**Text Mining / Project / Instructions for Running Code**

Matthew Sudmann-Day

**Note**

This project is a subset of my master’s project which includes LDA analysis. For the text mining project, I am only using sentiment analysis. However, the code and the output you will see are the superset of the two projects. So you will see stuff about LDA that is not technically part of this project.

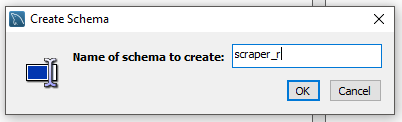
**MySQL Workbench**

From the top menu menu, select Server -> Data Import.

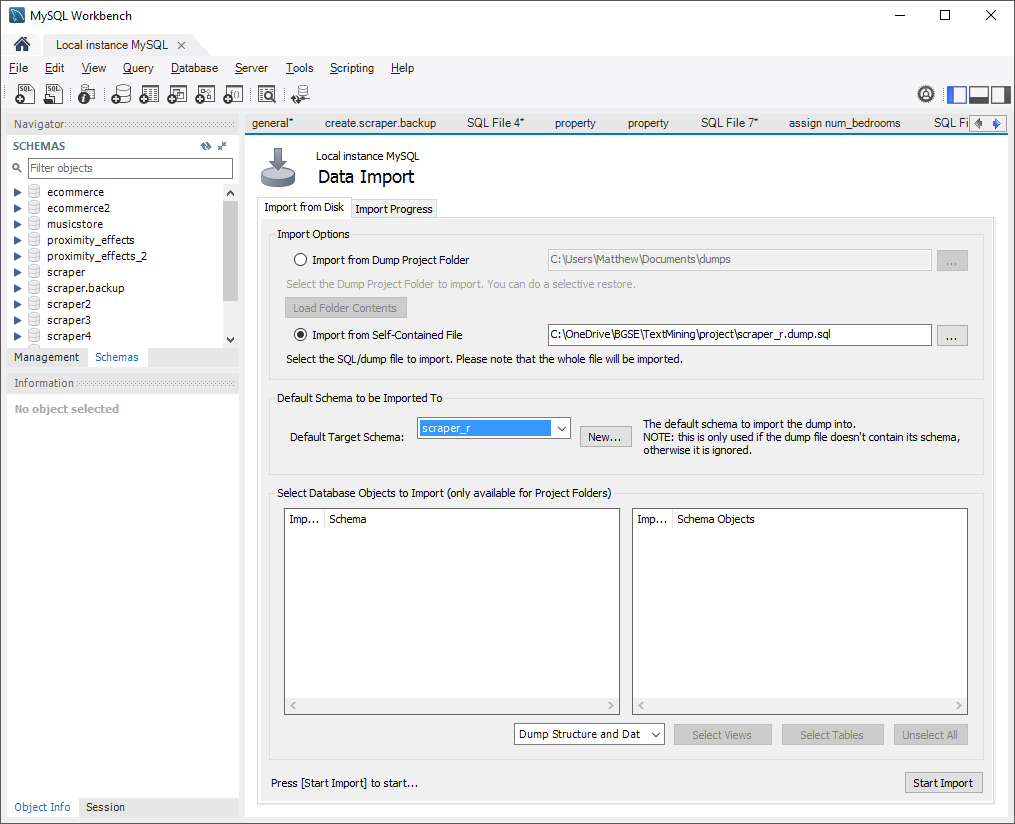
Select “Import from Self-Contained File”. Locate the dump file.

This contains a random 300 records of scraped data. It is not enough for a good analysis, but will demonstrate the process working from end to end.

For the default target schema, create a new one called ‘scraper\_r’. The ‘r’ is for Rishabh, not the programming language.



The screen should look like this:



And click “Start Import”.

**Python 2.7 (I used a 64-bit version)**

You probably only need the following libraries as these are the only ones that are directly called from the code you will run.

numpy, pickle, lda, scipy, MySQLdb, datetime, time

However, other parts of my code also reference the following:

bs4, requests, nltk, pysentiment

Open the folder ‘Python scripts’.

Open the script Database.py in an editor. Starting on line 22, you will see the following:

def \_\_init\_\_(self):

self.host = "127.0.0.1"

self.user = "root"

self.passwd = "root"

self.db = "scraper\_r"

Make sure that the connection information is correct and change it if not.

Execute the script Corpus.py. It will take one to two minutes to run. You should see output that looks like what you see below. It prompts you to ‘PRESS ENTER’ in places so that you can see what’s happened before it moves on.

