NLP PROJECT

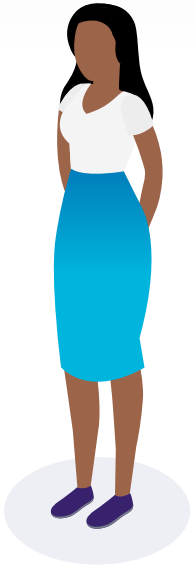
TEXT CLASSIFICATION FOR CAD FROM DISCHARGE SUMMARIES

Jill Han

12/02/2019

Outline

- Background
- Methods
- Results
- Discussions

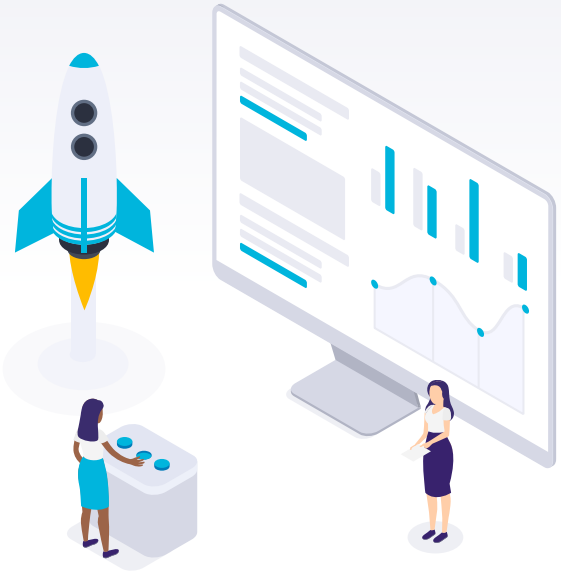


1 Background

Coronary artery disease (CAD)

Natural Language Processing (NLP)

Text Classification



Background

CAD

The most common type of heart disease.

Special considerations for cancer patients with CAD.

NLP

EHR

Build models using NLP with EHR.

Transfer learning.

Text Classification

Identify patients with CAD from discharge summaries.

Improve work flow and treatment in cancer patients with CAD.

2 Methods

NLTK

Bag of Words, TF-IDF, Ngram TF-IDF & XGBoost

Fastai

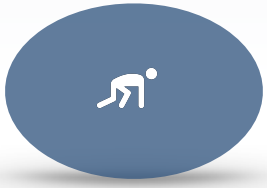
Pipeline

- EDA
- Tokenization
- Combined text and labels

DATA PRECESSING

- Recall
- Precision
- Confusion metrix

PREDICTION



DATA COLLECTION

- Applied for 2010 i2b2 dataset: https://portal.dbmi.hms.harvard.edu/projects/n2c2-nlp/
- Annotated files with 2 classes:

  0 = no CAD
  1 = CAD

MODELING

- Feature Extraction
- XGBoost
- Fastai

2010 i2b2 dataset

	0	1	T
Train	333	104	437
Test	192	64	256
Total	525	168	693

annotated_for_test.txt

```
ID,HF,CAD,Description
'0001,1,1, CAD(2)/CHF(3)
0002,0,0
0005,0,0
0006,0,1, cardiaccath(1)/CAD(2)
0009,0,0, AF(1)
0010,0,1, left coronary artery stenosis(1)
0013,0,0
0014,0,0
0017,0,0
0018,0,0, AF(2)
0021,0,1, left-sided infiltrate versus effusion(1)/CAD status post CABG/MVR(2)
0022,0,0
```

011080908_SC.txt

```
011080908 SC
01459638
2838870
3/6/2003 12:00:00 AM
Discharge Summary
Signed
DIS
Admission Date :
Report Status :
Signed
Discharge Date :
03/06/2003
Date of Discharge :
03/06/2003
ATTENDING :
BLOCK AISTAKEFLICHSKI MD
The patient was a 78 year old male with a history of COPD and IPF , chronically vented on trach with
prolonged Lilum stay from September to December who presents from his nursing home with hypoxia and
hypertension .
HISTORY OF PRESENT ILLNESS :
The patient was most recently admitted to Middso Memorial Hospital &apos;s on October 2 with a
myocardial infarction .
He was taken to Cath where he had an IV , bypass lalema to an LAD .
His postop course was complicated by difficulty weaning off the vent , multiple aspirations .
He was given a trach and a bag .
He was ultimately DC &apos;d to rehab and has had most recently been at Cootend Care .
On February 11 , the patient was noted to be tachypneic , four days with pulmonary edema .
His eyes and nose at that time were 1.6 in and 400 out .
He was given an additional dose of 40 mg of Lasix and ABG at that time revealed a pH of 7.47 , PCO2 of
63 , 81 and 96% .
At 10 p.m. that evening , the patient was found to have a blood pressure of 70/40 and a heart rate of
80 with an O2 sat of 95% .
His hematocrit was 23 .
He was given 2 liters of normal saline , vancomycin and levo .
His blood pressure did not respond so he was transferred to Ca Valley Hospital .
It is unclear from the progress notes , but at some point the patient was found to have an O2 sat of
59% , which increased to 93% with suction .
PAST MEDICAL HISTORY :
The patient &apos;s past medical history is significant for A. fib , coronary artery disease and status
post one vessel CABG in September of this year , VT status post AICD placement , peripheral vascular
disease , restrictive lung disease with an FEV1 of 45% at 1.6 , FVC2 of 2 and 47% and a total lung
capacity of 3.7 liters .
```

