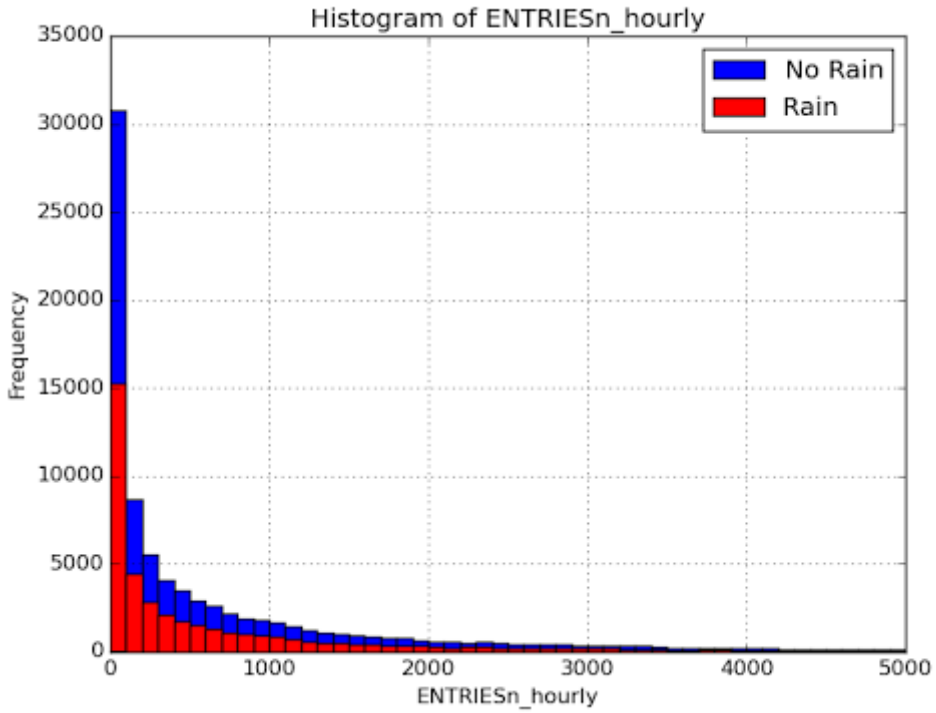
Analyzing the NYC Subway Dataset

Section 1. Statistical Test

In the beginning, I considered two kind of tests to apply:

- Welch's t-test
- Mann-Whitney u-test

I made histogram



This distribution does not seem normal so parametric test as Welch's t-test is not applicable. I proceeded with u-test which does not require that exploring population have some specific distribution. Null hypothesis H_0 is $P(x > y) = 0.5$

Further exploration discovered that

the mean of entries with rain	1105.446
the mean of entries without rain	1090.278
the Mann-Whitney U-statistic	1924409167
p-value (one-tailed)	0.0249
p-value (two-tailed)	0.0498

Critical p-value (two-tailed) is 0.05. Our value $0.0498 < 0.05$ so we can discard null hypothesis and drew conclusion that difference in numbers of entries with and without rain is statistically significance.

Section 2. Linear Regression

I have applied both linear regression with gradient descent learning algorithm and OLS from Statsmodels during optional exercise.

Most of features are numerical but there is one categorical feature that should be expanded into dummy variables, which is 'UNIT'. After substituting this feature with dummy one, all features are numerical and algorithm can be applied.

When selecting features set, I followed simple algorithm:

