Logistic regression classifier for Indian law cases with 900 known and 100 unknown data

Tony Nguyen

6/26/2020

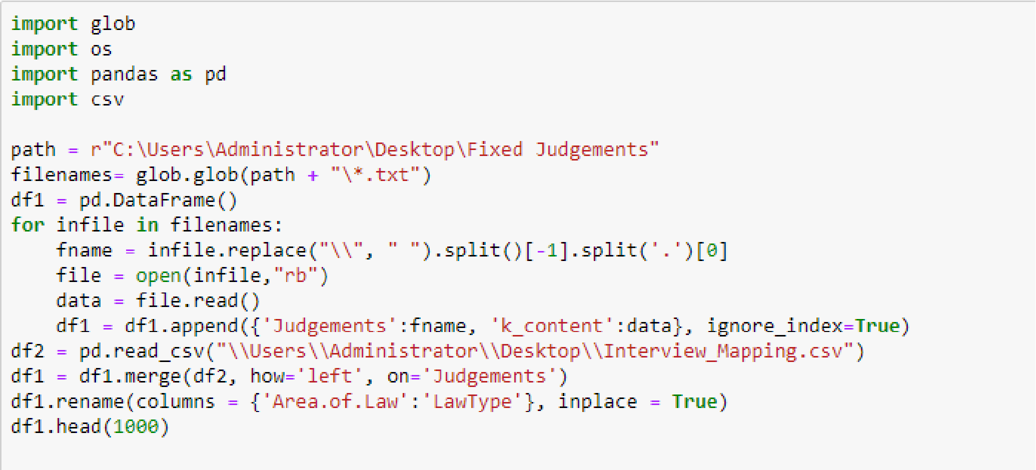
Table of content

1 Abstract

The goal of the project is to build a classifier that can classify 100 given unknown Indian law. We first generate a dataframe that can hold the data we were given since both are not the same file types.To build the classifier we will need to clean the data. Removing words that are not necessary is important since machines do not build base off context rather build patterns based of certain features. After cleaning the data, we use a logistic regression classifier to train and test the data. When testing we improve the data by adding more features and balance out certain topics so that the machine can value less on one and more on another. A confusion matrix is generated to show how well the log reg classfier worked and what it mostly think the data is. The classifier accuracy was orginally at 40 percent but adding features and cleaning the data better help improve it to 90 percent.

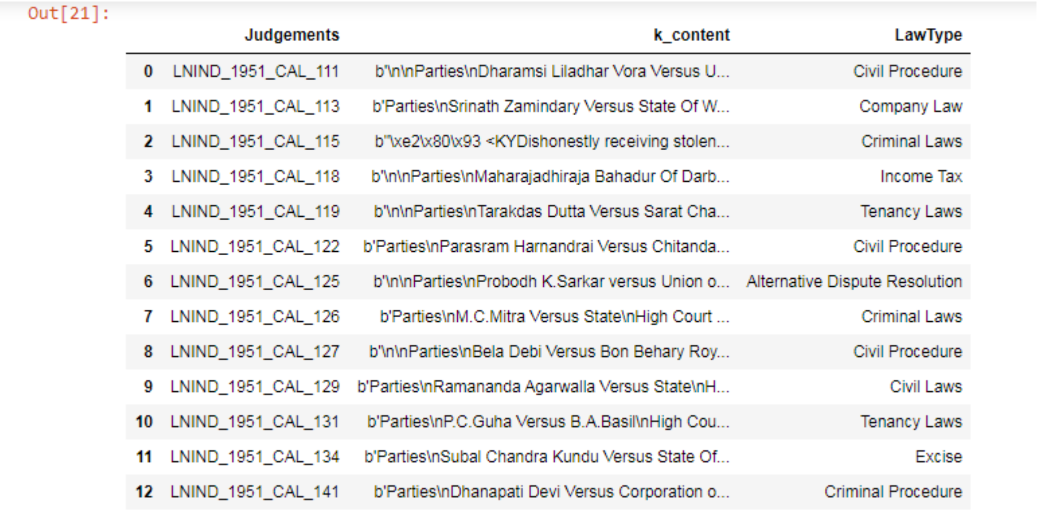
2 Generating a dataframe with given data

