Jeremey_code_BKUP202107300829

July 30, 2021

1 DS 5110 Group Project

Team: Alexandra Cathcart (adc6fs), Benjamin Feciura (bmf3bw), Jeremey Donovan (jdd5dw), Jordan Hiatt (jdh2e)

Original data: https://www.kaggle.com/reddit/reddit-comments-may-2015

1.1 Includes & Spark Setup

[2]:
```
import pandas as pd
import time

from pyspark import StorageLevel

from pyspark.mllib.evaluation import BinaryClassificationMetrics,␣
↪MulticlassMetrics

from pyspark.ml import Pipeline, PipelineModel
from pyspark.ml.classification import LogisticRegression
from pyspark.ml.evaluation import BinaryClassificationEvaluator,␣
↪MulticlassClassificationEvaluator
from pyspark.ml.feature import *
from pyspark.ml.tuning import CrossValidator, CrossValidatorModel,␣
↪ParamGridBuilder

from pyspark.sql import SparkSession
from pyspark.sql.functions import col, countDistinct, lower, size, split, udf,␣
↪when
from pyspark.sql.types import ArrayType, FloatType, IntegerType,  StringType,␣
↪StructType
```

[3]:
```
#from pyspark import SparkContext
spark = SparkSession.builder.getOrCreate()
sc=spark.sparkContext
```

1.2 Code Control

[4]:
```
# EDA is slow so runEDA = 1 to run
runEDA=0
```

[5]:
```
# train, test, holdout
# key code control b/c we have been experiencing memory issues with a 50/50␣
 ↪train/test split
trainPct=0.1
testPct=0.1
holdoutPct=0.8
```

[6]:
```
# persisting trainDF should speed training but we have been experiencing memory␣
 ↪issues
persistTrainDF=0
```

[7]:
```
# Over-ride parallelism: We have been expreiencing memory issues. Set to 0 for␣
 ↪no over-ride.
# Otherwise, set to the desired over-ride integer value
overrideParallelism=1
```

[8]:
```
# loadCVmodel: If blank, then do not load and instead run CV
# otherwise, provide name of cv model to load
#loadCVmodel="lrModel20210729-170848"
loadCVmodel=""
```

1.3 Data Import and Pre-Processing

1.3.1 Data Import

[9]:
```
# Import the reddit data
full_path = '/project/ds5559/r-slash-group8/sample.csv'

df = spark.read.csv(full_path,  inferSchema=True, header = True)
```

[10]:
```
# Import the Bad Word data
schema = StructType().add("badWord",StringType(),True)
dfBW=spark.read.format("csv").schema(schema).load('bad_words.csv')
#  dfBW.show(5)  # not showing since words are quite vulgar

# Also create in list format
listBW=list(dfBW.select('badWord').toPandas()['badWord'])
# listBW
```

[11]:
```
# Create a regex with all the bad words
# if there is an issue, try \\\\b instead; just \b probably has issues
listBW=list(map(lambda line: "\\b" + line + "\\b",listBW))
```

```
delim='|'
strBW=delim.join(listBW)
```

1.3.2 Filtering

[12]:
```
# Drop unneeded cols from dataframe
df=df.
 ↪drop('_c0','created_utc','subreddit_id','link_id','name','score_hidden','author_flair_css_c
 ↪'gilded', \
        'author_flair_text','id','archived','retrieved_on',␣
 ↪'edited','controversiality','parent_id','score')

# convert integer cols (ups, downs, and gilded) to integers
# Note: we could have done this by defining a schema before the csv read
df=df.withColumn("ups",df.ups.cast(IntegerType()))
df=df.withColumn("downs",df.downs.cast(IntegerType()))
#df=df.withColumn("gilded",df.gilded.cast(IntegerType()))  # Removed gilded␣
 ↪since not used in this analysis

# Confirm new schema
df.printSchema()
df.show(5)
```

```
root
 |-- ups: integer (nullable = true)
 |-- subreddit: string (nullable = true)
 |-- removal_reason: string (nullable = true)
 |-- downs: integer (nullable = true)
 |-- author: string (nullable = true)
 |-- body: string (nullable = true)
 |-- distinguished: string (nullable = true)

+----+---------+--------------+-----+-----------+--------------------+---------
----+
| ups|subreddit|removal_reason|downs|     author|
body|distinguished|
+----+---------+--------------+-----+-----------+--------------------+---------
----+
|   4|soccer_jp|            NA|    0|      rx109|                    |
null|
|null|     null|          null| null|       null|                null|
null|
|   0|     null|          null| null|       null|                null|
null|
|   4|      nba|            NA|    0|  WyaOfWade|gg this one's ove…|
NA|
|   0| politics|            NA|    0|Wicked_Truth|Are you really im…|
```

```
NA|
+----+---------+--------------+-----+------------+-------------------+---------
----+
only showing top 5 rows
```

[13]:
```
# Count the number of rows before removing NA
df.count()
# There are 15,317,725 rows
```

[13]:
```
15317725
```

[14]:
```
# Remove rows where up, down, or body is null. We  do this since inference of␣
 ↪these values is not applicable
df=df.filter(df['ups'].isNotNull())
df=df.filter(df['downs'].isNotNull())
df=df.filter(df['body'].isNotNull())

df.show(5)
```

```
+---+---------+--------------+-----+--------------+-------------------+--------
-----+
|ups|subreddit|removal_reason|downs|        author|
body|distinguished|
+---+---------+--------------+-----+--------------+-------------------+--------
-----+
|  4|soccer_jp|            NA|    0|         rx109|                   |
null|
|  4|      nba|            NA|    0|     WyaOfWade|gg this one's ove…|
NA|
|  0| politics|            NA|    0|  Wicked_Truth|Are you really im…|
NA|
|  3|AskReddit|            NA|    0|      jesse9o3|No one has a Euro…|
NA|
|  3|AskReddit|            NA|    0|beltfedshooter|"That the kid ""…|
NA|
+---+---------+--------------+-----+--------------+-------------------+--------
-----+
only showing top 5 rows
```

[15]:
```
# Remove rows where the author was '[deleted]'
df=df.filter(df['author']!='[deleted]')

# Remove author "0"
df=df.filter(df['author']!='0')
```

```
# Remove rows where the author was 'AutoModerator'
# see https://www.reddit.com/wiki/automoderator
df=df.filter(df['author']!='AutoModerator')
```

[16]:
```
# Count the number of rows AFTER removing NA
df.count()
# There now 9,226,090 rows
```

[16]: 9226090

1.3.3 Binning & Feature Engineering

[17]:
```
# Lowercase all body text
df=df.withColumn('body',lower(col('body')))
```

[18]:
```
# Even though we dropped the column, adding score back into dataframe by␣
↪computing it
df=df.withColumn('score',df['ups']-df['downs'])
df=df.withColumn("score",df.score.cast(IntegerType()))
df.show(5)
```

```
+---+---------+--------------+-----+--------------+--------------------+--------
-----+-----+
|ups|subreddit|removal_reason|downs|        author|
body|distinguished|score|
+---+---------+--------------+-----+--------------+--------------------+--------
-----+-----+
|  4|soccer_jp|            NA|    0|         rx109|                    |
null|    4|
|  4|      nba|            NA|    0|     WyaOfWade|gg this one's ove…|
NA|    4|
|  0| politics|            NA|    0|  Wicked_Truth|are you really im…|
NA|    0|
|  3|AskReddit|            NA|    0|      jesse9o3|no one has a euro…|
NA|    3|
|  3|AskReddit|            NA|    0|beltfedshooter|"that the kid ""…|
NA|    3|
+---+---------+--------------+-----+--------------+--------------------+--------
-----+-----+
only showing top 5 rows
```

[19]:
```
# Determine a scoreSentiment as either postive, neutral, or negative.
# This will be our response variable

# Drop scoreSentiment if it already exists
```

```
df=df.drop('scoreSentiment')

# Set up bucketizer
splits = [-float("inf"), -0.1,0.1, float("inf")]
bkt = Bucketizer(splits=splits, inputCol="score", outputCol="scoreSentiment")

# Transform to add scoreSentiment: 0=negative; 1=neutral; 2=positive.
df=bkt.transform(df)

# !!! Cannot shift to -1,0,1 since LR must start with 0 !!!
# To make things more clear, shift to -1=negative; 0=neutral; 1=positive
#df=df.withColumn("scoreSentiment", \
#                 when(df['scoreSentiment']==0,-1) \
#                 .when(df['scoreSentiment']==1,0) \
#                 .otherwise(1)
#                )

df.show(2)
```

```
+---+---------+--------------+-----+---------+--------------------+-------------
+-----+--------------+
|ups|subreddit|removal_reason|downs|   author|
body|distinguished|score|scoreSentiment|
+---+---------+--------------+-----+---------+--------------------+-------------
+-----+--------------+
|  4|soccer_jp|            NA|    0|    rx109|                    |         null|
4|           2.0|
|  4|      nba|            NA|    0|WyaOfWade|gg this one's ove…|
NA|    4|           2.0|
+---+---------+--------------+-----+---------+--------------------+-------------
+-----+--------------+
only showing top 2 rows
```