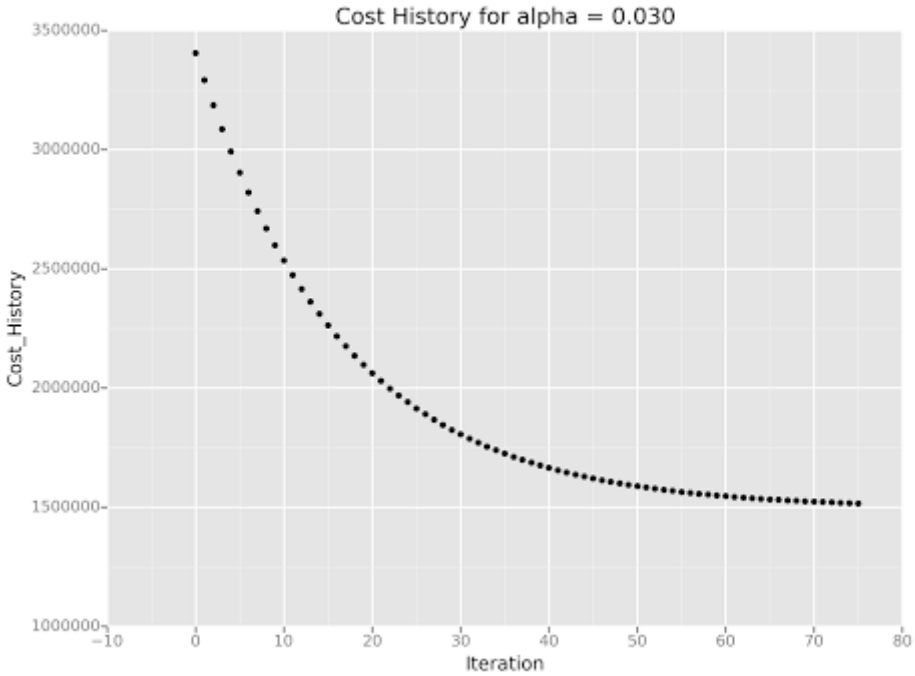
1. Selected initial narrow set of features I believed should contribute to the predictive power of my model.
2. Remove one by one features from that set.
    a. If R^2 decreased then the model became worse and I remained that feature in the set.
    b. If R^2 grew up then the model became better and I removed that feature in the set.
3. After step 2 I still did not get required accuracy. Then I continued follow the same logic with adding features one by one to my set from initial list.
    a. If after adding new feature R^2 decreased I did not add that feature to the set.
    b. If R^2 grew up then I added that feature in the set.

Finally, I got this list for linear regression:

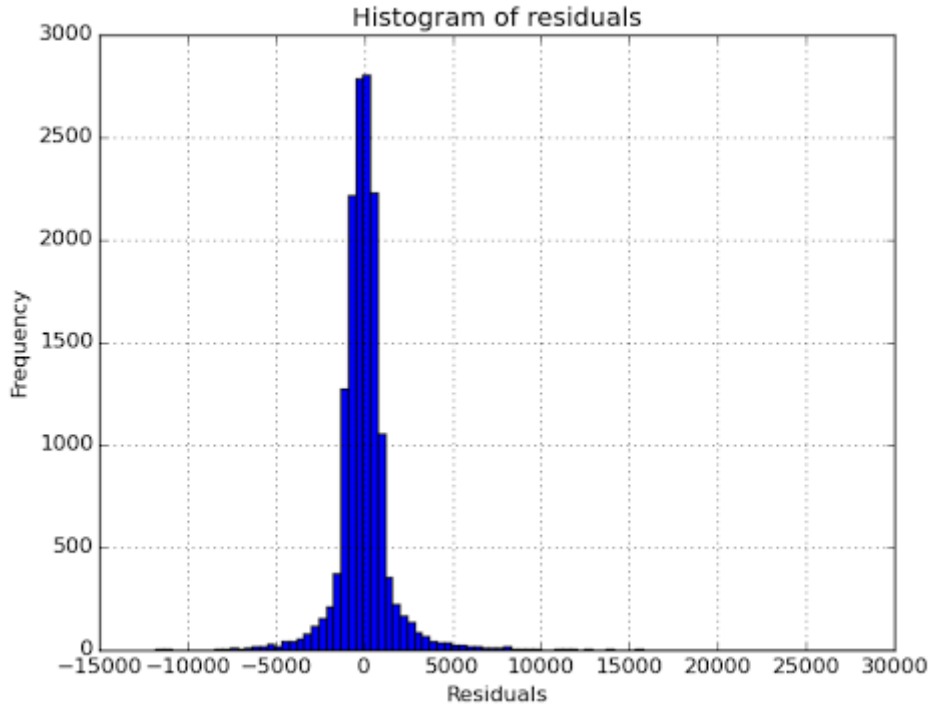
- 'rain'
- 'Hour'
- 'meantempi'
- 'maxpressurei'
- 'maxdewpti'
- 'mindewpti'
- 'minpressurei'
- 'meandewpti'
- 'meanpressurei'
- 'fog'
- 'meanwindspdi'
- 'mintempi'
- 'maxtempi'
- 'precipi'

Surprisingly it was almost all initial set. R^2 was 0.45838431946 which is quite good.

Cost history graph shows that we got almost maximum from training:



Residuals analysis also proved that our model is good:



Histograms shows that distribution is close to normal one.

