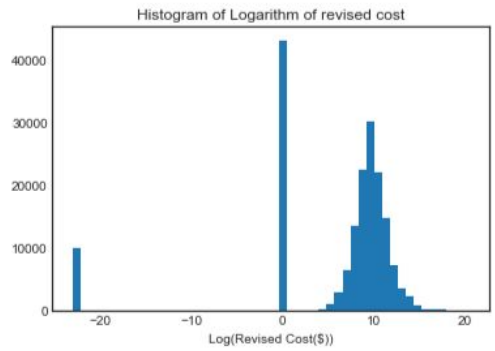


Fig 3.3a Histogram of (Log Revised Cost)

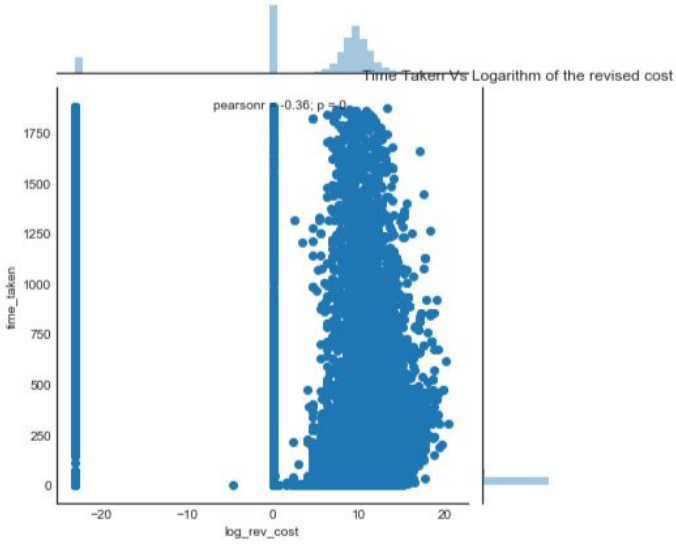


Fig 3.3b Scatter plot of time taken Vs Log (Revised cost)

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, even that 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 descriptive 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

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

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 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 significance 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?

We could not conduct statistical tests to compare the average wait times across various permit types because conditions required for ANOVA test were not satisfied

5. Modeling and Predicting

5.01 Machine Learning Problem in the context of Building business:

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

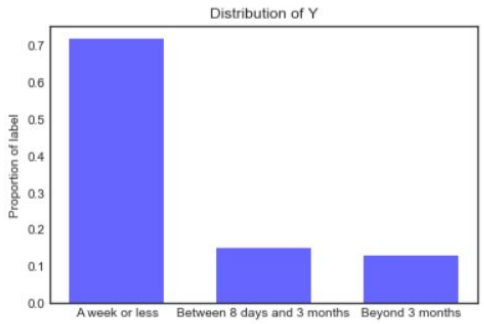
Other possible definitions :

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.

5.02 Target and Predictor Variables

Let us call our target variable as y and define,

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



	Y
0	0.718
1	0.152
2	0.131

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, we are left with 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:

We will be fitting the data into several machine learning models and evaluating their performance on the dev data. In binary classification problem, normally compare models 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, we choose primarily f1-score as given by the classification_report function of scikit-learn as the metric to compare different models. We will also ensure that model chosen with this metric gives better dev set accuracy compared to other models.

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, we chose to generate 2 new predictors, dff_use = existing_use != proposed_use, and diff_story = existing_use != proposed_story
b) For revised cost predictor, we took Logarithm of revised cost and used it 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 it to Logistic Regression model. The correlation was examined and some redundant ones were dropped. Fig 5.2 shows the heatmap after dropping redundant categories.