The first step would be to build a dataframe that can display the data that can be easily visualize and understand. What we want to know is the type of Judgements given, what the content contains, and the type of law this is. In figure 1, we created an alogorithm that combines two different files together(one given as a csv the other text files). It will display a dataframe with three columns: judgements, kcontent, and lawtype, this is what we need as we will be using this to analyze what is important in the content for our classifier.The reason why we are using glob in this project is because we are given two different data type: one is given as a csv file, the other as a txt. The txt files contain multiple files not just one. The end: df1.head() will display the dataframe up to the number you put in the (),in figure 2 we will only display the first 13 as 1000 would take up a lot of space as a figure.



```
import glob
import os
import pandas as pd
import csv

path = r"C:\Users\Administrator\Desktop\Fixed Judgements"
filenames= glob.glob(path + "\*.txt")
df1 = pd.DataFrame()
for infile in filenames:
    fname = infile.replace("\\", " ").split()[-1].split('.')[0]
    file = open(infile,"rb")
    data = file.read()
    df1 = df1.append({'Judgements':fname, 'k_content':data}, ignore_index=True)
df2 = pd.read_csv("\\Users\\Administrator\\Desktop\\Interview_Mapping.csv")
df1 = df1.merge(df2, how='left', on='Judgements')
df1.rename(columns = {'Area.of.Law':'LawType'}, inplace = True)
df1.head(1000)
```

Figure 1: using glob to create a datafile



Out[21]:

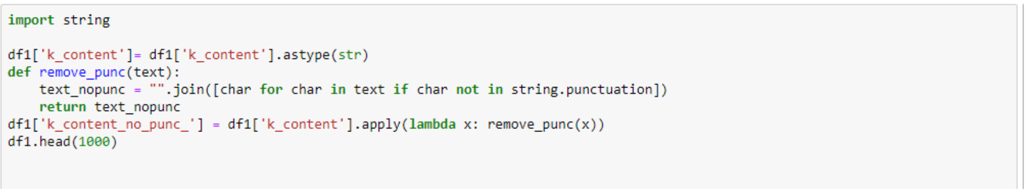
	Judgements	k_content	LawType
0	LNIND_1951_CAL_111	b"\n\nParties\nDharamsi Liladhar Vora Versus U...	Civil Procedure
1	LNIND_1951_CAL_113	b'Parties\nSrinath Zamindary Versus State Of W...	Company Law
2	LNIND_1951_CAL_115	b"\xe2\x80\x93 <KYDishonestly receiving stolen...	Criminal Laws
3	LNIND_1951_CAL_118	b"\n\nParties\nMaharajadhiraja Bahadur Of Darb...	Income Tax
4	LNIND_1951_CAL_119	b"\n\nParties\nTarakdas Dutta Versus Sarat Cha...	Tenancy Laws
5	LNIND_1951_CAL_122	b'Parties\nParasram Harnandrai Versus Chitanda...	Civil Procedure
6	LNIND_1951_CAL_125	b"\n\nParties\nProbodh K.Sarkar versus Union o...	Alternative Dispute Resolution
7	LNIND_1951_CAL_126	b'Parties\nM.C.Mitra Versus State\nHigh Court ...	Criminal Laws
8	LNIND_1951_CAL_127	b"\n\nParties\nBela Debi Versus Bon Behary Roy...	Civil Procedure
9	LNIND_1951_CAL_129	b'Parties\nRamananda Agarwalla Versus State\nH...	Civil Laws
10	LNIND_1951_CAL_131	b'Parties\nP.C.Guha Versus B.A.Basil\nHigh Cou...	Tenancy Laws
11	LNIND_1951_CAL_134	b'Parties\nSubal Chandra Kundu Versus State Of...	Excise
12	LNIND_1951_CAL_141	b'Parties\nDhanapati Devi Versus Corporation o...	Criminal Procedure

Figure 2: output of figure 1 displaying the first 13

Figure 2 displays the output from the code. This is what we needed as we want to be able to visually see and read the judgements, content, and law type. From here what we need to do is figure out what parts of the content is not necessary or a big influence to our classifier.

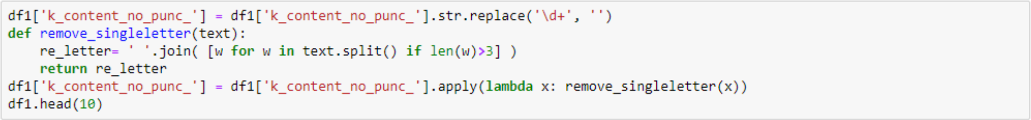
2.1 Cleaning text data

The next part once we build our dataframe is to clean the data. We want to remove punctuations, single letters, stopwords, nonenglish words. We first will need to change the dataframe to strings since this data is in bytes. After converting to strings we remove the punctuation shown in figure 3. The point of removing punctuation is because computers don't recognize nor value punctuations. The next part of cleaning would be to remove single letters to lower the number of features that aren't valuable to us. Having letters such as r, s or etc can lower the classifier's accuracy.Figure 4 shows that we want to remove single letters with length less than 3 meaning words with less than 3 letters. After removing single letters, we now want to remove nonenglish words and words that are too repetitive. Nonenglish words should be remove in this case as they have no value in our classifier, if this classifier was meant to distinguish nonenglish words then we should keep, but this is meant for classifying law types.