[20]:
```
# Flag comments containing bad words
df=df.withColumn('bwFlag',col('body').rlike(strBW))
```

[21]:
```
# Append bodyWordCount
df=df.withColumn("bodyWordCount", size(split(df['body'], ' ')))
#df.show(5)
```

[22]:
```
# Though not the cleanest thing to do from a data sci perspective, we
# are going to drop the neutral sentiment rows so we can do binomial
# rather than multinomial regression; neutral currently "1"
df=df.filter(df['scoreSentiment']!=1)
# Shift positive from 2 to 1
df=df.withColumn("scoreSentiment", \
```

```
                  when(df['scoreSentiment']==2,1) \
                  .when(df['scoreSentiment']==0,0) \
                  .otherwise(-1)
                 )
# we should never have the otherwise case!!!
```

[23]:
```
# Cross-validator explicity wants response to be called "label"
# so copying scoreSentiment to label in all DFs
df=df.withColumn("label", df["scoreSentiment"])
```

1.4 Data Splitting & Sampling

[24]:
```
seed=314
trainDF,testDF, holdoutDF=df.randomSplit([trainPct,testPct,holdoutPct],seed)
```

1.5 EDA

[25]:
```
if runEDA:
    # How many comments have bad words?
    # Confirm the flagging worked by looking at how many comments contain bad␣
 ↪words vs good
    # NOTE: This has a rather long runtime!!!
    df.groupby('bwFlag').agg({"bwFlag":"count"}).show()
    #df.filter(df['bwFlag']==True).show(5,False)
```

[26]:
```
if runEDA:
    # How many authors are there?
    df.select(countDistinct('author')).show()
    # There are 1,216,598 authors
```

[27]:
```
if runEDA:
    # Show the top 10 authors with sum of ups and downs
    df.groupby('author').agg({"author":"count","ups":"sum","downs":
 ↪"sum","score":"sum"}).sort(col('count(author)').desc()).show(10)
```

Odd that the preceding authors have no down but this is correct

[28]:
```
if runEDA:
    # Show authors with the lowest scores
    df.groupby('author').agg({"score":"sum","ups":"sum","downs":"sum"}).
 ↪sort(col('sum(score)').asc()).show(10)
```

[29]:
```
if runEDA:
    # Get a summary of score sentiment by label
    #tmpDF.groupby('scoreSentiment').agg({"scoreSentiment":"count"}).show()
    df.groupby('scoreSentiment').agg({"scoreSentiment":"count"}).show()
```

[30]:
```
#if runEDA:
    # Show graphical Distribution of sentiment (TBD)
```

1.6 Model: Predict Sentiment from body

1.6.1 Set up pipeline

[31]:
```
# Create TF (Term Frequency) feature
tok = Tokenizer(inputCol="body", outputCol="words")
htf = HashingTF(inputCol="words", outputCol="tf")  # numFeatures will be a␣
 ↪hyper-parameter

#testing
tmpDF=tok.transform(df)
tmpDF=htf.transform(tmpDF)
tmpDF.select('words','tf').show(2)
```

```
+--------------------+--------------------+
|               words|                  tf|
+--------------------+--------------------+
|                [ ]|(262144,[85691],[…|
|[gg, this, one's,…|(262144,[5674,905…|
+--------------------+--------------------+
only showing top 2 rows
```

[32]:
```
# Create w2v (word to vec) feature

# the comment string needs to be turned into a vector for w2v to work
# unfortunately, VectorAssember does not work on string so we need a UDF

# Create UDF (note: split(anything,0) simply means don't split)
str_to_vec=spark.udf.register("str_to_vec",
                              lambda row:row.split("#",0),
                              ArrayType(StringType()))

# set up the tranformation
rva=SQLTransformer(statement="SELECT *, str_to_vec(body) bodyVec FROM __THIS__")

w2v = Word2Vec(inputCol='bodyVec', outputCol='w2v')  # not setting minCount

# testing
"""
tmpDF=rva.transform(df)
model=w2v.fit(tmpDF)
tmpDF=model.transform(tmpDF)
tmpDF.show(2)
```

```
"""
```

[32]:
```
'\ntmpDF=rva.transform(df)\nmodel=w2v.fit(tmpDF)\ntmpDF=model.transform(tmpDF)\n
tmpDF.show(2)\n'
```

[33]:
```
# Assemble predictors
va=VectorAssembler(inputCols=['tf','w2v','bwFlag','bodyWordCount'],outputCol='features')
```

[34]:
```
# Set up the regression model; regParam & elasticNetParam will be␣
 ↪hyper-parameters
# CrossVal currently requires the labelCol to be precisely called 'label'
#lr = LogisticRegression(labelCol='scoreSentiment',maxIter=10)
lr = LogisticRegression(labelCol='label',maxIter=10)
```

[35]:
```
# Build the pipeline
#pipeline=Pipeline(stages=[bkt,tok,htf,rva,w2v,va,lr])  # took out bkt since␣
 ↪this is pre-EDA
pipeline=Pipeline(stages=[tok,htf,rva,w2v,va,lr])
```

1.6.2 Set up hyperparameter tuning & Cross-Validation

[36]:
```
# Set up the parameter grid

"""
# This version works and homes in on elasticNetParam=0
paramGrid = ParamGridBuilder() \
    .addGrid(htf.numFeatures, [200]) \
    .addGrid(lr.regParam, [0.1]) \
    .addGrid(lr.elasticNetParam, [0.0, 0.5, 1.0]) \
    .build()
"""

paramGrid = ParamGridBuilder() \
    .addGrid(htf.numFeatures, [200]) \
    .addGrid(lr.regParam, [0.1, 0.01]) \
    .addGrid(lr.elasticNetParam, [0.0, 0.5, 1.0]) \
    .build()

# This paramGrid for testing
"""
paramGrid = ParamGridBuilder() \
    .addGrid(htf.numFeatures, [200]) \
    .addGrid(lr.regParam, [0.3]) \
    .addGrid(lr.elasticNetParam, [0.5]) \
    .build()
"""
```

[36]:
```
'\nparamGrid = ParamGridBuilder()     .addGrid(htf.numFeatures, [200])
.addGrid(lr.regParam, [0.3])     .addGrid(lr.elasticNetParam, [0.5])
.build()\n'
```

[37]:
```
# Too inspect paramGrid, uncomment next 4 lines
"""
print('-'*30)
#print('paramGrid', paramGrid, '\n')
#print('len(paramGrid): {}'.format(len(paramGrid)))
print('-'*30)
"""
```

[37]:
```
"\nprint('-'*30)\n#print('paramGrid', paramGrid, '\n')\n#print('len(paramGrid):
{}'.format(len(paramGrid)))\nprint('-'*30)\n"
```

[38]:
```
# Treat the Pipeline as an Estimator, wrapping it in a CrossValidator instance.
# Using the pipeline as the estimator slows things down but is necessary if␣
 ↪tuning featurziers.  If not, set the
# model specification as the estimator with estimator=lr (I think; though not␣
 ↪sure if that means lr needs to be removed from pipeline)
numFolds=5
crossval = CrossValidator(estimator=pipeline,
                          estimatorParamMaps=paramGrid,
                         ␣
 ↪evaluator=BinaryClassificationEvaluator(labelCol='label'),
                          numFolds=numFolds,
                          collectSubModels=True)
```

1.6.3 Train the Model

[39]:
```
# Determine parallelism
# This resource: see https://databricks.com/session/
 ↪model-parallelism-in-spark-ml-cross-validation
# says that best practice is parallelism = (# cores)/(# partitions) but␣
 ↪generally not more than 10
numPartitions=trainDF.rdd.getNumPartitions()
numCores=sc.defaultParallelism
parallelism=int(round(numCores/numPartitions,0))
# also see https://stackoverflow.com/questions/42171499/
 ↪get-current-number-of-partitions-of-a-dataframe

# constrain to between 1 and 10
if parallelism<1:
    parallelism=1
elif parallelism > 10:
    parallelism=10
```

```
# ------------------------------------------------------------------------

"""
# Another thing we can do is treat cores as fixed and repartition to get a␣
 ↪target parallelism
# while avoiding memory issues that occur when != cores/partitions
# in the future: verify cores/partitions is correct; might want to do something␣
 ↪to avoid having
# too few partitions
parallelism=2
targetNumPartitions=int(round(numCores/parallelism,0))
if (targetNumPartitions>=1):
    if (targetNumPartitions<numPartitions):
        trainDF = trainDF.coalesce(targetNumPartitions) # no shuffling but can␣
 ↪only be used for decreasing numPartitions
    else:
        trainDF = trainDF.repartition(targetNumPartitions)  # this involves␣
 ↪shuffling to less efficient
"""

# ------------------------------------------------------------------------


# However, elsewhere, you typically see that partitions should be 2x to 4x the␣
 ↪number of cores!
# So, we could just override (note: 4 yielded memory errors)
if overrideParallelism:
    parallelism=overrideParallelism
```