Loading documents from MySQL

300 documents loaded

Loading vocabulary

2478 terms loaded

PRESS ENTER TO CONTINUE

Generating DTM

DTM dimensions: (300, 2478)

Fitting LDA model

INFO:lda:n\_documents: 242

INFO:lda:vocab\_size: 2478

INFO:lda:n\_words: 21095

INFO:lda:n\_topics: 5

INFO:lda:n\_iter: 50

WARNING:lda:all zero column in document-term matrix found

INFO:lda:<0> log likelihood: -190634

INFO:lda:<10> log likelihood: -155272

INFO:lda:<20> log likelihood: -150452

INFO:lda:<30> log likelihood: -148769

INFO:lda:<40> log likelihood: -147755

INFO:lda:<49> log likelihood: -147145

LDA model fitted

PRESS ENTER TO SEE LDA TOPICS

\*\*\* Topic 0

['avail' 'includ' 'deposit' 'hob' 'decor' 'road' 'freezer' 'central' 'must'

'fridg' 'landlord' 'may' 'basin' 'agre' 'condit' 'content' 'make'

'attract' 'month']

\*\*\* Topic 1

['apart' 'station' 'open' 'floor' 'locat' 'fulli' 'also' 'spaciou'

'develop' 'plan' 'shop' 'street' 'fit' 'high' 'recept' 'offer' 'minut'

'suit' 'space']

\*\*\* Topic 2

['fee' 'properti' 'tenanc' 'referenc' 'applic' 'pleas' 'make' 'charg'

'agreement' 'inventori' 'guidanc' 'per' 'provid' 'rent' 'payabl' 'offer'

'would' 'guarantor' 'room']

\*\*\* Topic 3

['doubl' 'floor' 'room' 'bedroom' 'rear' 'kitchen' 'glaze' 'fit' 'window'

'garden' 'front' 'shower' 'bathroom' 'loung' 'radiat' 'heat' 'entranc'

'carpet' 'x']

\*\*\* Topic 4

['bedroom' 'properti' 'bathroom' 'park' 'doubl' 'kitchen' 'view' 'locat'

'two' 'area' 'modern' 'room' 'compris' 'present' 'larg' 'access' 'close'

'within' 'garden']

Generating projections

Saving projections to MySQL

You won't see sentiment analysis here. It was

performed when the data was scraped and saved

with all the other metadata.

See the function analyseSentiment() in Property.py.

FINISHED - PRESS ANY KEY TO EXIT

The thread 'MainThread' (0x1498) has exited with code 0 (0x0).

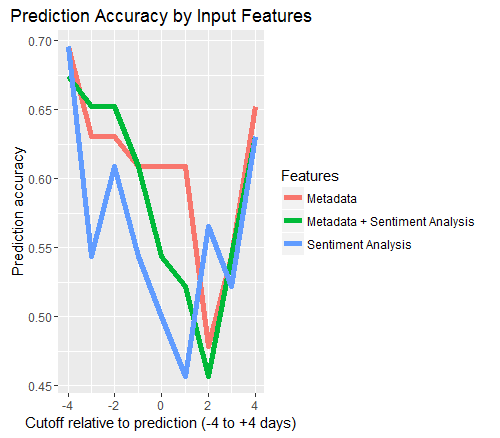
**R**

Now that the database has been updated with the LDA topic values from the Python script, you can run the R script which will load the data and analyze it.

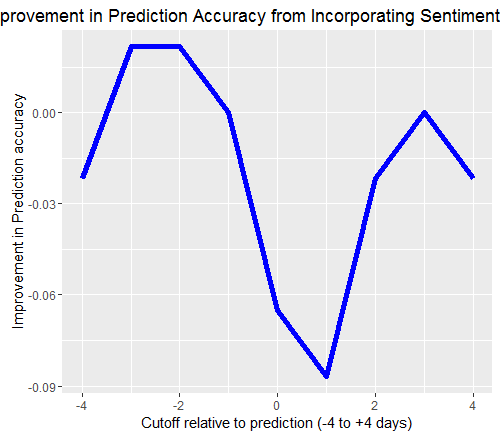
Open the script analysis.R. On line 9, make sure your MySQL credentials are correct.

Because a few files will be created when you run this script, if you are using R Studio, select the menu “Session -> Set Working Directory -> To Source File Location”.

Run the script. It only takes a few seconds. At the end of the script two plots will be created looking something like this:



and this:



Data will be written to improvement.csv:

|  |  |  |
| --- | --- | --- |
|  | offset | diff |
| 2 | -4 | -0.02174 |
| 3 | -3 | 0.021739 |
| 4 | -2 | 0.021739 |
| 5 | -1 | 0 |
| 6 | 0 | -0.06522 |
| 7 | 1 | -0.08696 |
| 8 | 2 | -0.02174 |
| 9 | 3 | 0 |
| 10 | 4 | -0.02174 |

And to mean\_decrease\_gini.csv:

|  |  |
| --- | --- |
|  | MeanDecreaseGini |
| is\_share | 0.351215 |
| num\_bedrooms | 4.053616 |
| availability\_lag | 7.21148 |
| monthly\_price | 10.7636 |
| furnishing | 1.159543 |
| is\_london | 1.532944 |
| is\_other\_city | 1.075325 |
| sent\_pol | 8.62637 |
| sent\_subj | 12.33288 |