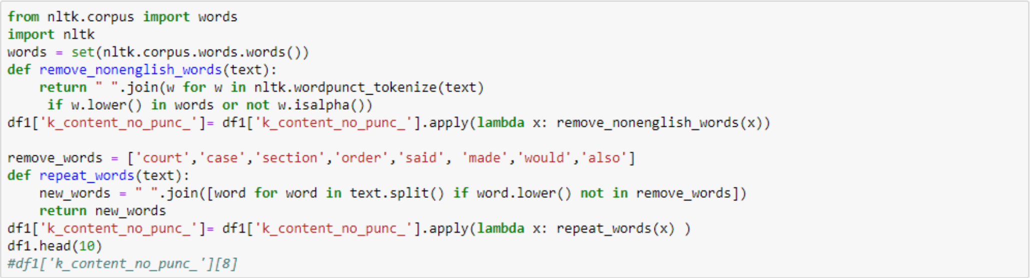
```
import string

df1['k_content']= df1['k_content'].astype(str)
def remove_punc(text):
    text_nopunc = "".join([char for char in text if char not in string.punctuation])
    return text_nopunc
df1['k_content_no_punc_'] = df1['k_content'].apply(lambda x: remove_punc(x))
df1.head(1000)
```

Figure 3: code to converting contents to string and removing punctuations



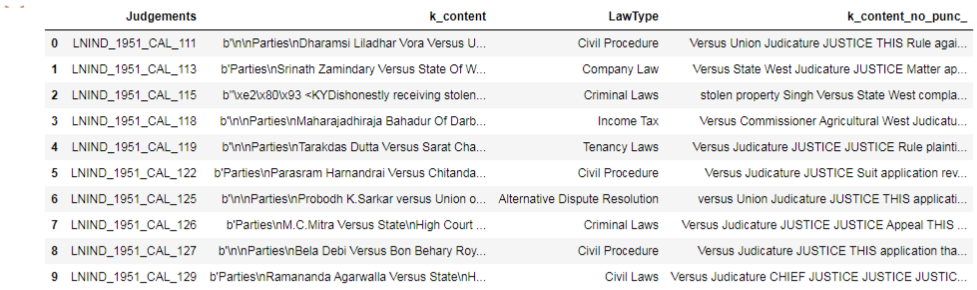
```
df1['k_content_no_punc_'] = df1['k_content_no_punc_'].str.replace('\d+', '')
def remove_singleletter(text):
    re_letter= ' '.join( [w for w in text.split() if len(w)>3] )
    return re_letter
df1['k_content_no_punc_'] = df1['k_content_no_punc_'].apply(lambda x: remove_singleletter(x))
df1.head(10)
```

Figure 4: removing single letters



```python
from nltk.corpus import words
import nltk
words = set(nltk.corpus.words.words())
def remove_nonenglish_words(text):
    return " ".join(w for w in nltk.wordpunct_tokenize(text)
     if w.lower() in words or not w.isalpha())
df1['k_content_no_punc_']= df1['k_content_no_punc_'].apply(lambda x: remove_nonenglish_words(x))

remove_words = ['court','case','section','order','said', 'made','would','also']
def repeat_words(text):
    new_words = " ".join([word for word in text.split() if word.lower() not in remove_words])
    return new_words
df1['k_content_no_punc_']= df1['k_content_no_punc_'].apply(lambda x: repeat_words(x) )
df1.head(10)
#df1['k_content_no_punc_'][8]
```