[40]:
```
# print out parallelism
parallelism
```

[40]: 1

[41]:
```
# Cache trainDF to speed up cross validation; we could use .select(colnames...)␣
 ↪to use less memory
# Cache & persist failed with 96GB and down to 50/50 train test split
# yeah! worked with 25/25/50 train/test/holdout split with 96GB allocated!!!
if persistTrainDF:
    #trainDF=trainDF.cache()
    trainDF=trainDF.persist(StorageLevel.MEMORY_AND_DISK)
    trainDF.count()  # call count to actually cache the data
```

[42]:
```
if len(loadCVmodel)==0:
```

```
    # Run cross-validation, and choose the best set of parameters. Print the␣
↪training time.
    import time
    t0 = time.time()
    if parallelism==1:
        cvModel = crossval.fit(trainDF) # train models (no parallelism)
    else:
        cvModel = crossval.setParallelism(parallelism).fit(trainDF) # train␣
↪models in parallel
    print("train time:", time.time() - t0)
    print('-'*30)
    # Took 3580 secs (~1hr) to run single params set with 50/50 split, 5 fold␣
↪on 8 cores with 32 GB memmory & no parallelism & no cache/persist
    # 10/10/90 train/test/holdout without parallelism took 3111 secs

    # save the model with a timestamp
    timestr = time.strftime("%Y%m%d-%H%M%S")
    cvModel.save("lrModel"+timestr)
    pipeline.save("lrPipeline"+timestr)
else:
    # Load the model and the pipeline  (should these be preceded by "val")
    cvModel = CrossValidatorModel.load(loadCVmodel)
    #val sameModel = PipelineModel.load("/path-to-my-pipeline/
↪spark-log-reg-transfer-pipeline")
```

```
train time: 3352.6148397922516
------------------------------
```

[43]:
```
# release the cache
if persistTrainDF:
    trainDF.unpersist()
```

[44]:
```
cvModel.bestModel.stages[-1].extractParamMap()
# best model has the following:
# elasticNetParam = 0
# regParam = 0.01
```

[44]:
```
{Param(parent='LogisticRegression_4446ecd1f38f', name='aggregationDepth',
doc='suggested depth for treeAggregate (>= 2).'): 2,
 Param(parent='LogisticRegression_4446ecd1f38f', name='elasticNetParam',
doc='the ElasticNet mixing parameter, in range [0, 1]. For alpha = 0, the
penalty is an L2 penalty. For alpha = 1, it is an L1 penalty.'): 0.0,
 Param(parent='LogisticRegression_4446ecd1f38f', name='family', doc='The name of
family which is a description of the label distribution to be used in the model.
Supported options: auto, binomial, multinomial'): 'auto',
 Param(parent='LogisticRegression_4446ecd1f38f', name='featuresCol',
doc='features column name.'): 'features',
```

```
 Param(parent='LogisticRegression_4446ecd1f38f', name='fitIntercept',
doc='whether to fit an intercept term.'): True,
 Param(parent='LogisticRegression_4446ecd1f38f', name='labelCol', doc='label
column name.'): 'label',
 Param(parent='LogisticRegression_4446ecd1f38f', name='maxIter', doc='max number
of iterations (>= 0).'): 10,
 Param(parent='LogisticRegression_4446ecd1f38f', name='predictionCol',
doc='prediction column name.'): 'prediction',
 Param(parent='LogisticRegression_4446ecd1f38f', name='probabilityCol',
doc='Column name for predicted class conditional probabilities. Note: Not all
models output well-calibrated probability estimates! These probabilities should
be treated as confidences, not precise probabilities.'): 'probability',
 Param(parent='LogisticRegression_4446ecd1f38f', name='rawPredictionCol',
doc='raw prediction (a.k.a. confidence) column name.'): 'rawPrediction',
 Param(parent='LogisticRegression_4446ecd1f38f', name='regParam',
doc='regularization parameter (>= 0).'): 0.01,
 Param(parent='LogisticRegression_4446ecd1f38f', name='standardization',
doc='whether to standardize the training features before fitting the model.'):
True,
 Param(parent='LogisticRegression_4446ecd1f38f', name='threshold',
doc='Threshold in binary classification prediction, in range [0, 1]. If
threshold and thresholds are both set, they must match.e.g. if threshold is p,
then thresholds must be equal to [1-p, p].'): 0.5,
 Param(parent='LogisticRegression_4446ecd1f38f', name='tol', doc='the
convergence tolerance for iterative algorithms (>= 0).'): 1e-06}
```

[68]:
```
#cvModel.subModels
#cvModel.subModels[1][1].stages[-1].extractParamMap()
```

1.7 Model Evaluation with AUROC

[46]:
```
# Create the necessary evaluators
evaluator=BinaryClassificationEvaluator(labelCol='label')
mcEvaluator = MulticlassClassificationEvaluator(metricName="accuracy")
```

[47]:
```
# Generate predictions
predict_train=cvModel.transform(trainDF)
predict_test=cvModel.transform(testDF)
```

[48]:
```
#predict_test.show(3)
# besides initial df cols and those created by pipeline, we ahve label,␣
 ↪rawPrediction, probability, and prediction
```

1.7.1 Accuracy

[49]:
```
print("Train Accuracy:", mcEvaluator.evaluate(predict_train))
print("Test Accuracy:", mcEvaluator.evaluate(predict_test))
```

```
Train Accuracy: 0.9549860734338707
Test Accuracy: 0.9552607024095798
```

1.7.2 precision, recall, F1 score

Source: https://stackoverflow.com/questions/60772315/how-to-evaluate-a-classifier-with-pyspark-2-4-5

[50]:
```
weightedPrecision = mcEvaluator.evaluate(predict_test, {mcEvaluator.metricName:␣
↪"weightedPrecision"})
print("Test precision is {}".format(weightedPrecision))

weightedRecall = mcEvaluator.evaluate(predict_test, {mcEvaluator.metricName:␣
↪"weightedRecall"})
print("Test recall is {}".format(weightedRecall))

f1 = mcEvaluator.evaluate(predict_test, {mcEvaluator.metricName: "f1"})
print("Test f1 is {}".format(f1))
```

```
Test precision is 0.9214841866897692
Test recall is 0.9552607024095798
Test f1 is 0.9334140348304236
```

[51]:
```
weightedPrecision = mcEvaluator.evaluate(predict_train, {mcEvaluator.metricName:
↪ "weightedPrecision"})
print("Train precision is {}".format(weightedPrecision))

weightedRecall = mcEvaluator.evaluate(predict_train, {mcEvaluator.metricName:␣
↪"weightedRecall"})
print("Train recall is {}".format(weightedRecall))

f1 = mcEvaluator.evaluate(predict_train, {mcEvaluator.metricName: "f1"})
print("Train f1 is {}".format(f1))
```

```
Train precision is 0.9232611846478553
Train recall is 0.9549860734338705
Train f1 is 0.9330062499313787
```

1.7.3 Confusion Matrix

Source: https://stackoverflow.com/questions/58404845/confusion-matrix-to-get-precsion-recall-f1score

Confusion matrix references that may be helpful if the above does not work:

https://gist.github.com/ispmarin/05feacd8be5e2901cf2b35453a148060

https://shihaojran.com/distributed-machine-learning-using-pyspark/

[52]:
```
#important: need to cast to float type, and order by prediction, else it won't␣
 ↪work
preds_and_labels = predict_test.select(['prediction','label']).
 ↪withColumn('label', col('label').cast(FloatType())).orderBy('prediction')

#select only prediction and label columns
preds_and_labels = preds_and_labels.select(['prediction','label'])

metrics = MulticlassMetrics(preds_and_labels.rdd.map(tuple))
print("Confustion matrix for test:")
print(metrics.confusionMatrix().toArray())
```

```
Confustion matrix for test:
[[2.00000e+00 3.94530e+04]
 [8.00000e+00 8.42558e+05]]
```

[53]:
```
#important: need to cast to float type, and order by prediction, else it won't␣
 ↪work
preds_and_labels = predict_train.select(['prediction','label']).
 ↪withColumn('label', col('label').cast(FloatType())).orderBy('prediction')

#select only prediction and label columns
preds_and_labels = preds_and_labels.select(['prediction','label'])

metrics = MulticlassMetrics(preds_and_labels.rdd.map(tuple))
print("Confustion matrix for train:")
print(metrics.confusionMatrix().toArray())
```

```
Confustion matrix for train:
[[2.00000e+00 3.97020e+04]
 [6.00000e+00 8.42417e+05]]
```

1.7.4 AUC

[54]:
```
evalTrain=evaluator.evaluate(predict_train)
evalTest=evaluator.evaluate(predict_test)
```

[55]:
```
print("The area under ROC for train set after CV  is {}".format(evalTrain))
print("The area under ROC for test set after CV  is {}".format(evalTest))

# source: https://dhiraj-p-rai.medium.com/
 ↪logistic-regression-in-spark-ml-8a95b5f5434c
```

```
The area under ROC for train set after CV  is 0.6010364050742824
```

```
The area under ROC for test set after CV  is 0.5955939049250416
```