Further, with OLS I have improved result up to 0.484 with following formula:

ENTRIESn_hourly ~ 1 + Hour + rain + maxpressurei + np.log(maxpressurei) + C(UNIT)

During feature selection, I followed the same approach is in case of linear regression with gradient descent. But in this case I did not try all possible options and finished selection with good enough result.

I've made comparison of both approaches results in following table.

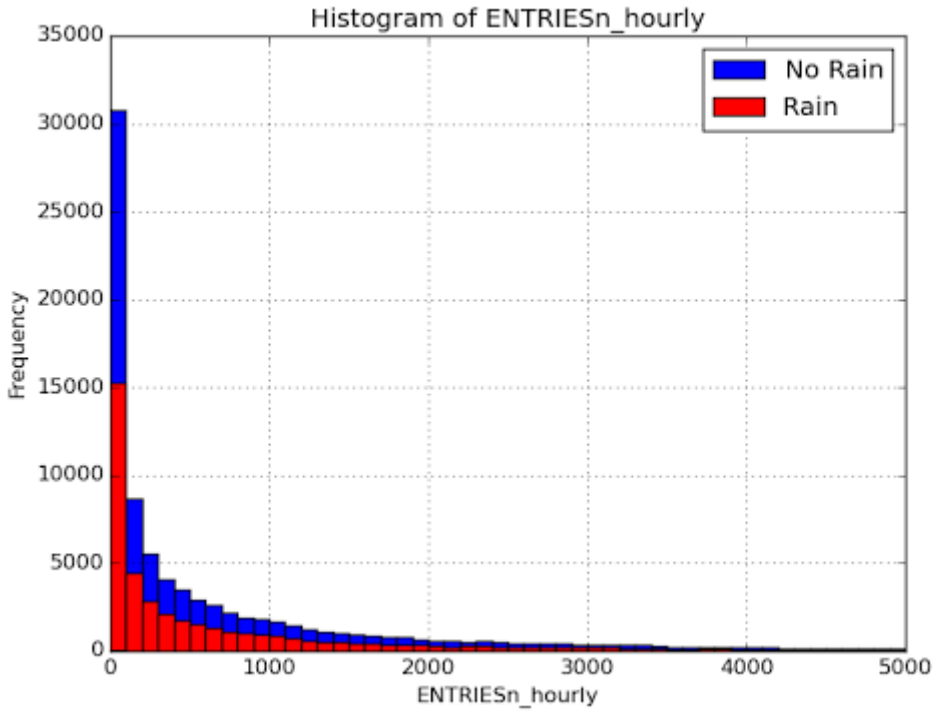
	Linear regression	OLS
R^2	0.458	0.484
Model's coefficient		
rain	-1.797	43.105
Hour	406.716	62.163
meantempi	-16.041	
maxpressurei	-10.477	-24930
maxdewpti	15.631	
mindewpti	-21.914	
minpressurei	-34.521	
meandewpti	-7.503	
meanpressurei	-2.761	
fog	43.197	
meanwindspdi	44.643	
mintempi	-46.142	
maxtempi	13.986	
precipi	-3.257	
log(maxpressurei)		748200

You can see that 'Hour' and 'maxpressurei' coefficients have the same sign (though different magnitude which is ok) so their input to model as similar while 'rain' surprisingly have different sign. Most probably it is effect of multicollinearity in linear regression model. I did not resolve it since multicollinearity does not affect R^2 value though we should keep in mind that we cannot interpret models coefficients in this case straightforward. We need to resolve this issue and then rebuild model.

I can conclude that linear regression can be used for predictions with required accuracy.

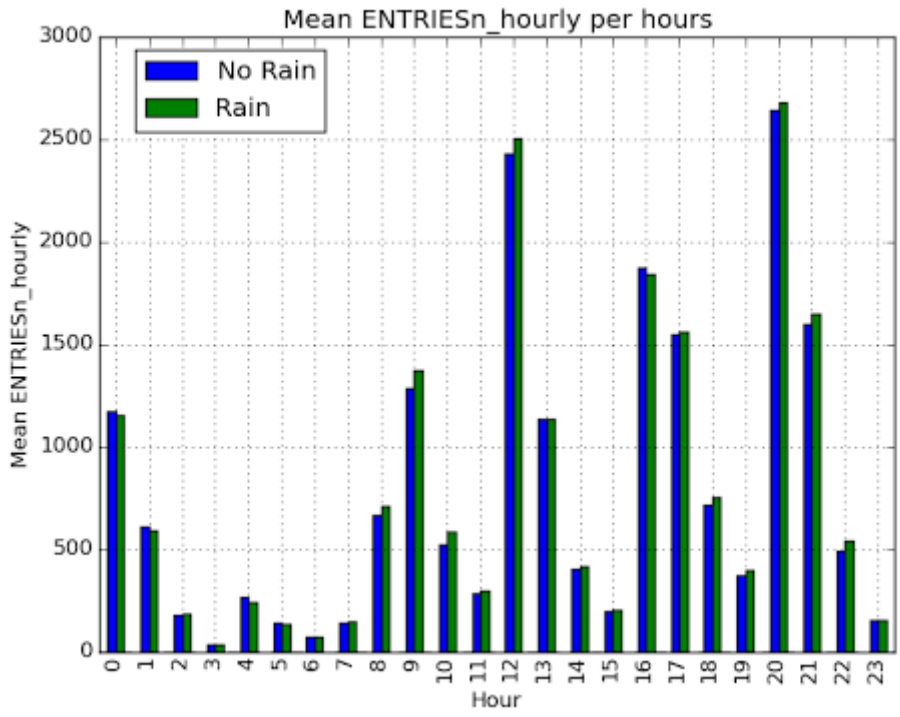
Section 3. Visualization

Histogram of ENTRIESn_hourly for rainy and no rainy days:



Histograms show that distributions for both groups follow similar distribution though set for rainy days has fewer samples than no rainy one.

I can provide a picture showing mean number of entries per hour for rainy and non-rainy days.



It shows changes of entries per hour during day. Though visual difference between data for rainy and no rainy days is not so obvious. It is hardly possible to use this graph for visual analysis to make decision.

I'd conclude that data can be represented in number of ways showing data from different points of view and it depend on aim which presentation to choose.

Section 4. Conclusion

Statistical u-test showed that difference between data for rainy and non-rainy days is statistically significant. p-value below critical value 0.05 for two-tailed test. The mean value of entries with rain 1105.446 is larger than the mean value of entries without rain 1090.278. Therefore, I can conclude that people ride more in rainy days.

Features applied during linear regression

- 'rain'
- 'Hour'
- 'meantempi'
- 'maxpressurei'
- 'maxdewpti'
- 'mindewpti'
- 'minpressurei'
- 'meandewpti'
- 'meanpressurei'
- 'fog'
- 'meanwindspdi'
- 'mintempi'
- 'maxtempi'
- 'precipi'

show what influence on ridership – hour of day and weather conditions. High coefficients values for ‘Hour’ and ‘rain’ evidence that they are important influencers.

Section 5. Reflection

I like u-test which produces quite reliable result.

Linear regression can produce moderate-to-good result in terms of R-squared but it can be affected by multicollinearity effect if too many attributes selected for modelling. This effect does not affect predictability but make hard to interpret calculated coefficients. Good side of linear regression is that it produce simple model with easy-to-describe coefficients that can be used in descriptive analysis. Coefficients corrupted by multicollinearity are not useful for it.

If I have more time I'd spend more time on feature engineering. I'd calculate cross-correlation among attributes to find correlated groups, then join them into one attribute per group which is combination of others. This would reduce dimensionality and help avoid multicollinearity.

Model validation was done against the same data set which is usually not enough. It did not guarantee that it will perform well on unseen data. Model can be overfitted to training data. If I have more time I'd change training approach to divide data set into training and validation sets and validate model against another set then one used for training.

Also I'd consider such phenomenon as seasoning. Current data are not year-wide. People's behavior could vary from month to another, from one season to another. Probably rain in winder or autumn is more unpleasant than it is in summer. I'd gather data for the whole year or several years and check it.

Section 6. References

1. http://www.itl.nist.gov/div898/handbook/pri/section2/pri24.htm
2. http://blog.minitab.com/blog/adventures-in-statistics/how-high-should-r-squared-be-in-regression-analysis
3. http://blog.minitab.com/blog/adventures-in-statistics/how-to-interpret-regression-analysis-results-p-values-and-coefficients
4. http://www.statsoft.com/Textbook/Multiple-Regression#cresidual
5. http://en.wikipedia.org/wiki/Multicollinearity