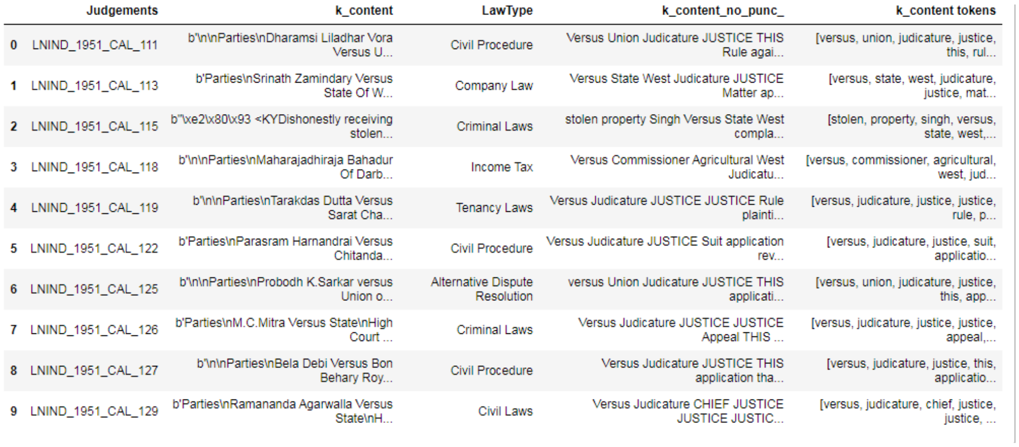
Figure 5: removing nonenglish and specific words



	Judgements	k_content	LawType	k_content_no_punc_
0	LNIND_1951_CAL_111	b'\n\nParties\nDharamsi Liladhar Vora Versus U...	Civil Procedure	Versus Union Judicature JUSTICE THIS Rule agai...
1	LNIND_1951_CAL_113	b'Parties\nSrinath Zamindary Versus State Of W...	Company Law	Versus State West Judicature JUSTICE Matter ap...
2	LNIND_1951_CAL_115	b"\xe2\x80\x93 <KYDishonestly receiving stolen...	Criminal Laws	stolen property Singh Versus State West compla...
3	LNIND_1951_CAL_118	b'\n\nParties\nMaharajadhiraja Bahadur Of Darb...	Income Tax	Versus Commissioner Agricultural West Judicatu...
4	LNIND_1951_CAL_119	b'\n\nParties\nTarakdas Dutta Versus Sarat Cha...	Tenancy Laws	Versus Judicature JUSTICE JUSTICE Rule plainti...
5	LNIND_1951_CAL_122	b'Parties\nParasram Harnandrai Versus Chitanda...	Civil Procedure	Versus Judicature JUSTICE Suit application rev...
6	LNIND_1951_CAL_125	b'\n\nParties\nProbodh K.Sarkar versus Union o...	Alternative Dispute Resolution	versus Union Judicature JUSTICE THIS applicati...
7	LNIND_1951_CAL_126	b'Parties\nM.C.Mitra Versus State\nHigh Court ...	Criminal Laws	Versus Judicature JUSTICE JUSTICE Appeal THIS ...
8	LNIND_1951_CAL_127	b'\n\nParties\nBela Debi Versus Bon Behary Roy...	Civil Procedure	Versus Judicature JUSTICE THIS application tha...
9	LNIND_1951_CAL_129	b'Parties\nRamananda Agarwalla Versus State\nH...	Civil Laws	Versus Judicature CHIEF JUSTICE JUSTICE JUSTIC...

Figure 6: after cleaning data the output of the first 10 law data

2.2 Tokenizing the content



	Judgements	k_content	LawType	k_content_no_punc_	k_content tokens
0	LNIND_1951_CAL_111	b'\n\nParties\nDharamsi Liladhar Vora Versus U...	Civil Procedure	Versus Union Judicature JUSTICE THIS Rule agai...	[versus, union, judicature, justice, this, rul...
1	LNIND_1951_CAL_113	b'Parties\nSrinath Zamindary Versus State Of W...	Company Law	Versus State West Judicature JUSTICE Matter ap...	[versus, state, west, judicature, justice, mat...
2	LNIND_1951_CAL_115	b"\xe2\x80\x93 <KYDishonestly receiving stolen...	Criminal Laws	stolen property Singh Versus State West compla...	[stolen, property, singh, versus, state, west,...
3	LNIND_1951_CAL_118	b'\n\nParties\nMaharajadhiraja Bahadur Of Darb...	Income Tax	Versus Commissioner Agricultural West Judicatu...	[versus, commissioner, agricultural, west, jud...
4	LNIND_1951_CAL_119	b'\n\nParties\nTarakdas Dutta Versus Sarat Cha...	Tenancy Laws	Versus Judicature JUSTICE JUSTICE Rule plainti...	[versus, judicature, justice, justice, rule, p...
5	LNIND_1951_CAL_122	b'Parties\nParasram Harnandrai Versus Chitanda...	Civil Procedure	Versus Judicature JUSTICE Suit application rev...	[versus, judicature, justice, suit, applicatio...
6	LNIND_1951_CAL_125	b'\n\nParties\nProbodh K.Sarkar versus Union o...	Alternative Dispute Resolution	versus Union Judicature JUSTICE THIS applicati...	[versus, union, judicature, justice, this, app...
7	LNIND_1951_CAL_126	b'Parties\nM.C.Mitra Versus State\nHigh Court ...	Criminal Laws	Versus Judicature JUSTICE JUSTICE Appeal THIS ...	[versus, judicature, justice, justice, appeal,...
8	LNIND_1951_CAL_127	b'\n\nParties\nBela Debi Versus Bon Behary Roy...	Civil Procedure	Versus Judicature JUSTICE THIS application tha...	[versus, judicature, justice, this, applicatio...
9	LNIND_1951_CAL_129	b'Parties\nRamananda Agarwalla Versus State\nH...	Civil Laws	Versus Judicature CHIEF JUSTICE JUSTICE JUSTIC...	[versus, judicature, chief, justice, justice, ...

Figure 7: tokenizing data

3 Building the logistic regression classifier

3.1 Improving training and testing score

3.2 Visually seeing the classifier with confusion matrix

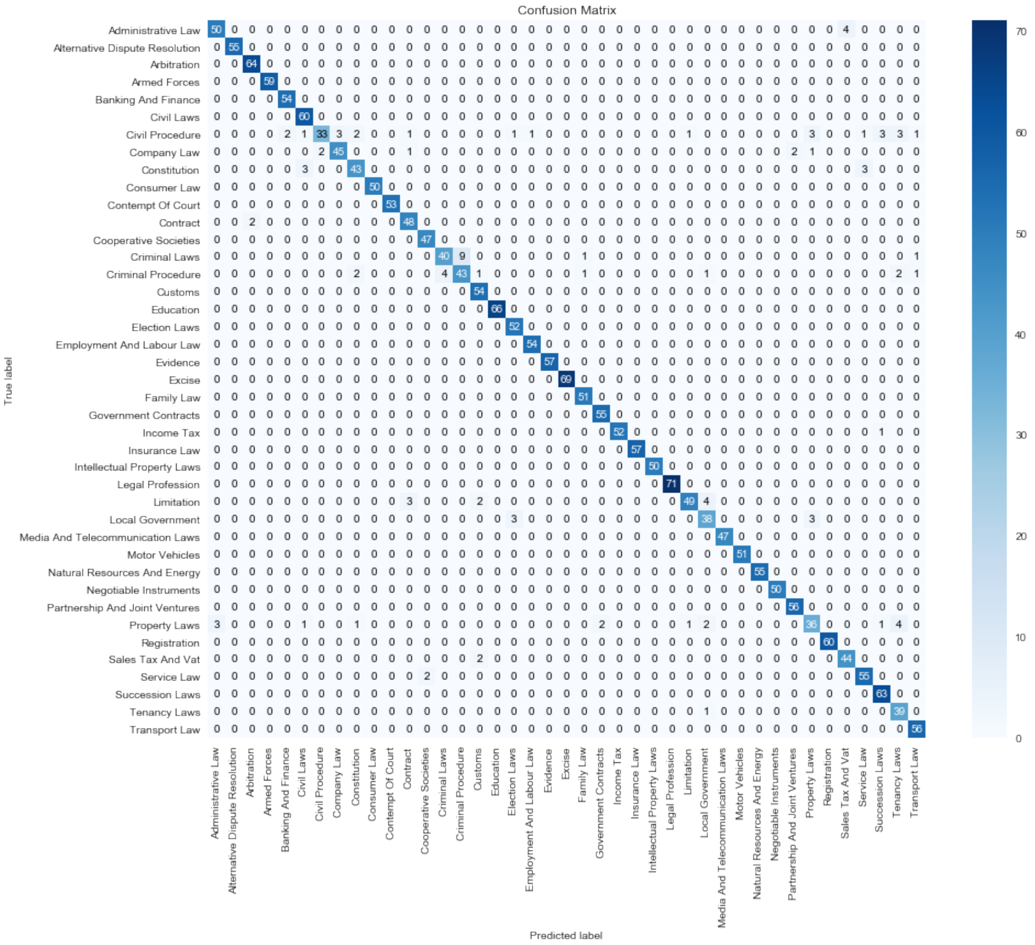


Figure 8: Diagonal is the true/correct values with labels Actual on y axis, Predicted on x axis