1.8 Sensitivity Analysis

[72]:
```
# Our best model has elasticNetParam=0 and regParam=0.01.  So, we can do␣
 ↪sensitivity analysis
# by comparing to regParam=0.1 (we have this in subModels but I could not out␣
 ↪how to access)
paramGrid_sens = ParamGridBuilder() \
    .addGrid(htf.numFeatures, [200]) \
    .addGrid(lr.regParam, [0.1]) \
    .addGrid(lr.elasticNetParam, [0.0]) \
    .build()
```

[73]:
```
# Treat the Pipeline as an Estimator, wrapping it in a CrossValidator instance.
# Using the pipeline as the estimator slows things down but is necessary if␣
 ↪tuning featurziers.  If not, set the
# model specification as the estimator with estimator=lr (I think; though not␣
 ↪sure if that means lr needs to be removed from pipeline)
numFolds=5
crossval_sens = CrossValidator(estimator=pipeline,
                           estimatorParamMaps=paramGrid_sens,
                           ␣
 ↪evaluator=BinaryClassificationEvaluator(labelCol='label'),
                           numFolds=numFolds,
                           collectSubModels=False)
```

[ ]:
```
cvModel_sens = crossval_sens.fit(trainDF) # train models (no parallelism)
```

[ ]:
```
# Generate predictions
predict_train_sens=cvModel_sens.transform(trainDF)
predict_test_sens=cvModel_sens.transform(testDF)
```

[ ]:
```
evaluator=BinaryClassificationEvaluator(labelCol='label')
evalTrain=evaluator.evaluate(predict_train_sens)
evalTest=evaluator.evaluate(predict_test_sens)
```

[ ]:
```
print("The area under ROC for train set after CV  is {}".format(evalTrain))
print("The area under ROC for test set after CV  is {}".format(evalTest))

# source: https://dhiraj-p-rai.medium.com/
 ↪logistic-regression-in-spark-ml-8a95b5f5434c
```

2 CODE ONLY OK ABOVE THIS POINT !!!!

[30]:
```
# Fit the multinomial logistic regression model; this is old - before␣
 ↪implememenenting cross validation
mlrModel=pipeline.fit(trainDF)
```

[24]:
```
# Training Summary
# source: https://spark.apache.org/docs/latest/ml-classification-regression.html

# Fix source: https://stackoverflow.com/questions/37278999/
 ↪logistic-regression-with-spark-ml-data-frames
lrm=mlrModel.stages[-1]

# Print the coefficients and intercept for multinomial logistic regression
print("Coefficients: \n" + str(lrm.coefficientMatrix))
print("Intercept: " + str(lrm.interceptVector))

trainingSummary = lrm.summary

# Obtain the objective per iteration
objectiveHistory = trainingSummary.objectiveHistory
print("objectiveHistory:")
for objective in objectiveHistory:
    print(objective)

# for multiclass, we can inspect metrics on a per-label basis
print("False positive rate by label:")
for i, rate in enumerate(trainingSummary.falsePositiveRateByLabel):
    print("label %d: %s" % (i, rate))

print("True positive rate by label:")
for i, rate in enumerate(trainingSummary.truePositiveRateByLabel):
    print("label %d: %s" % (i, rate))

print("Precision by label:")
for i, prec in enumerate(trainingSummary.precisionByLabel):
    print("label %d: %s" % (i, prec))

print("Recall by label:")
for i, rec in enumerate(trainingSummary.recallByLabel):
    print("label %d: %s" % (i, rec))

print("F-measure by label:")
for i, f in enumerate(trainingSummary.fMeasureByLabel()):
    print("label %d: %s" % (i, f))

accuracy = trainingSummary.accuracy
```

```
falsePositiveRate = trainingSummary.weightedFalsePositiveRate
truePositiveRate = trainingSummary.weightedTruePositiveRate
fMeasure = trainingSummary.weightedFMeasure()
precision = trainingSummary.weightedPrecision
recall = trainingSummary.weightedRecall
print("Accuracy: %s\nFPR: %s\nTPR: %s\nF-measure: %s\nPrecision: %s\nRecall: %s"
      % (accuracy, falsePositiveRate, truePositiveRate, fMeasure, precision,␣
 ↪recall))
```

```
Coefficients:
3 X 300 CSRMatrix

Intercept: [-1.0264586714522934,-1.0107704204551655,2.037229091907459]
objectiveHistory:
0.35298803317465105
False positive rate by label:
label 0: 0.0
label 1: 0.0
label 2: 1.0
True positive rate by label:
label 0: 0.0
label 1: 0.0
label 2: 1.0
Precision by label:
label 0: 0.0
label 1: 0.0
label 2: 0.9139359489937812
Recall by label:
label 0: 0.0
label 1: 0.0
label 2: 1.0
F-measure by label:
label 0: 0.0
label 1: 0.0
label 2: 0.9550329513109017
Accuracy: 0.9139359489937812
FPR: 0.9139359489937812
TPR: 0.9139359489937812
F-measure: 0.8728389466766606
Precision: 0.8352789188631633
Recall: 0.9139359489937812
```

[25]:
```
# Make preductions on the test data
mlrPrediction=mlrModel.transform(testDF)
```

[26]:
```
mlrPrediction.select('scoreSentiment','prediction').show(3)
```

```
+--------------+----------+
|scoreSentiment|prediction|
+--------------+----------+
|           0.0|       2.0|
|           0.0|       2.0|
|           0.0|       2.0|
+--------------+----------+
only showing top 3 rows
```

2.0.1 TBD: Evaluate the predictions. Judging from the training though, it seems to over-predict category 2 "positive" — which is the most prevalent

[27]:
```
# Stuff with ngrams not currently used

#May need to drop col when rerunning
#df=df.drop('body2grams')
#df=df.drop('body3grams')

# Create 2grams
#ngram = NGram(n=2, inputCol="words", outputCol="body2grams")
#df = ngram.transform(df)

# Create 3grams
#ngram = NGram(n=3, inputCol="words", outputCol="body3grams")
#df = ngram.transform(df)
```

[28]:
```
# NOT USED since scoreSentiment is multinomial response not predictor
# OneHotEncoding of Score_sentiment
# since it is already numeric, no need for StringIndexer
#encoder = OneHotEncoder(inputCol="score_sentiment",␣
 ↪outputCol="scoreSentimentVec")
#model = encoder.fit(df)
#df = model.transform(df)
```

[ ]:
```

```

2.1 Save notebook as PDF document

[76]:
```
# Save notebook as PDF document
!jupyter nbconvert --to pdf `pwd`/*.ipynb
```

```
[NbConvertApp] Converting notebook
/sfs/qumulo/qhome/jdd5dw/ds5110-project/Jeremey_code.ipynb to pdf
[NbConvertApp] Writing 57104 bytes to notebook.tex
[NbConvertApp] Building PDF
[NbConvertApp] Running xelatex 3 times: ['xelatex', 'notebook.tex', '-quiet']
```

```
[NbConvertApp] Running bibtex 1 time: ['bibtex', 'notebook']
[NbConvertApp] WARNING | bibtex had problems, most likely because there were no
citations
[NbConvertApp] PDF successfully created
[NbConvertApp] Writing 68569 bytes to
/sfs/qumulo/qhome/jdd5dw/ds5110-project/Jeremey_code.pdf
[NbConvertApp] Converting notebook
/sfs/qumulo/qhome/jdd5dw/ds5110-project/test_file.ipynb to pdf
[NbConvertApp] Writing 26544 bytes to notebook.tex
[NbConvertApp] Building PDF
[NbConvertApp] Running xelatex 3 times: ['xelatex', 'notebook.tex', '-quiet']
[NbConvertApp] CRITICAL | xelatex failed: ['xelatex', 'notebook.tex', '-quiet']
This is XeTeX, Version 3.14159265-2.6-0.99999 (TeX Live 2019/dev/Debian)
(preloaded format=xelatex)
 restricted \write18 enabled.
entering extended mode
(./notebook.tex
LaTeX2e <2018-12-01>
(/usr/share/texlive/texmf-dist/tex/latex/base/article.cls
Document Class: article 2018/09/03 v1.4i Standard LaTeX document class
(/usr/share/texlive/texmf-dist/tex/latex/base/size11.clo))
(/usr/share/texlive/texmf-dist/tex/latex/tcolorbox/tcolorbox.sty
(/usr/share/texlive/texmf-dist/tex/latex/pgf/basiclayer/pgf.sty
(/usr/share/texlive/texmf-dist/tex/latex/pgf/utilities/pgfrcs.sty
(/usr/share/texlive/texmf-dist/tex/generic/pgf/utilities/pgfutil-common.tex
(/usr/share/texlive/texmf-dist/tex/generic/pgf/utilities/pgfutil-common-lists.t
ex)) (/usr/share/texlive/texmf-dist/tex/generic/pgf/utilities/pgfutil-latex.def
(/usr/share/texlive/texmf-dist/tex/latex/ms/everyshi.sty))
(/usr/share/texlive/texmf-dist/tex/generic/pgf/utilities/pgfrcs.code.tex
(/usr/share/texlive/texmf-dist/tex/generic/pgf/pgf.revision.tex)))
(/usr/share/texlive/texmf-dist/tex/latex/pgf/basiclayer/pgfcore.sty
(/usr/share/texlive/texmf-dist/tex/latex/graphics/graphicx.sty
(/usr/share/texlive/texmf-dist/tex/latex/graphics/keyval.sty)
(/usr/share/texlive/texmf-dist/tex/latex/graphics/graphics.sty
(/usr/share/texlive/texmf-dist/tex/latex/graphics/trig.sty)
(/usr/share/texlive/texmf-dist/tex/latex/graphics-cfg/graphics.cfg)
(/usr/share/texlive/texmf-dist/tex/latex/graphics-def/xetex.def)))
(/usr/share/texlive/texmf-dist/tex/latex/pgf/systemlayer/pgfsys.sty
(/usr/share/texlive/texmf-dist/tex/generic/pgf/systemlayer/pgfsys.code.tex
(/usr/share/texlive/texmf-dist/tex/generic/pgf/utilities/pgfkeys.code.tex
(/usr/share/texlive/texmf-dist/tex/generic/pgf/utilities/pgfkeysfiltered.code.t
ex)) (/usr/share/texlive/texmf-dist/tex/generic/pgf/systemlayer/pgf.cfg)
(/usr/share/texlive/texmf-dist/tex/generic/pgf/systemlayer/pgfsys-xetex.def
(/usr/share/texlive/texmf-dist/tex/generic/pgf/systemlayer/pgfsys-dvipdfmx.def
(/usr/share/texlive/texmf-dist/tex/generic/pgf/systemlayer/pgfsys-common-pdf.de
f))))
(/usr/share/texlive/texmf-dist/tex/generic/pgf/systemlayer/pgfsyssoftpath.code.
tex)
```