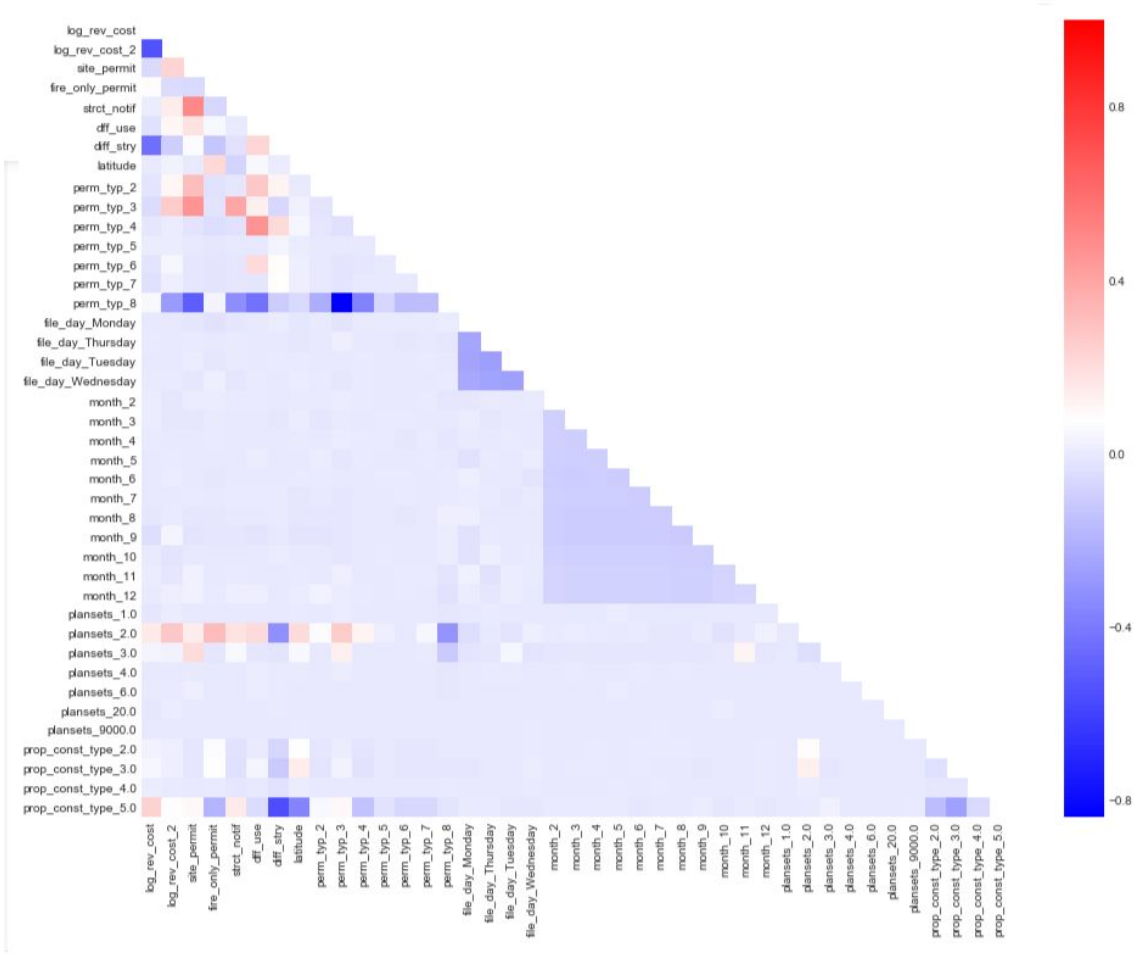


Fig 5.2 Heat Map of predictors for Logistic Regression

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

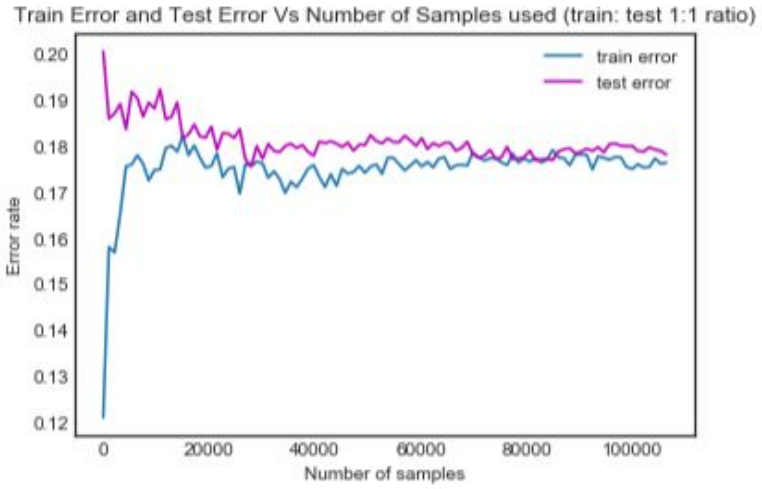


Fig 5.3 Train and Test Error Rate curve for different sample sizes

Now let us look at results of fitting and predicting the class y using a few well known machine learning models. Tried Logistic Regression, Support Vector Machines (with reduced records), 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	Confusion Matrix on Dev Data	Time taken to Train + Predict
Logistic Regression	C:[10,1000,100000] Chosen: 1000	82.02% , 0.80	[24584 1124 16] [3330 1691 413] [1047 517 3127]	35 sec
Decision Tree	max_depth: [6,8,12,14], min_samples_leaf: [1,2,4,6] Chosen:12,1 respectively Works the same without any parameters as well	82.54% , 0.81	[24347 1294 83] [2998 2075 361] [846 678 3167]	54 sec