Create dataset

```
trainfile_lst = [os.path.basename(file) for file in glob.iglob(train_path+'/*.txt')]

testfile_lst = [os.path.basename(file) for file in glob.iglob(test_path+'/*.txt')]
```

```
In [5]: trainfile_lst

Out[5]: ['284487129.txt',
         '331144840.txt',
         'record-51.txt',
         'record-45.txt',
         '522011500_ELMVH.txt',
         '641987347_RWH.txt',
         '198175916.txt',
         'record-107.txt',
         '213763231.txt',
         '719127931.txt',
         '558603822.txt',
```

```
train_filename_lst = [lst.split('.')[0] for lst in trainfile_lst]
train_filename_lst = [element.replace("-", "_") for element in train_filename_lst]
test_lst = [filename.split('.')[0] for filename in testfile_lst]
```

eg: '044687343 ELMVH' to '044687343_ELMVH'

eg: ’record-121’ to 'record_121'

https://www.geeksforgeeks.org/python-os-path-basename-method/

```
In [28]: df_test_tag.head()
```

Out[28]:

	ID	HF	CAD	Description
0	0001	1	1	CAD(2)/CHF(3)
1	0002	0	0	NaN
2	0005	0	0	NaN
3	0006	0	1	cardiaccath(1)/CAD(2)
4	0009	0	0	AF(1)

```
In [31]: df_train_tag.head()
```

Out[31]:

	ID	HF	CAD	Description
0	record_13	0	1	CAD(1)
1	record_14	0	0	NaN
2	record_15	1	1	CAD(1) / HF(2)
3	record_16	0	1	CAD(1)
4	record_17	0	0	Rule out myocardial infarction

Concat files content with ID

```
def concat_df(df,filenamelist,col_list,path):
    """
    Concat text and CAD columns with same ID
    Return the merged dataframe.
    """
    df1 = pd.DataFrame(df)[col_list]
    id_lst = []
    text_lst = []

    def read_text(text):
        with open(text, "r") as file:
            raw = file.readlines()
        return raw

    for file in filenamelist:
        id_lst.append(file.split('.')[0])
        text = read_text(path+file)
        text_lst.append(text)

    id_lst = [file.replace("-","_") for file in id_lst]
    lst_of_tuples = list(zip(id_lst, text_lst))
    df2 = pd.DataFrame(lst_of_tuples, columns = ['ID','text'])
    merge_df = pd.merge(left = df1,right=df2, left_on='ID', right_on='ID')
    return merge_df
```

```
merge_train.head()
```

	ID	CAD	text
0	record_13	1	[Admission Date :\n, 2018-10-25\n, Discharge D...
1	record_14	0	[Admission Date :\n, 2011-03-10\n, Discharge D...
2	record_15	1	[Admission Date :\n, Discharge Date :\n, 2014-...
3	record_16	1	[Admission Date :\n, 2015-10-28\n, Discharge D...
4	record_17	0	[Admission Date:\n, 2011-02-08\n, Discharge Da...

```
merge_test.head()
```

	ID	CAD	text
0	0001	1	[Admission Date :\n, 2012-10-31\n, Discharge ...
1	0002	0	[004668411\n, CTMC\n, 68299235\n, 763052\n, 9/...
2	0005	0	[Admission Date :\n, 2016-12-24\n, Discharge ...
3	0006	1	[006544894 \n, NVH \n, 65104826 \n, 1/2/2004 1...
4	0009	0	[Admission Date :\n, 2015-10-10\n, Discharge ...