(/usr/share/texlive/texmf-dist/tex/generic/pgf/systemlayer/pgfsysprotocol.code.
tex)) (/usr/share/texlive/texmf-dist/tex/latex/xcolor/xcolor.sty
(/usr/share/texlive/texmf-dist/tex/latex/graphics-cfg/color.cfg))
(/usr/share/texlive/texmf-dist/tex/generic/pgf/basiclayer/pgfcore.code.tex
(/usr/share/texlive/texmf-dist/tex/generic/pgf/math/pgfmath.code.tex
(/usr/share/texlive/texmf-dist/tex/generic/pgf/math/pgfmathcalc.code.tex
(/usr/share/texlive/texmf-dist/tex/generic/pgf/math/pgfmathutil.code.tex)
(/usr/share/texlive/texmf-dist/tex/generic/pgf/math/pgfmathparser.code.tex)
(/usr/share/texlive/texmf-dist/tex/generic/pgf/math/pgfmathfunctions.code.tex
(/usr/share/texlive/texmf-dist/tex/generic/pgf/math/pgfmathfunctions.basic.code
.tex)
(/usr/share/texlive/texmf-dist/tex/generic/pgf/math/pgfmathfunctions.trigonomet
ric.code.tex)
(/usr/share/texlive/texmf-dist/tex/generic/pgf/math/pgfmathfunctions.random.cod
e.tex)
(/usr/share/texlive/texmf-dist/tex/generic/pgf/math/pgfmathfunctions.comparison
.code.tex)
(/usr/share/texlive/texmf-dist/tex/generic/pgf/math/pgfmathfunctions.base.code.
tex)
(/usr/share/texlive/texmf-dist/tex/generic/pgf/math/pgfmathfunctions.round.code
.tex)
(/usr/share/texlive/texmf-dist/tex/generic/pgf/math/pgfmathfunctions.misc.code.
tex)
(/usr/share/texlive/texmf-dist/tex/generic/pgf/math/pgfmathfunctions.integerari
thmetics.code.tex)))
(/usr/share/texlive/texmf-dist/tex/generic/pgf/math/pgfmathfloat.code.tex))
(/usr/share/texlive/texmf-dist/tex/generic/pgf/basiclayer/pgfcorepoints.code.te
x)
(/usr/share/texlive/texmf-dist/tex/generic/pgf/basiclayer/pgfcorepathconstruct.
code.tex)
(/usr/share/texlive/texmf-dist/tex/generic/pgf/basiclayer/pgfcorepathusage.code
.tex)
(/usr/share/texlive/texmf-dist/tex/generic/pgf/basiclayer/pgfcorescopes.code.te
x)
(/usr/share/texlive/texmf-dist/tex/generic/pgf/basiclayer/pgfcoregraphicstate.c
ode.tex)
(/usr/share/texlive/texmf-dist/tex/generic/pgf/basiclayer/pgfcoretransformation
s.code.tex)
(/usr/share/texlive/texmf-dist/tex/generic/pgf/basiclayer/pgfcorequick.code.tex
)
(/usr/share/texlive/texmf-dist/tex/generic/pgf/basiclayer/pgfcoreobjects.code.t
ex)
(/usr/share/texlive/texmf-dist/tex/generic/pgf/basiclayer/pgfcorepathprocessing
.code.tex)
(/usr/share/texlive/texmf-dist/tex/generic/pgf/basiclayer/pgfcorearrows.code.te
x)
(/usr/share/texlive/texmf-dist/tex/generic/pgf/basiclayer/pgfcoreshade.code.tex
)

```
(/usr/share/texlive/texmf-dist/tex/generic/pgf/basiclayer/pgfcoreimage.code.tex

(/usr/share/texlive/texmf-dist/tex/generic/pgf/basiclayer/pgfcoreexternal.code.
tex))
(/usr/share/texlive/texmf-dist/tex/generic/pgf/basiclayer/pgfcorelayers.code.te
x)
(/usr/share/texlive/texmf-dist/tex/generic/pgf/basiclayer/pgfcoretransparency.c
ode.tex)
(/usr/share/texlive/texmf-dist/tex/generic/pgf/basiclayer/pgfcorepatterns.code.
tex)
(/usr/share/texlive/texmf-dist/tex/generic/pgf/basiclayer/pgfcorerdf.code.tex))
)
(/usr/share/texlive/texmf-dist/tex/generic/pgf/modules/pgfmoduleshapes.code.tex
) (/usr/share/texlive/texmf-dist/tex/generic/pgf/modules/pgfmoduleplot.code.tex
)
(/usr/share/texlive/texmf-dist/tex/latex/pgf/compatibility/pgfcomp-version-0-65
.sty)
(/usr/share/texlive/texmf-dist/tex/latex/pgf/compatibility/pgfcomp-version-1-18
.sty)) (/usr/share/texlive/texmf-dist/tex/latex/tools/verbatim.sty)
(/usr/share/texlive/texmf-dist/tex/latex/environ/environ.sty
(/usr/share/texlive/texmf-dist/tex/latex/trimspaces/trimspaces.sty))
(/usr/share/texlive/texmf-dist/tex/latex/etoolbox/etoolbox.sty)
(/usr/share/texlive/texmf-dist/tex/latex/tcolorbox/tcbbreakable.code.tex
Library (tcolorbox): 'tcbbreakable.code.tex' version '4.15'
)) (/usr/share/texlive/texmf-dist/tex/latex/parskip/parskip.sty
(/usr/share/texlive/texmf-dist/tex/latex/oberdiek/kvoptions.sty
(/usr/share/texlive/texmf-dist/tex/generic/oberdiek/ltxcmds.sty)
(/usr/share/texlive/texmf-dist/tex/generic/oberdiek/kvsetkeys.sty
(/usr/share/texlive/texmf-dist/tex/generic/oberdiek/infwarerr.sty)
(/usr/share/texlive/texmf-dist/tex/generic/oberdiek/etexcmds.sty
(/usr/share/texlive/texmf-dist/tex/generic/oberdiek/ifluatex.sty)))))
(/usr/share/texlive/texmf-dist/tex/generic/iftex/iftex.sty)
(/usr/share/texlive/texmf-dist/tex/latex/fontspec/fontspec.sty
(/usr/share/texlive/texmf-dist/tex/latex/l3packages/xparse/xparse.sty
(/usr/share/texlive/texmf-dist/tex/latex/l3kernel/expl3.sty
(/usr/share/texlive/texmf-dist/tex/latex/l3kernel/expl3-code.tex)
(/usr/share/texlive/texmf-dist/tex/latex/l3kernel/l3xdvipdfmx.def)))
(/usr/share/texlive/texmf-dist/tex/latex/fontspec/fontspec-xetex.sty
(/usr/share/texlive/texmf-dist/tex/latex/base/fontenc.sty
(/usr/share/texlive/texmf-dist/tex/latex/base/tuenc.def))
(/usr/share/texlive/texmf-dist/tex/latex/fontspec/fontspec.cfg)))
(/usr/share/texlive/texmf-dist/tex/latex/caption/caption.sty
(/usr/share/texlive/texmf-dist/tex/latex/caption/caption3.sty))
(/usr/share/texlive/texmf-dist/tex/latex/float/float.sty)
(/usr/share/texlive/texmf-dist/tex/latex/tools/enumerate.sty)
(/usr/share/texlive/texmf-dist/tex/latex/geometry/geometry.sty
(/usr/share/texlive/texmf-dist/tex/generic/oberdiek/ifpdf.sty)
(/usr/share/texlive/texmf-dist/tex/generic/oberdiek/ifvtex.sty)
```

```
(/usr/share/texlive/texmf-dist/tex/generic/ifxetex/ifxetex.sty))
(/usr/share/texlive/texmf-dist/tex/latex/amsmath/amsmath.sty
For additional information on amsmath, use the `?' option.
(/usr/share/texlive/texmf-dist/tex/latex/amsmath/amstext.sty
(/usr/share/texlive/texmf-dist/tex/latex/amsmath/amsgen.sty))
(/usr/share/texlive/texmf-dist/tex/latex/amsmath/amsbsy.sty)
(/usr/share/texlive/texmf-dist/tex/latex/amsmath/amsopn.sty))
(/usr/share/texlive/texmf-dist/tex/latex/amsfonts/amssymb.sty
(/usr/share/texlive/texmf-dist/tex/latex/amsfonts/amsfonts.sty))
(/usr/share/texlive/texmf-dist/tex/latex/base/textcomp.sty
(/usr/share/texlive/texmf-dist/tex/latex/base/ts1enc.def))
(/usr/share/texlive/texmf-dist/tex/latex/upquote/upquote.sty)
(/usr/share/texlive/texmf-dist/tex/latex/eurosym/eurosym.sty)
(/usr/share/texlive/texmf-dist/tex/latex/ucs/ucs.sty
(/usr/share/texlive/texmf-dist/tex/latex/ucs/data/uni-global.def))
(/usr/share/texlive/texmf-dist/tex/latex/fancyvrb/fancyvrb.sty
Style option: `fancyvrb' v3.2a <2019/01/15> (tvz))
(/usr/share/texlive/texmf-dist/tex/latex/oberdiek/grffile.sty
(/usr/share/texlive/texmf-dist/tex/generic/oberdiek/pdftexcmds.sty))
(/usr/share/texlive/texmf-dist/tex/latex/adjustbox/adjustbox.sty
(/usr/share/texlive/texmf-dist/tex/latex/xkeyval/xkeyval.sty
(/usr/share/texlive/texmf-dist/tex/generic/xkeyval/xkeyval.tex
(/usr/share/texlive/texmf-dist/tex/generic/xkeyval/xkvutils.tex)))
(/usr/share/texlive/texmf-dist/tex/latex/adjustbox/adjcalc.sty)
(/usr/share/texlive/texmf-dist/tex/latex/adjustbox/trimclip.sty
(/usr/share/texlive/texmf-dist/tex/latex/collectbox/collectbox.sty)
(/usr/share/texlive/texmf-dist/tex/latex/adjustbox/tc-xetex.def))
(/usr/share/texlive/texmf-dist/tex/latex/ifoddpage/ifoddpage.sty)
(/usr/share/texlive/texmf-dist/tex/latex/varwidth/varwidth.sty))
(/usr/share/texlive/texmf-dist/tex/latex/hyperref/hyperref.sty
(/usr/share/texlive/texmf-dist/tex/generic/oberdiek/hobsub-hyperref.sty
(/usr/share/texlive/texmf-dist/tex/generic/oberdiek/hobsub-generic.sty))
