Predicting Building Permit Issue Times

A Data Science Project
By Aparna Shastry

Content

- Business Problem
- Data / Data Wrangling
- Exploratory Data Analysis
- Predictive Modeling
- Conclusion / Future Work

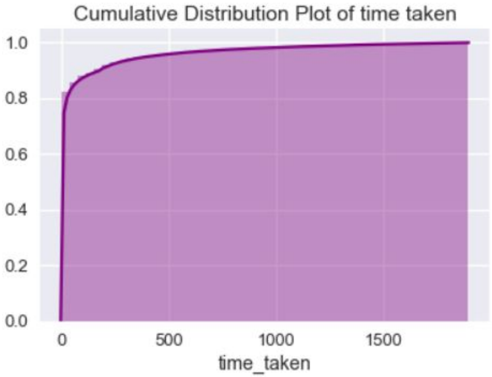
The Business Problem

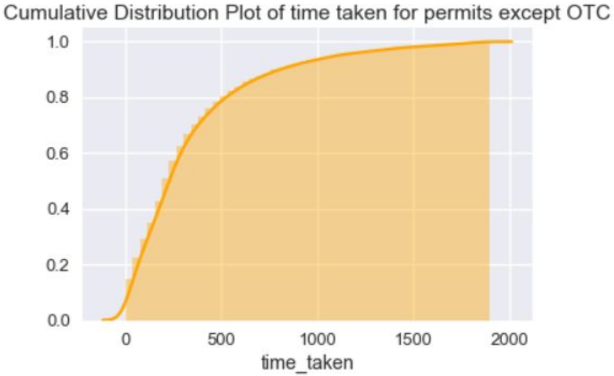
- Building Permit: A Document issued by city council to builders
- **Delays in issuance:** Major inconvenience and Revenue loss
  - Trulia Study
  - Vancouver City Problems
- **Aim of this Study:** Explore the data and model the delays
- **Possible Clients:** Builders / Contractors / Real Estate Agencies
- **How this can help:**
  - Uncertainty Reduction
  - Better planning
  - Proactive follow up by expecting delays

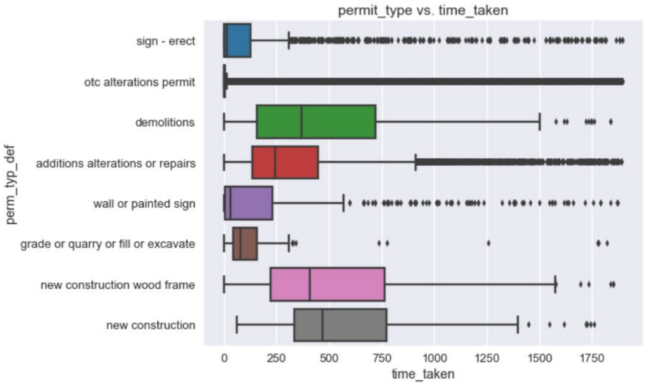
Data / Data Wrangling

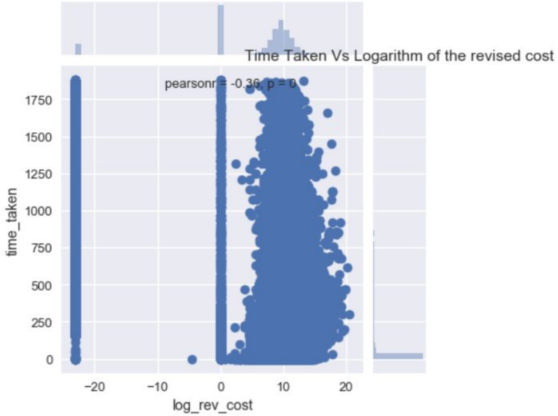
- Any building permit application data
- Took San Francisco city open data, for prototyping
- Raw data: ***198900k records, 43 columns***
  - Download Date Feb 25th, 2018
  - Most recent 5 years
- **Cleaning:**
  - Retain useful columns and rows
  - Fill a few blank cells with “N”
  - Leave a few blank cells as it is
  - Convert object to dates/float when applicable
  - Change invalid weekdays into valid entries
- After clean up, left with ***180811 records, 16 useful features***

Exploratory Data Analysis: Visualizations







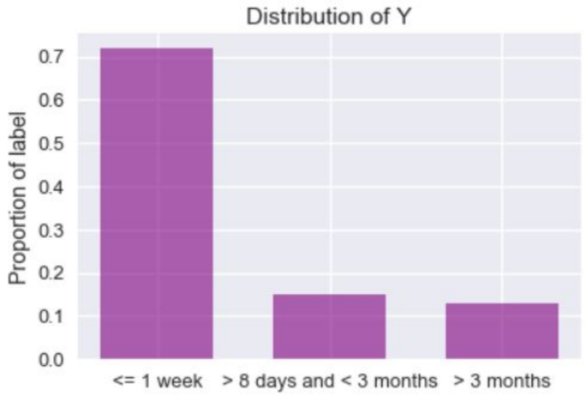


Exploratory Data Analysis: Key Findings

- Monday, the least crowded day.
- 90% of permits are OTC alterations.
  - 60%+ issued the same day.
  - 75% of issued within 5 days.
- Median time for new construction type: about 1 year 3 months
- Median of time taken for alterations: close to 6 months
- Most applications have one of the 2 Plansets (although there are 7)
- Absence of cost entry: Major cause of delay
  - ***Fill the cost field.***

Predictive Modeling

- **Target variable Y:**
  - Y = 0 if time < 8 days
  - 1 elseif time < 92 days
  - 2 else
- Why not estimate in days (regression)?
  - Too many outliers
  - Records without issue dates
- Why not just have 2 classes?
  - Too trivial, several permit types, not a true representative
- **Metrics used:** Accuracy, Recall and weighted F1 score



Training / Testing / Feature Engineering

- Data split: Train / Validation / Test in the ratio 60:20:20
- Hyperparameter tuned with 5-fold cross validation

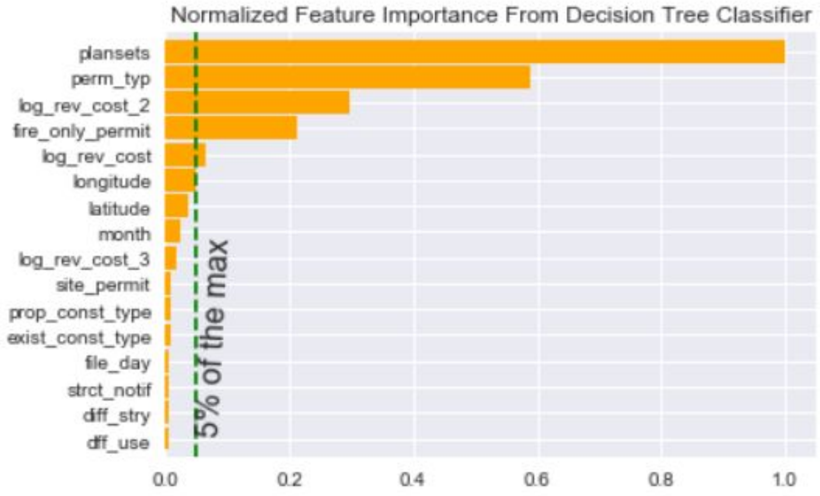
Feature Engineering

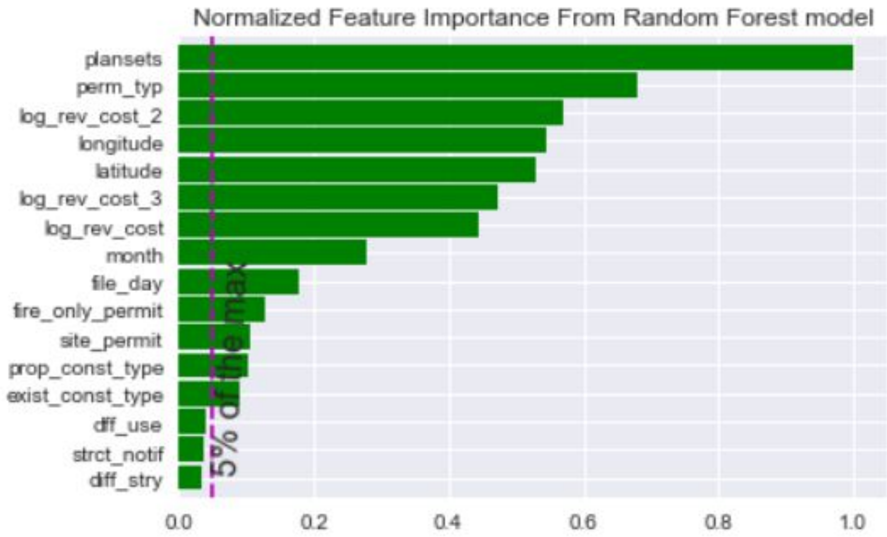
- Log of revision cost from revision cost
- Square and cube of log of revision cost
- dff_use as existing_use != proposed_use,
- diff_story as existing_use != proposed_story
- Week day and month extracted from filing date
- Location opened up as latitude and longitude

Comparisons of Models Tried

Classifier Model	Hyperparameters	Sensitivity on labels 0,1,2	F1 score
Logistic Regression	C = 0.1, class_Weight = balanced	0.82,0.75,0.67 Overall 79%	0.81
Decision Tree	max_depth=12,min_samples_leaf =6,class_weight = None	0.94,0.41,0.67 Overall 83%	0.82
Bagging	n_estimators=50	0.94,0.51,0.73 Overall 84.6%	0.84
Random Forests	min_samples_leaf=2,class_weight = balanced,n_estimators=200	0.87,0.69,0.73 Overall 83%	0.84
Gradient Boosting	min_samples_leaf=2,n_estimator s=50	0.94,0.65,0.67 Overall 82%	0.83

Feature importance





Conclusions / Future work

Conclusions

- **The Random Forest** is the chosen model. Performance on test set was 83% accuracy, no surprises.
- 5 features are the key deterministic factors: Plansets, Permit type, log_rev_cost_2, longitude, latitude
- Rest of them together add around 2% accuracy

Future Work

- Collect housing/crime data to increase features
- Undersampling, oversampling techniques to handle class imbalance

Code   Detailed Report **(References in Report)**