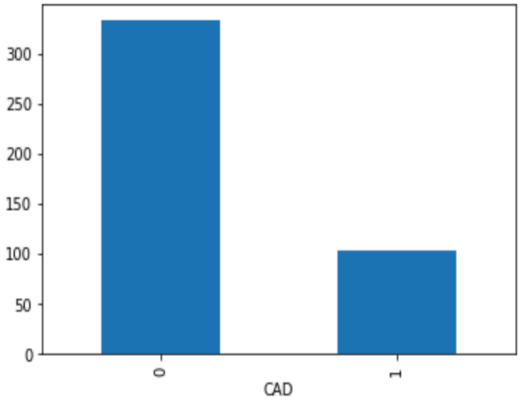
EDA

```
Training.CAD.value_counts()
```

```
0    333
1    104
Name: CAD, dtype: int64
```

```
Training.groupby("CAD")['ID'].count().plot.bar()
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x1a295804d0>
```




```
Training.isnull().values.any(), Training.isnull().values.sum()
```

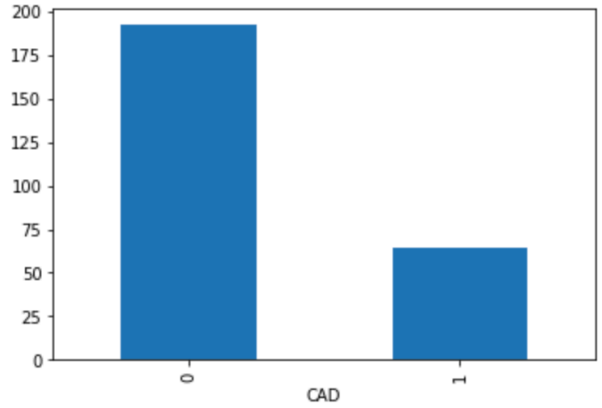
```
(False, 0)
```

```
Test.CAD.value_counts()
```

```
0    192
1     64
Name: CAD, dtype: int64
```

```
Test.groupby('CAD')['ID'].count().plot.bar()
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x1a22939990>
```




```
Test.isnull().values.any(), Test.isnull().values.sum()
```

```
(False, 0)
```

Data Processing

```python
def cleanText(sentence):
    """
    Remove special characters and numbers.
    change all letters into lowercase.

    """
    return re.sub('[^a-zA-Z]+',' ', sentence).lower()
```

1

```python
def normalize_word(sentence):
    """
    Remove stop words, stem and lemmatize words.
    Return a new words list.
    """
    stop_words = set(stopwords.words('english'))
    ps = PorterStemmer()
    word_tokens = word_tokenize(sentence)

    # Remove stop words
    filtered_words = [w for w in word_tokens if w not in stop_words]

    # Stem
    stem_words = [ps.stem(w) for w in filtered_words]

    # Lemmatize
    lemma_words = [WordNetLemmatizer().lemmatize(w) for w in stem_words]

    # Create a list having the words whose length are larger than 1.
    words = [word for word in lemma_words if len(word)>1]
    return words
```

2

```python
def token_words(lst):
    """
    Return a list of word tokens from text.
    """
    new_lst = []
    for sentence in lst:
        # Remove words of categories
        if ":" not in sentence:
            clean_sen = cleanText(sentence)
            if clean_sen != ' ':
                new_sen = normalize_word(clean_sen)
                if new_sen:
                    new_lst+=new_sen
    return new_lst
```

3

https://towardsdatascience.com/sentiment-analysis-with-python-part-2-4f71e7bde59a

Data Preparation

```
def prepared_data(df, df_dir, filename):
    """
    Create columns "word_tokens","clean_text".
    Shuffle the indexes.
    Write a csv file used for model to the target path.
    """
    df['word_tokens'] = df.apply(lambda x: token_words(x['text']),axis=1)
    df['clean_text'] = df.apply(lambda x: " ".join(x['word_tokens']),axis=1)
    df_new = shuffle(df, random_state=42)
    df_new.reset_index(drop=True)
    df_new.to_csv(os.path.join(df_dir,filename), encoding='utf-8', index = None,header='true')
```