(/usr/share/texlive/texmf-dist/tex/latex/oberdiek/auxhook.sty)
(/usr/share/texlive/texmf-dist/tex/latex/hyperref/pd1enc.def)
(/usr/share/texlive/texmf-dist/tex/latex/latexconfig/hyperref.cfg)
(/usr/share/texlive/texmf-dist/tex/latex/url/url.sty))
(/usr/share/texlive/texmf-dist/tex/latex/hyperref/hxetex.def
(/usr/share/texlive/texmf-dist/tex/latex/hyperref/puenc.def)
(/usr/share/texlive/texmf-dist/tex/generic/oberdiek/stringenc.sty)
(/usr/share/texlive/texmf-dist/tex/latex/oberdiek/rerunfilecheck.sty))
(/usr/share/texlive/texmf-dist/tex/latex/titling/titling.sty)
(/usr/share/texlive/texmf-dist/tex/latex/tools/longtable.sty)
(/usr/share/texlive/texmf-dist/tex/latex/booktabs/booktabs.sty)
(/usr/share/texlive/texmf-dist/tex/latex/enumitem/enumitem.sty)
(/usr/share/texlive/texmf-dist/tex/generic/ulem/ulem.sty)
(/usr/share/texlive/texmf-dist/tex/latex/jknapltx/mathrsfs.sty)
No file notebook.aux.
(/usr/share/texlive/texmf-dist/tex/latex/base/ts1cmr.fd)
```

```
ABD: EveryShipout initializing macros
(/usr/share/texlive/texmf-dist/tex/latex/caption/ltcaption.sty)
*geometry* driver: auto-detecting
*geometry* detected driver: xetex
*geometry* verbose mode - [ preamble ] result:
* driver: xetex
* paper: <default>
* layout: <same size as paper>
* layoutoffset:(h,v)=(0.0pt,0.0pt)
* modes:
* h-part:(L,W,R)=(72.26999pt, 469.75502pt, 72.26999pt)
* v-part:(T,H,B)=(72.26999pt, 650.43001pt, 72.26999pt)
* \paperwidth=614.295pt
* \paperheight=794.96999pt
* \textwidth=469.75502pt
* \textheight=650.43001pt
* \oddsidemargin=0.0pt
* \evensidemargin=0.0pt
* \topmargin=-37.0pt
* \headheight=12.0pt
* \headsep=25.0pt
* \topskip=11.0pt
* \footskip=30.0pt
* \marginparwidth=59.0pt
* \marginparsep=10.0pt
* \columnsep=10.0pt
* \skip\footins=10.0pt plus 4.0pt minus 2.0pt
* \hoffset=0.0pt
* \voffset=0.0pt
* \mag=1000
* \@twocolumnfalse
* \@twosidefalse
* \@mparswitchfalse
* \@reversemarginfalse
* (1in=72.27pt=25.4mm, 1cm=28.453pt)

(/usr/share/texlive/texmf-dist/tex/latex/ucs/ucsencs.def)
(/usr/share/texlive/texmf-dist/tex/latex/hyperref/nameref.sty
(/usr/share/texlive/texmf-dist/tex/generic/oberdiek/gettitlestring.sty))

Package hyperref Warning: Rerun to get /PageLabels entry.

(/usr/share/texlive/texmf-dist/tex/latex/amsfonts/umsa.fd)
(/usr/share/texlive/texmf-dist/tex/latex/amsfonts/umsb.fd)
(/usr/share/texlive/texmf-dist/tex/latex/jknapltx/ursfs.fd)

LaTeX Warning: No \author given.
```

```
(/usr/share/texlive/texmf-dist/tex/generic/oberdiek/se-ascii-print.def)

LaTeX Warning: File `attachment:27d67016-1a87-45ec-acc8-cf6a498607c6.png' not f
ound on input line 398.

! Unable to load picture or PDF file 'attachment:27d67016-1a87-45ec-acc8-cf6a49
8607c6.png'.
<to be read again>
                   }
l.398 …27d67016-1a87-45ec-acc8-cf6a498607c6.png}

?
! Emergency stop.
<to be read again>
                   }
l.398 …27d67016-1a87-45ec-acc8-cf6a498607c6.png}

No pages of output.
Transcript written on notebook.log.

Traceback (most recent call last):
  File "/opt/conda/bin/jupyter-nbconvert", line 11, in <module>
    sys.exit(main())
  File "/opt/conda/lib/python3.7/site-packages/jupyter_core/application.py",
line 254, in launch_instance
    return super(JupyterApp, cls).launch_instance(argv=argv, **kwargs)
  File "/opt/conda/lib/python3.7/site-packages/traitlets/config/application.py",
line 845, in launch_instance
    app.start()
  File "/opt/conda/lib/python3.7/site-packages/nbconvert/nbconvertapp.py", line
350, in start
    self.convert_notebooks()
  File "/opt/conda/lib/python3.7/site-packages/nbconvert/nbconvertapp.py", line
524, in convert_notebooks
    self.convert_single_notebook(notebook_filename)
  File "/opt/conda/lib/python3.7/site-packages/nbconvert/nbconvertapp.py", line
489, in convert_single_notebook
    output, resources = self.export_single_notebook(notebook_filename,
resources, input_buffer=input_buffer)
  File "/opt/conda/lib/python3.7/site-packages/nbconvert/nbconvertapp.py", line
418, in export_single_notebook
    output, resources = self.exporter.from_filename(notebook_filename,
resources=resources)
  File "/opt/conda/lib/python3.7/site-packages/nbconvert/exporters/exporter.py",
line 181, in from_filename
    return self.from_file(f, resources=resources, **kw)
  File "/opt/conda/lib/python3.7/site-packages/nbconvert/exporters/exporter.py",
line 199, in from_file
```

```
    return self.from_notebook_node(nbformat.read(file_stream, as_version=4),
resources=resources, **kw)
  File "/opt/conda/lib/python3.7/site-packages/nbconvert/exporters/pdf.py", line
183, in from_notebook_node
    self.run_latex(tex_file)
  File "/opt/conda/lib/python3.7/site-packages/nbconvert/exporters/pdf.py", line
154, in run_latex
    self.latex_count, log_error, raise_on_failure)
  File "/opt/conda/lib/python3.7/site-packages/nbconvert/exporters/pdf.py", line
143, in run_command
    command=command, output=out))
nbconvert.exporters.pdf.LatexFailed: PDF creating failed, captured latex output:
Failed to run "['xelatex', 'notebook.tex', '-quiet']" command:
This is XeTeX, Version 3.14159265-2.6-0.99999 (TeX Live 2019/dev/Debian)
(preloaded format=xelatex)
 restricted \write18 enabled.
entering extended mode
(./notebook.tex
LaTeX2e <2018-12-01>
(/usr/share/texlive/texmf-dist/tex/latex/base/article.cls
Document Class: article 2018/09/03 v1.4i Standard LaTeX document class
(/usr/share/texlive/texmf-dist/tex/latex/base/size11.clo))
(/usr/share/texlive/texmf-dist/tex/latex/tcolorbox/tcolorbox.sty
(/usr/share/texlive/texmf-dist/tex/latex/pgf/basiclayer/pgf.sty
(/usr/share/texlive/texmf-dist/tex/latex/pgf/utilities/pgfrcs.sty
(/usr/share/texlive/texmf-dist/tex/generic/pgf/utilities/pgfutil-common.tex
(/usr/share/texlive/texmf-dist/tex/generic/pgf/utilities/pgfutil-common-lists.t
ex)) (/usr/share/texlive/texmf-dist/tex/generic/pgf/utilities/pgfutil-latex.def
(/usr/share/texlive/texmf-dist/tex/latex/ms/everyshi.sty))
(/usr/share/texlive/texmf-dist/tex/generic/pgf/utilities/pgfrcs.code.tex
(/usr/share/texlive/texmf-dist/tex/generic/pgf/pgf.revision.tex)))
(/usr/share/texlive/texmf-dist/tex/latex/pgf/basiclayer/pgfcore.sty
(/usr/share/texlive/texmf-dist/tex/latex/graphics/graphicx.sty
(/usr/share/texlive/texmf-dist/tex/latex/graphics/keyval.sty)
(/usr/share/texlive/texmf-dist/tex/latex/graphics/graphics.sty
(/usr/share/texlive/texmf-dist/tex/latex/graphics/trig.sty)
(/usr/share/texlive/texmf-dist/tex/latex/graphics-cfg/graphics.cfg)
(/usr/share/texlive/texmf-dist/tex/latex/graphics-def/xetex.def)))
(/usr/share/texlive/texmf-dist/tex/latex/pgf/systemlayer/pgfsys.sty
(/usr/share/texlive/texmf-dist/tex/generic/pgf/systemlayer/pgfsys.code.tex
(/usr/share/texlive/texmf-dist/tex/generic/pgf/utilities/pgfkeys.code.tex
(/usr/share/texlive/texmf-dist/tex/generic/pgf/utilities/pgfkeysfiltered.code.t
ex)) (/usr/share/texlive/texmf-dist/tex/generic/pgf/systemlayer/pgf.cfg)
(/usr/share/texlive/texmf-dist/tex/generic/pgf/systemlayer/pgfsys-xetex.def
(/usr/share/texlive/texmf-dist/tex/generic/pgf/systemlayer/pgfsys-dvipdfmx.def
(/usr/share/texlive/texmf-dist/tex/generic/pgf/systemlayer/pgfsys-common-pdf.de
f))))
(/usr/share/texlive/texmf-dist/tex/generic/pgf/systemlayer/pgfsyssoftpath.code.
```