Bagging Classifier	n_estimators:[40,50,100,200,300] Chosen: 100	84%,0.84	[24584 1124 16] [3329 1691 414] [1048 517 3126]	> 5min
Random Forest	n_estimators: [40,50,100,200,300] Chosen: 200	84.5%,, 0.84	[24407 1299 18] [2940 2200 294] [875 677 3139]]	> 5 min
Gradient Boosting Classifier	n_estimators:[50,100,200,300] Chosen: 300	82.73%,0. 0.82	[24327 1383 14] [2900 2233 301] [853 739 3099]	Close to 10 min

Table 5.1 Comparison of Results

5.2 Comparison of Feature importance:

Feature Importance (Normalized to make max = 1)

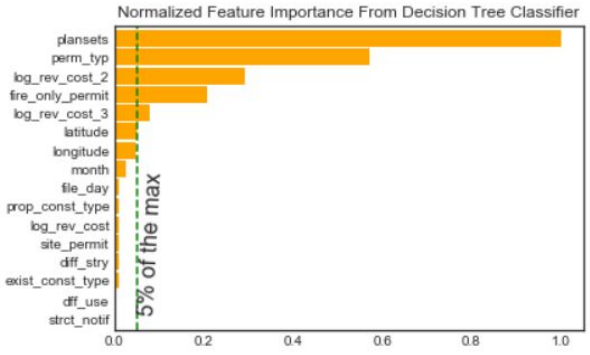


Fig 5.4a Feature Importance in Decision Trees

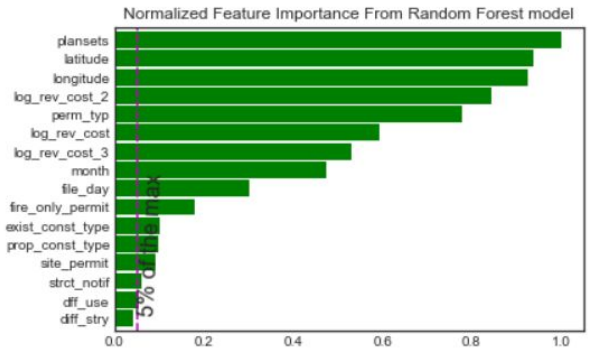


Fig 5.4b Feature Importance in Random Forest

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 6-7 features and Random Forest does better with addition of each, until all are added to reach the highest dev set accuracy.

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 give good results

4. All models were tried with cross validation of default score function (accuracy) as well. Did not see any difference in overall accuracy
5. Class weight argument hyperparameter tuning was considered and tried, but not adopted, as the classes were not extremely imbalanced, but just imbalanced.

6. Conclusion / Work Remaining

We 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. It was pretty OK, but we could do more.

6.1 Choice of Model

Decision Tree: Both Logistic Regression and Decision Tree perform similarly on the dev data with 82.x % accuracy. Decision Tree is a lot faster to train than other methods given in table 5.1 and more interpretable than Logistic Regression. Logistic regression suffers from sparsity of Predictor variable values and abundance of number of predictors introduced by one-hot encoding. They are a lot faster to train and less memory consuming than the others.
Random Forest: Random Forest gives training accuracy of 99.x% and dev accuracy of 84.5%, thus a seemingly overfit. However, taking measures to decrease the gap is found to reduce accuracy of both instead of increase in dev set accuracy. Hence Random Forest has to be used with defaults for all arguments except number of estimators set to 200. The extra time taken to train is fine, because prediction happens fast enough.

Random Forest has to be chosen and delivered to the client.

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.
3. Evaluate the models with test set

7. References

1. Introduction to Statistical Learning
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
	1.0/03.06.2018	Modified after incorporating Mentor's feedback