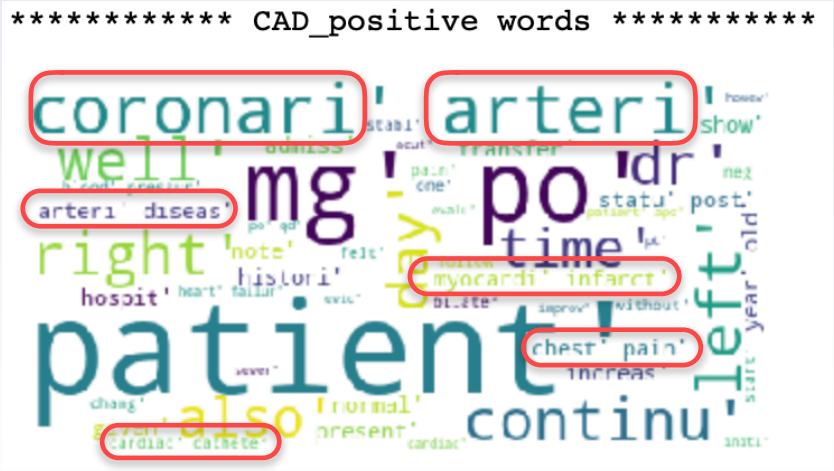
```
Training.head()
```

	ID	CAD	text	word_tokens	clean_text
0	857888116	1	['857888116\n', 'CTMC\n', '36320247\n', '66801...	['ctmc', 'leukemia', 'pnuemoia', 'bone', 'marr...	ctmc leukemia pnuemoia bone marrow transplant ...
1	095889687_WGH	0	['095889687 WGH\n', '9786774\n', '224527\n', '...	['wgh', 'ed', 'discharg', 'summari', 'unsign',...	wgh ed discharg summari unsign di unsign ed di...
2	428878172_ELMVH	0	['428878172 ELMVH\n', '77066399\n', '833854\n'...	['elmvh', 'vagin', 'birth', 'di', 'discharg', ...	elmvh vagin birth di discharg order liepkol zo...
3	511454195	0	['511454195\n', 'HLGMC\n', '0150441\n', '60106...	['hlgmc', 'discharg', 'summari', 'sign', 'di',...	hlgmc discharg summari sign di sign fongfyfeno...
4	601874915	1	['601874915\n', 'TGCHO\n', '7350631\n', '48597...	['tgcho', 'discharg', 'summari', 'unsign', 'di...	tgcho discharg summari unsign di unsign right ...

```python
def visulization_data(df):
    """
    Make a wordcloud visualization of the data
    """
    df_pos = df.loc[df['CAD'] == 1]
    df_neg = df.loc[df['CAD'] == 0]
    combined_text_pos = " ".join([word for word in df_pos['word_tokens'].tolist()])
    combined_text_neg = " ".join([word for word in df_neg['word_tokens'].tolist()])

    wc = WordCloud(background_color='white', max_words=50,
        # update stopwords to include common words like film and movie
              stopwords = STOPWORDS)

    print ("************ CAD_positive words ***********")
    plt.imshow(wc.generate(combined_text_pos))
    plt.axis('off')
    plt.show()

    print ("************ CAD_negtive words ************")
    plt.imshow(wc.generate(combined_text_neg))
    plt.axis('off')
    plt.show()
```

https://www.datacamp.com/community/tutorials/wordcloud-python

********** CAD_positive words **********



********** CAD_negtive words **********

Visualization of words in training dataset

Feature extraction

```
def extract_features(train_df, test_df, mode, vocabulary_size):
    """
    Use different mode to create train and test arrays for XGBoost model.
    Create vocabulary dictionary to extract important features in the following steps.
    Return arrays and vocabulary dictionary.
    """
    if mode == "BOW":
        bow_vector = CountVectorizer(analyzer='word', token_pattern=r'\w{1,}', max_features=vocabulary_size)
        bow_vector.fit(train_df['clean_text'].unique())
        features_train = bow_vector.transform(train_df['clean_text'].values).toarray()
        features_test = bow_vector.transform(test_df['clean_text'].values).toarray()
        vocab_dic = bow_vector.vocabulary_

    elif mode == "TFIDF":
        tfidf_vector = TfidfVectorizer(analyzer='word', token_pattern=r'\w{1,}', max_features=vocabulary_size)
        tfidf_vector.fit(train_df['clean_text'].unique())
        features_train = tfidf_vector.transform(train_df['clean_text'].values).toarray()
        features_test = tfidf_vector.transform(test_df['clean_text'].values).toarray()
        vocab_dic = tfidf_vector.vocabulary_

    elif mode == "Ngram_TFIDF":
        ngramTfidf_vector = TfidfVectorizer(analyzer='word', token_pattern=r'\w{1,}',
                                            ngram_range=(2,5), max_features=vocabulary_size)
        ngramTfidf_vector.fit(train_df['clean_text'].unique())
        features_train = ngramTfidf_vector.transform(train_df['clean_text'].values).toarray()
        features_test = ngramTfidf_vector.transform(test_df['clean_text'].values).toarray()
        vocab_dic = ngramTfidf_vector.vocabulary_

    y_train = train_df['CAD'].values
    y_test = test_df['CAD'].values
    return features_train, y_train, features_test, y_test, vocab_dic
```

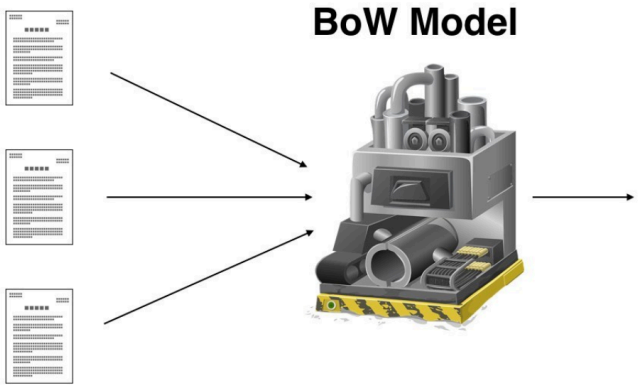
https://github.com/Jill3240/NLP-with-Python/blob/master/BOW_TFIDF_Xgboost_update.ipynb

1. I love dogs.

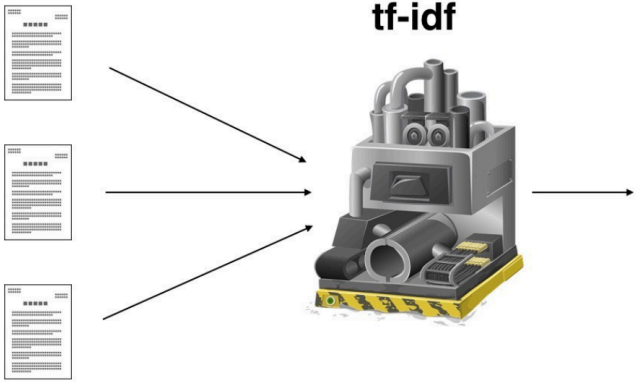
2. I hate dogs and knitting.

3. Knitting is my hobby and my passion.

http://datameetsmedia.com/bag-of-words-tf-idf-explained/



	I	love	dogs	hate	and	knittin	is	my	hobby	passi
Doc 1	1	1	1							
Doc 2	1		1	1	1	1				
Doc 3					1	1	1	2	1	1



	I	love	dogs	hate	and	knittin	is	my	hobby	passi
Doc 1	0.18	0.48	0.18							
Doc 2	0.18		0.18	0.48	0.18	0.18				
Doc 3					0.18	0.18	0.48	0.95	0.48	0.48

Modeling, prediction and important features

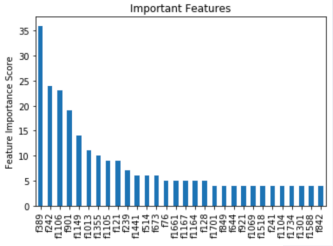
```
def important_feature(estimator, vocab_dic):

    feat_imp = pd.Series(estimator.get_booster().get_score(importance_type='weight')).sort_values(ascending=False).iloc
    feat_lst = feat_imp.index.tolist()
    key_lst = list(vocab_dic.keys())
    feat_num = [(int(w.replace("f",""))) for w in feat_lst]
    imp_features = [key_lst[i-1] for i in feat_num]
    print ("Important features: \n")
    print (imp_features)
    feat_imp.plot(kind='bar', title = "Important Features")
    plt.ylabel('Feature Importance Score')
```

```
def model_pred(X_train, y_train, X_test, y_test, vocab_dic):
    xgb_model = xgb.XGBClassifier(max_depth=50, n_estimators=80, learning_rate=0.1,
                                  colsample_bytree=.7, gamma=0, reg_alpha=4, objective='binary:logistic',
                                  eta=0.3, silent=1, subsample=0.8)
    xgb_model.fit(X_train, y_train)
    pred_prob_train = xgb_model.predict_proba(X_train)
    pred_prob_test = xgb_model.predict_proba(X_test)

    print ("\nModel Report: \n")
    print ("Auc Train Score: %.4g" % metrics.roc_auc_score(y_train,pred_prob_train[:,1]))
    print ("Auc Test Score: %f" % metrics.roc_auc_score(y_test,pred_prob_test[:,1]))

    print ("Detailed classification report: \n")
    y_true, y_pred = y_test,xgb_model.predict(X_test)
    print ((classification_report(y_true, y_pred)))
    important_feature(xgb_model, vocab_dic)
```

https://github.com/Jill3240/NLP-with-Python/blob/master/BOW_TFIDF_Xgboost_update.ipynb

Prediction and important features

BOW XGBoost results:

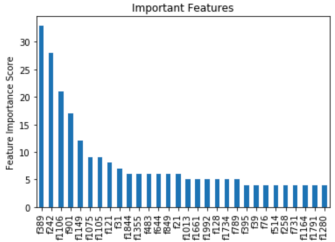


Model Report:

Auc Train Score: 0.9994
Auc Test Score: 0.968180
Detailed classification report:

	precision	recall	f1-score	support
0	0.96	0.94	0.95	192
1	0.83	0.89	0.86	64
accuracy			0.93	256
macro avg	0.89	0.91	0.90	256
weighted avg	0.93	0.93	0.93	256

Word level TF-IDF XGBoost results:

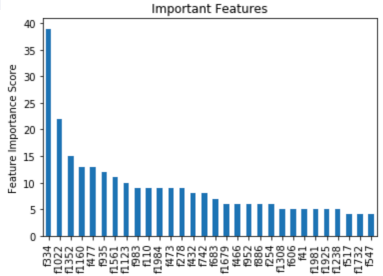


Model Report:

Auc Train Score: 0.9999
Auc Test Score: 0.966349
Detailed classification report:

	precision	recall	f1-score	support
0	0.96	0.94	0.95	192
1	0.83	0.89	0.86	64
accuracy			0.93	256
macro avg	0.89	0.91	0.90	256
weighted avg	0.93	0.93	0.93	256

Ngram level TF-IDF XGBoost results:

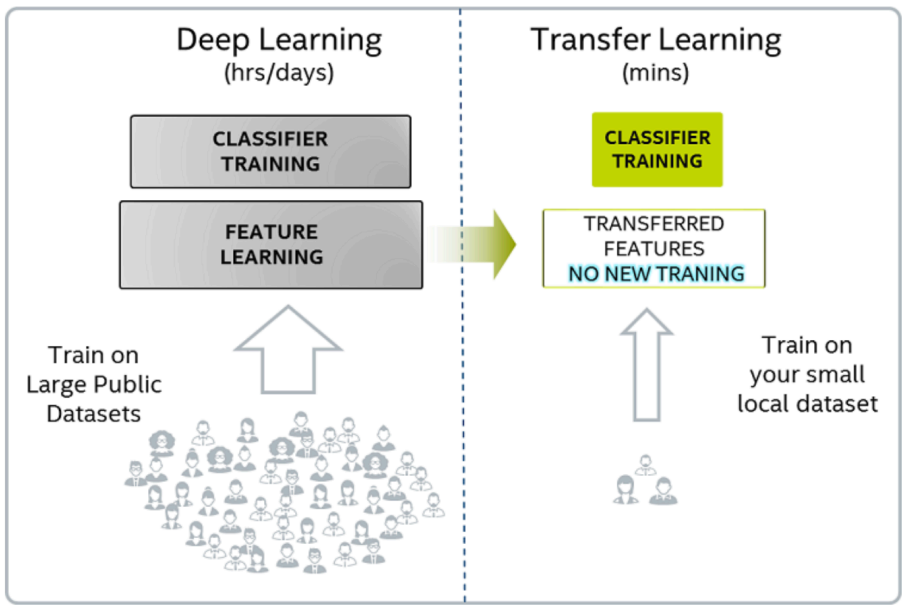


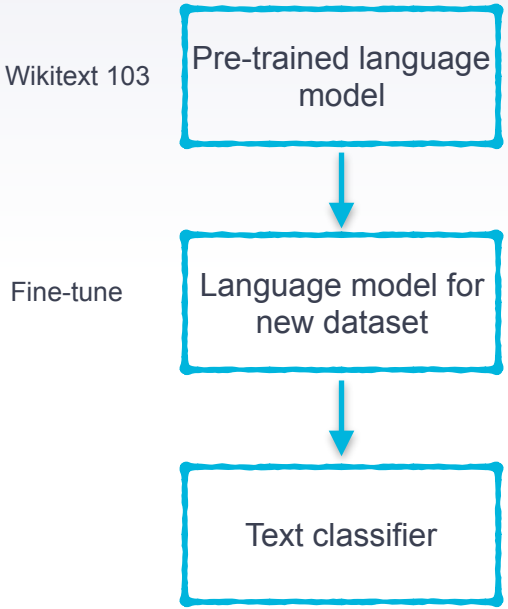
Model Report:

Auc Train Score: 0.9983
Auc Test Score: 0.973307
Detailed classification report:

	precision	recall	f1-score	support
0	0.95	0.94	0.95	192
1	0.83	0.84	0.84	64
accuracy			0.92	256
macro avg	0.89	0.89	0.89	256
weighted avg	0.92	0.92	0.92	256

fastai





https://medium.com/free-code-camp/asl-recognition-using-transfer-learning-918ba054c004

https://medium.com/analytics-vidhya/tutorial-on-text-classification-nlp-using-ulmfit-and-fastai-library-in-python-2f15a2aac065

Data precessing

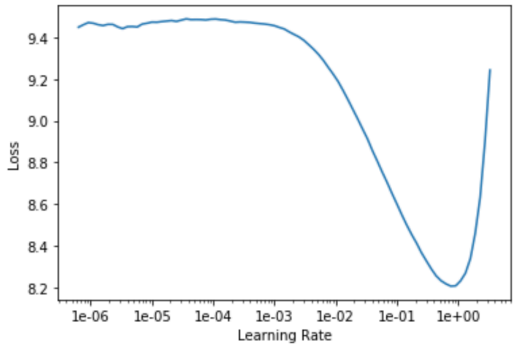
```
data_lm = TextLMDataBunch.from_df(train_df = training_df, valid_df = test_df, path = "")
```

Language model

```
learner = language_model_learner(data_lm, AWD_LSTM, drop_mult=0.7)
```

Fine-tuning

```
learner.lr_find()
learner.recorder.plot()
```




```
learner.fit_one_cycle(1,1e-1)
```

epoch	train_loss	valid_loss	accuracy	time
0	6.951151	6.076974	0.147259	00:14

```
learner.unfreeze()
learner.fit_one_cycle(1,1e-2)
```

epoch	train_loss	valid_loss	accuracy	time
0	5.600085	5.485857	0.189795	00:17

```
learner.save_encoder('ft_encoder')
```

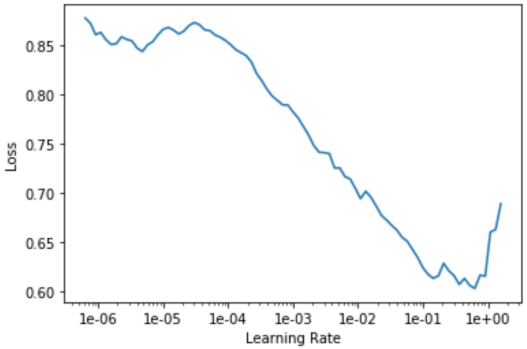
https://towardsdatascience.com/machine-learning-text-classification-language-modelling-using-fast-ai-b1b334f2872d

Classifier

```
data_class = TextClasDataBunch.from_df (path = "", train_df = training_df, valid_df = test_df,
                                        vocab=data_lm.train_ds.vocab, bs=32)
classifier = text_classifier_learner(data_class, AWD_LSTM, drop_mult=0.5)
classifier.load_encoder('ft_encoder')
```

Fine-tuning

```
classifier.lr_find()
classifier.recorder.plot()
```




```
classifier.fit_one_cycle(1, 1e-2)
```

epoch	train_loss	valid_loss	accuracy	time
0	0.657913	0.560812	0.773438	00:14

```
classifier.freeze_to(-2)
classifier.fit_one_cycle(1, slice(5e-3/2., 5e-3))
```

epoch	train_loss	valid_loss	accuracy	time
0	0.467267	0.422267	0.882812	00:18

```
classifier.freeze_to(-3)
classifier.fit_one_cycle(1, slice(5e-3/2., 5e-3))
```

epoch	train_loss	valid_loss	accuracy	time
0	0.414015	0.289455	0.906250	00:22

```
classifier.unfreeze()
#classifier.fit_one_cycle(1, slice(1e-3/100, 1e-3),moms = (0.8,0.7))
#classifier.fit_one_cycle(5, slice(1e-3/100, 1e-3),moms = (0.8,0.7))
classifier.fit_one_cycle(1, slice(1e-3/100, 1e-3),moms = (0.8,0.7))
```

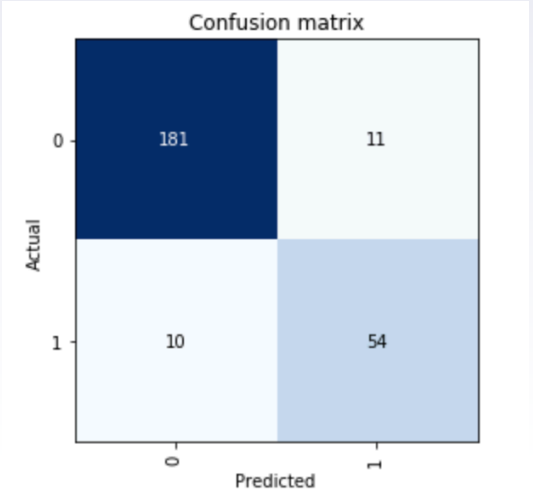
epoch	train_loss	valid_loss	accuracy	time
0	0.341836	0.288586	0.917969	00:29

https://towardsdatascience.com/machine-learning-text-classification-language-modelling-using-fast-ai-b1b334f2872d

Prediction

```
classifier.show_results(ds_type=DatasetType.Train, rows=20)
```

text	target	prediction
xxbos tgcho discharg summari unsign di unsign breundail netfa esophag bleed hepatocellular carcinoma diabet mellitu cirrhosi hepat patient year old man histori hepat cirrhosi hepatocellular carcinoma diabet mellitu usual state health day prior admiss episod hematemesi discus friend xxunk xxunk xxunk memori hospit seek medic attent hope would go away morn admiss mr kote awok sleep immedi vomit larg quantiti blood said larg quantiti associ right upper quadrant pain shortli	0	0
xxbos medicin patient record known allergi drug short breath endotrach intub central venou cathet swan ganz cathet placement arteri line placement right side thoracentesi male cad lad ptca yr ago copd dm aicd pocket infect present juan worsen sob day feel cloudi patient current xxunk region hospit began develop worsen sob past day talk daughter phone note desatur high brought hospit orthoped juan evalu patient state felt cloudi late seem	1	1
xxbos fih discharg summari unsign di unsign abreun xxunk necrot fasciti right thigh diabet mellitu hypothyroid anasarca pulmonari nodul anemia preren azotemia methicillin resist staphylococcu aureu pneumonia wound debrid split thick skin graft august tracheostomi gastrostomi placement endoscop retrograd cholangiopancreatographi abreun fifti nine year old white femal recent diagnos may non insulin depend diabet mellitu hypothyroid iron defici anemia late may fell bath tub small xxunk fractur left elbow one	0	0
xxbos puomc discharg summari unsign di unsign xxunk leach year old male admit transfer morehegron valley hospit xxunk virginia evalu manag acceler chest pain syndrom mr leach long complex histori coronari arteri diseas without known cardiovascular ill suffer acut anterior myocardi infarct august treat time tpa subsequ cardiac catheter reveal lesion proxim left anterior descend coronari arteri minor diseas elsewher coronari distribut eject fraction mid apo angina follow catheter ultim	1	1
xxbos bh left proxim humeru fractur unsign di unsign junk kacholera left proxim humeru fractur adult onset diabet mellitu year histori silent myocardi infarct coronari arteri bypass graft three vessel histori chronic stabl angina pectori fournier apo gangren versu necrot fasciiti congest heart failur hospit rib fractur glaucoma diabet retinopathi blind right eye histori gastric stapl left rib fractur right femur surgeri left shoulder hemiarthroplasti upper endoscopi patient year old	1	1
xxbos medicin losartan aspirin lisinopril hctz intub airway protect tunnel line chang wire femal wtih pmh signific esrd hd type dm recent pe result pea arrest admit juan sepsi unknown etiolog pt recent admit hospit orthoped xxunk hypotens thought secondari xxunk hd howev septic compon hypotens also consid pt found citrobact urin diff stool pt discharg arbour xxunk hospit senior life normal state health three day ago son report develop	0	0



	precision	recall	f1-score	support
0	0.97	0.91	0.94	192
1	0.77	0.91	0.83	64
accuracy			0.91	256
macro avg	0.87	0.91	0.89	256
weighted avg	0.92	0.91	0.91	256

Discussions

- Text cleaning: remove more stop words.
- XGBoost: larger dataset, hyperparameter tuning
- Transfer learning: exploring more parameters.

THANKS!