tex)
(/usr/share/texlive/texmf-dist/tex/generic/pgf/systemlayer/pgfsysprotocol.code.
tex)) (/usr/share/texlive/texmf-dist/tex/latex/xcolor/xcolor.sty
(/usr/share/texlive/texmf-dist/tex/latex/graphics-cfg/color.cfg))
(/usr/share/texlive/texmf-dist/tex/generic/pgf/basiclayer/pgfcore.code.tex
(/usr/share/texlive/texmf-dist/tex/generic/pgf/math/pgfmath.code.tex
(/usr/share/texlive/texmf-dist/tex/generic/pgf/math/pgfmathcalc.code.tex
(/usr/share/texlive/texmf-dist/tex/generic/pgf/math/pgfmathutil.code.tex)
(/usr/share/texlive/texmf-dist/tex/generic/pgf/math/pgfmathparser.code.tex)
(/usr/share/texlive/texmf-dist/tex/generic/pgf/math/pgfmathfunctions.code.tex
(/usr/share/texlive/texmf-dist/tex/generic/pgf/math/pgfmathfunctions.basic.code
.tex)
(/usr/share/texlive/texmf-dist/tex/generic/pgf/math/pgfmathfunctions.trigonomet
ric.code.tex)
(/usr/share/texlive/texmf-dist/tex/generic/pgf/math/pgfmathfunctions.random.cod
e.tex)
(/usr/share/texlive/texmf-dist/tex/generic/pgf/math/pgfmathfunctions.comparison
.code.tex)
(/usr/share/texlive/texmf-dist/tex/generic/pgf/math/pgfmathfunctions.base.code.
tex)
(/usr/share/texlive/texmf-dist/tex/generic/pgf/math/pgfmathfunctions.round.code
.tex)
(/usr/share/texlive/texmf-dist/tex/generic/pgf/math/pgfmathfunctions.misc.code.
tex)
(/usr/share/texlive/texmf-dist/tex/generic/pgf/math/pgfmathfunctions.integerari
thmetics.code.tex)))
(/usr/share/texlive/texmf-dist/tex/generic/pgf/math/pgfmathfloat.code.tex))
(/usr/share/texlive/texmf-dist/tex/generic/pgf/basiclayer/pgfcorepoints.code.te
x)
(/usr/share/texlive/texmf-dist/tex/generic/pgf/basiclayer/pgfcorepathconstruct.
code.tex)
(/usr/share/texlive/texmf-dist/tex/generic/pgf/basiclayer/pgfcorepathusage.code
.tex)
(/usr/share/texlive/texmf-dist/tex/generic/pgf/basiclayer/pgfcorescopes.code.te
x)
(/usr/share/texlive/texmf-dist/tex/generic/pgf/basiclayer/pgfcoregraphicstate.c
ode.tex)
(/usr/share/texlive/texmf-dist/tex/generic/pgf/basiclayer/pgfcoretransformation
s.code.tex)
(/usr/share/texlive/texmf-dist/tex/generic/pgf/basiclayer/pgfcorequick.code.tex
)
(/usr/share/texlive/texmf-dist/tex/generic/pgf/basiclayer/pgfcoreobjects.code.t
ex)
(/usr/share/texlive/texmf-dist/tex/generic/pgf/basiclayer/pgfcorepathprocessing
.code.tex)
(/usr/share/texlive/texmf-dist/tex/generic/pgf/basiclayer/pgfcorearrows.code.te
x)
(/usr/share/texlive/texmf-dist/tex/generic/pgf/basiclayer/pgfcoreshade.code.tex

```
)
(/usr/share/texlive/texmf-dist/tex/generic/pgf/basiclayer/pgfcoreimage.code.tex

(/usr/share/texlive/texmf-dist/tex/generic/pgf/basiclayer/pgfcoreexternal.code.
tex))
(/usr/share/texlive/texmf-dist/tex/generic/pgf/basiclayer/pgfcorelayers.code.te
x)
(/usr/share/texlive/texmf-dist/tex/generic/pgf/basiclayer/pgfcoretransparency.c
ode.tex)
(/usr/share/texlive/texmf-dist/tex/generic/pgf/basiclayer/pgfcorepatterns.code.
tex)
(/usr/share/texlive/texmf-dist/tex/generic/pgf/basiclayer/pgfcorerdf.code.tex))
)
(/usr/share/texlive/texmf-dist/tex/generic/pgf/modules/pgfmoduleshapes.code.tex
) (/usr/share/texlive/texmf-dist/tex/generic/pgf/modules/pgfmoduleplot.code.tex
)
(/usr/share/texlive/texmf-dist/tex/latex/pgf/compatibility/pgfcomp-version-0-65
.sty)
(/usr/share/texlive/texmf-dist/tex/latex/pgf/compatibility/pgfcomp-version-1-18
.sty)) (/usr/share/texlive/texmf-dist/tex/latex/tools/verbatim.sty)
(/usr/share/texlive/texmf-dist/tex/latex/environ/environ.sty
(/usr/share/texlive/texmf-dist/tex/latex/trimspaces/trimspaces.sty))
(/usr/share/texlive/texmf-dist/tex/latex/etoolbox/etoolbox.sty)
(/usr/share/texlive/texmf-dist/tex/latex/tcolorbox/tcbbreakable.code.tex
Library (tcolorbox): 'tcbbreakable.code.tex' version '4.15'
)) (/usr/share/texlive/texmf-dist/tex/latex/parskip/parskip.sty
(/usr/share/texlive/texmf-dist/tex/latex/oberdiek/kvoptions.sty
(/usr/share/texlive/texmf-dist/tex/generic/oberdiek/ltxcmds.sty)
(/usr/share/texlive/texmf-dist/tex/generic/oberdiek/kvsetkeys.sty
(/usr/share/texlive/texmf-dist/tex/generic/oberdiek/infwarerr.sty)
(/usr/share/texlive/texmf-dist/tex/generic/oberdiek/etexcmds.sty
(/usr/share/texlive/texmf-dist/tex/generic/oberdiek/ifluatex.sty)))))
(/usr/share/texlive/texmf-dist/tex/generic/iftex/iftex.sty)
(/usr/share/texlive/texmf-dist/tex/latex/fontspec/fontspec.sty
(/usr/share/texlive/texmf-dist/tex/latex/l3packages/xparse/xparse.sty
(/usr/share/texlive/texmf-dist/tex/latex/l3kernel/expl3.sty
(/usr/share/texlive/texmf-dist/tex/latex/l3kernel/expl3-code.tex)
(/usr/share/texlive/texmf-dist/tex/latex/l3kernel/l3xdvipdfmx.def)))
(/usr/share/texlive/texmf-dist/tex/latex/fontspec/fontspec-xetex.sty
(/usr/share/texlive/texmf-dist/tex/latex/base/fontenc.sty
(/usr/share/texlive/texmf-dist/tex/latex/base/tuenc.def))
(/usr/share/texlive/texmf-dist/tex/latex/fontspec/fontspec.cfg)))
(/usr/share/texlive/texmf-dist/tex/latex/caption/caption.sty
(/usr/share/texlive/texmf-dist/tex/latex/caption/caption3.sty))
(/usr/share/texlive/texmf-dist/tex/latex/float/float.sty)
(/usr/share/texlive/texmf-dist/tex/latex/tools/enumerate.sty)
(/usr/share/texlive/texmf-dist/tex/latex/geometry/geometry.sty
(/usr/share/texlive/texmf-dist/tex/generic/oberdiek/ifpdf.sty)
```

```
(/usr/share/texlive/texmf-dist/tex/generic/oberdiek/ifvtex.sty)
(/usr/share/texlive/texmf-dist/tex/generic/ifxetex/ifxetex.sty))
(/usr/share/texlive/texmf-dist/tex/latex/amsmath/amsmath.sty
For additional information on amsmath, use the `?' option.
(/usr/share/texlive/texmf-dist/tex/latex/amsmath/amstext.sty
(/usr/share/texlive/texmf-dist/tex/latex/amsmath/amsgen.sty))
(/usr/share/texlive/texmf-dist/tex/latex/amsmath/amsbsy.sty)
(/usr/share/texlive/texmf-dist/tex/latex/amsmath/amsopn.sty))
(/usr/share/texlive/texmf-dist/tex/latex/amsfonts/amssymb.sty
(/usr/share/texlive/texmf-dist/tex/latex/amsfonts/amsfonts.sty))
(/usr/share/texlive/texmf-dist/tex/latex/base/textcomp.sty
(/usr/share/texlive/texmf-dist/tex/latex/base/ts1enc.def))
(/usr/share/texlive/texmf-dist/tex/latex/upquote/upquote.sty)
(/usr/share/texlive/texmf-dist/tex/latex/eurosym/eurosym.sty)
(/usr/share/texlive/texmf-dist/tex/latex/ucs/ucs.sty
(/usr/share/texlive/texmf-dist/tex/latex/ucs/data/uni-global.def))
(/usr/share/texlive/texmf-dist/tex/latex/fancyvrb/fancyvrb.sty
Style option: `fancyvrb' v3.2a <2019/01/15> (tvz))
(/usr/share/texlive/texmf-dist/tex/latex/oberdiek/grffile.sty
(/usr/share/texlive/texmf-dist/tex/generic/oberdiek/pdftexcmds.sty))
(/usr/share/texlive/texmf-dist/tex/latex/adjustbox/adjustbox.sty
(/usr/share/texlive/texmf-dist/tex/latex/xkeyval/xkeyval.sty
(/usr/share/texlive/texmf-dist/tex/generic/xkeyval/xkeyval.tex
(/usr/share/texlive/texmf-dist/tex/generic/xkeyval/xkvutils.tex)))
(/usr/share/texlive/texmf-dist/tex/latex/adjustbox/adjcalc.sty)
(/usr/share/texlive/texmf-dist/tex/latex/adjustbox/trimclip.sty
(/usr/share/texlive/texmf-dist/tex/latex/collectbox/collectbox.sty)
(/usr/share/texlive/texmf-dist/tex/latex/adjustbox/tc-xetex.def))
(/usr/share/texlive/texmf-dist/tex/latex/ifoddpage/ifoddpage.sty)
(/usr/share/texlive/texmf-dist/tex/latex/varwidth/varwidth.sty))
(/usr/share/texlive/texmf-dist/tex/latex/hyperref/hyperref.sty
(/usr/share/texlive/texmf-dist/tex/generic/oberdiek/hobsub-hyperref.sty
(/usr/share/texlive/texmf-dist/tex/generic/oberdiek/hobsub-generic.sty))
(/usr/share/texlive/texmf-dist/tex/latex/oberdiek/auxhook.sty)
(/usr/share/texlive/texmf-dist/tex/latex/hyperref/pd1enc.def)
(/usr/share/texlive/texmf-dist/tex/latex/latexconfig/hyperref.cfg)
(/usr/share/texlive/texmf-dist/tex/latex/url/url.sty))
(/usr/share/texlive/texmf-dist/tex/latex/hyperref/hxetex.def
(/usr/share/texlive/texmf-dist/tex/latex/hyperref/puenc.def)
(/usr/share/texlive/texmf-dist/tex/generic/oberdiek/stringenc.sty)
(/usr/share/texlive/texmf-dist/tex/latex/oberdiek/rerunfilecheck.sty))
(/usr/share/texlive/texmf-dist/tex/latex/titling/titling.sty)
(/usr/share/texlive/texmf-dist/tex/latex/tools/longtable.sty)
(/usr/share/texlive/texmf-dist/tex/latex/booktabs/booktabs.sty)
(/usr/share/texlive/texmf-dist/tex/latex/enumitem/enumitem.sty)
(/usr/share/texlive/texmf-dist/tex/generic/ulem/ulem.sty)
(/usr/share/texlive/texmf-dist/tex/latex/jknapltx/mathrsfs.sty)
No file notebook.aux.
```

```
(/usr/share/texlive/texmf-dist/tex/latex/base/ts1cmr.fd)
ABD: EveryShipout initializing macros
(/usr/share/texlive/texmf-dist/tex/latex/caption/ltcaption.sty)
*geometry* driver: auto-detecting
*geometry* detected driver: xetex
*geometry* verbose mode - [ preamble ] result:
* driver: xetex
* paper: <default>
* layout: <same size as paper>
* layoutoffset:(h,v)=(0.0pt,0.0pt)
* modes:
* h-part:(L,W,R)=(72.26999pt, 469.75502pt, 72.26999pt)
* v-part:(T,H,B)=(72.26999pt, 650.43001pt, 72.26999pt)
* \paperwidth=614.295pt
* \paperheight=794.96999pt
* \textwidth=469.75502pt
* \textheight=650.43001pt
* \oddsidemargin=0.0pt
* \evensidemargin=0.0pt
* \topmargin=-37.0pt
* \headheight=12.0pt
* \headsep=25.0pt
* \topskip=11.0pt
* \footskip=30.0pt
* \marginparwidth=59.0pt
* \marginparsep=10.0pt
* \columnsep=10.0pt
* \skip\footins=10.0pt plus 4.0pt minus 2.0pt
* \hoffset=0.0pt
* \voffset=0.0pt
* \mag=1000
* \@twocolumnfalse
* \@twosidefalse
* \@mparswitchfalse
* \@reversemarginfalse
* (1in=72.27pt=25.4mm, 1cm=28.453pt)

(/usr/share/texlive/texmf-dist/tex/latex/ucs/ucsencs.def)
(/usr/share/texlive/texmf-dist/tex/latex/hyperref/nameref.sty
(/usr/share/texlive/texmf-dist/tex/generic/oberdiek/gettitlestring.sty))

Package hyperref Warning: Rerun to get /PageLabels entry.

(/usr/share/texlive/texmf-dist/tex/latex/amsfonts/umsa.fd)
(/usr/share/texlive/texmf-dist/tex/latex/amsfonts/umsb.fd)
(/usr/share/texlive/texmf-dist/tex/latex/jknapltx/ursfs.fd)

LaTeX Warning: No \author given.
```

```
(/usr/share/texlive/texmf-dist/tex/generic/oberdiek/se-ascii-print.def)

LaTeX Warning: File `attachment:27d67016-1a87-45ec-acc8-cf6a498607c6.png' not f
ound on input line 398.

! Unable to load picture or PDF file 'attachment:27d67016-1a87-45ec-acc8-cf6a49
8607c6.png'.
<to be read again>
                   }
l.398 …27d67016-1a87-45ec-acc8-cf6a498607c6.png}

?
! Emergency stop.
<to be read again>
                   }
l.398 …27d67016-1a87-45ec-acc8-cf6a498607c6.png}

No pages of output.
Transcript written on notebook.log.
```

[ ]: