

```
1 from google.colab import drive
```

```
1 drive.mount('/content/drive')
```

Mounted at /content/drive

```
1 !apt-get update
```

```
Ign:1 https://developer.download.nvidia.com/compute/cuda/repos/ubuntu1804/x86_64 InRelease
Get:2 https://cloud.r-project.org/bin/linux/ubuntu bionic-cran40/ InRelease [3,626 B]
Get:3 http://ppa.launchpad.net/c2d4u.team/c2d4u4.0+/ubuntu bionic InRelease [15.9 kB]
Ign:4 https://developer.download.nvidia.com/compute/machine-learning/repos/ubuntu1804/x86_64 InRelease
Get:5 http://security.ubuntu.com/ubuntu bionic-security InRelease [88.7 kB]
Get:6 https://developer.download.nvidia.com/compute/cuda/repos/ubuntu1804/x86_64 Release
Get:7 https://developer.download.nvidia.com/compute/machine-learning/repos/ubuntu1804/x86_64 Release
Hit:8 http://archive.ubuntu.com/ubuntu bionic InRelease
Get:9 https://developer.download.nvidia.com/compute/cuda/repos/ubuntu1804/x86_64 Release
Get:10 https://developer.download.nvidia.com/compute/machine-learning/repos/ubuntu1804/x86_64 Release
Get:11 http://archive.ubuntu.com/ubuntu bionic-updates InRelease [88.7 kB]
Get:12 http://ppa.launchpad.net/graphics-drivers/ppa/ubuntu bionic InRelease [21.3 kB]
Get:13 http://archive.ubuntu.com/ubuntu bionic-backports InRelease [74.6 kB]
Ign:14 https://developer.download.nvidia.com/compute/cuda/repos/ubuntu1804/x86_64 Packages
Get:14 https://developer.download.nvidia.com/compute/cuda/repos/ubuntu1804/x86_64 Packages
Get:15 https://developer.download.nvidia.com/compute/machine-learning/repos/ubuntu1804/x86_64 Packages
Get:16 http://ppa.launchpad.net/c2d4u.team/c2d4u4.0+/ubuntu bionic/main Sources [1,668 B]
Get:17 http://security.ubuntu.com/ubuntu bionic-security/universe amd64 Packages [1,312 B]
Get:18 http://archive.ubuntu.com/ubuntu bionic-updates/restricted amd64 Packages [24.7 kB]
Get:19 http://security.ubuntu.com/ubuntu bionic-security/main amd64 Packages [1,781 kB]
Get:20 http://archive.ubuntu.com/ubuntu bionic-updates/main amd64 Packages [2,198 kB]
Get:21 http://security.ubuntu.com/ubuntu bionic-security/multiverse amd64 Packages [1,312 B]
Get:22 http://security.ubuntu.com/ubuntu bionic-security/restricted amd64 Packages [1,312 B]
Get:23 http://ppa.launchpad.net/c2d4u.team/c2d4u4.0+/ubuntu bionic/main amd64 Packages [1,668 B]
Get:24 http://archive.ubuntu.com/ubuntu bionic-updates/multiverse amd64 Packages [46.1 kB]
Get:25 http://archive.ubuntu.com/ubuntu bionic-updates/universe amd64 Packages [2,129 kB]
Get:26 http://ppa.launchpad.net/graphics-drivers/ppa/ubuntu bionic/main amd64 Packages [1,668 B]
Fetched 11.4 MB in 3s (3,533 kB/s)
Reading package lists... Done
```

```
1 # Installing the spark and all related tools
```

```
2
```

```
3 !apt-get install openjdk-8-jdk-headless -qq > /dev/null
```

```
4 !wget -q https://www-us.apache.org/dist/spark/spark-3.0.1/spark-3.0.1-
```

```
5 !tar xf spark-3.0.1-bin-hadoop2.7.tgz
```

```
6 !pip install -q findspark
```

```
1 # setting the environment
```

```
2 import os
```

```
3 os.environ["JAVA_HOME"] = "/usr/lib/jvm/java-8-openjdk-amd64"
```

```
4 os.environ["SPARK_HOME"] = "/content/spark-3.0.3-bin-hadoop2.7"
```

```

1 # Initiating the spark on this notebook
2 import findspark

1 findspark.init('/content/spark-3.0.1-bin-hadoop2.7')

1 from pyspark.sql import SparkSession

```

▼ Spark

```

1 spark=SparkSession.builder.appName('final_project').getOrCreate()
2

```

▼ Reading the Data from Cloud

```

1 df=spark.read.csv('/content/drive/My Drive/Colab Notebooks/data/SMSSpa

1 df.printSchema()

root
 |-- _c0: string (nullable = true)
 |-- _c1: string (nullable = true)

1 df.show(50,truncate=False)

```

```

+----+-----+
|_c0|_c1|
+----+-----+
|ham|Go until jurong point, crazy.. Available only in bugis n great world la e buffe
|ham|Ok lar... Joking wif u oni...
|spam|Free entry in 2 a wkly comp to win FA Cup final tkts 21st May 2005. Text FA to
|ham|U dun say so early hor... U c already then say...
|ham|Nah I don't think he goes to usf, he lives around here though
|spam|FreeMsg Hey there darling it's been 3 week's now and no word back! I'd like son
|ham|Even my brother is not like to speak with me. They treat me like aids patent.
|ham|As per your request 'Melle Melle (Oru Minnaminunginte Nurungu Vettam)' has beer
|spam|WINNER!! As a valued network customer you have been selected to receive a £900 p
|spam|Had your mobile 11 months or more? U R entitled to Update to the latest colour
|ham|I'm gonna be home soon and i don't want to talk about this stuff anymore tonigh
|spam|SIX chances to win CASH! From 100 to 20,000 pounds txt> CSH11 and send to 87575
|spam|URGENT! You have won a 1 week FREE membership in our £100,000 Prize Jackpot! T
|ham|I've been searching for the right words to thank you for this breather. I promi
|ham|I HAVE A DATE ON SUNDAY WITH WILL!!
|spam|XXXMobileMovieClub: To use your credit, click the WAP link in the next txt mess
|ham|Oh k...i'm watching here:)
|ham|Eh u remember how 2 spell his name... Yes i did. He v naughty make until i v we
|ham|Fine if that's the way u feel. That's the way its gota b
|spam|England v Macedonia - dont miss the goals/team news. Txt ur national team to 87

```

```

|ham |Is that seriously how you spell his name?
|ham |I'm going to try for 2 months ha ha only joking
|ham |So ü pay first lar... Then when is da stock comin...
|ham |Aft i finish my lunch then i go str down lor. Ard 3 smth lor. U finish ur lunch
|ham |Fffffff. Alright no way I can meet up with you sooner?
|ham |Just forced myself to eat a slice. I'm really not hungry tho. This sucks. Mark
|ham |Lol your always so convincing.
|ham |Did you catch the bus ? Are you frying an egg ? Did you make a tea? Are you eat
|ham |I'm back & we're packing the car now, I'll let you know if there's room
|ham |Ahhh. Work. I vaguely remember that! What does it feel like? Lol
|ham |Wait that's still not all that clear, were you not sure about me being sarcastic
|ham |Yeah he got in at 2 and was v apologetic. n had fallen out and she was actin li
|ham |K tell me anything about you.
|ham |For fear of fainting with the of all that housework you just did? Quick have a
|spam|Thanks for your subscription to Ringtone UK your mobile will be charged £5/month
|ham |Yup... Ok i go home look at the timings then i msg ü again... Xuhui going to le
|ham |Oops, I'll let you know when my roommate's done
|ham |I see the letter B on my car
|ham |Anything lor... U decide...
|ham |Hello! How's you and how did saturday go? I was just texting to see if you'd de
|ham |Pls go ahead with watts. I just wanted to be sure. Do have a great weekend. Abi
|ham |Did I forget to tell you ? I want you , I need you, I crave you ... But most of
|spam|07732584351 - Rodger Burns - MSG = We tried to call you re your reply to our sn
|ham |WHO ARE YOU SEEING?
|ham |Great! I hope you like your man well endowed. I am <#> inches...
|ham |No calls..messages..missed calls
|ham |Didn't you get hep b immunisation in nigeria.
|ham |Fair enough, anything going on?
|ham |Yeah hopefully, if tyler can't do it I could maybe ask around a bit
|ham |U don't know how stubborn I am. I didn't even want to go to the hospital. I kep
+-----+
only showing top 50 rows

```

```
1 df.createOrReplaceTempView('table')
```

SQL Queries with Spark

- ▼ 1. How many rows are there is this data frame? 5574 rows.

```
1 spark.sql("select count(*) \
2         from table ").show()
```

```

+-----+
|count(1)|
+-----+
|    5574|
+-----+

```

▼ Clean and Prepare the Data

```
1 df=df.withColumnRenamed('_c0','class').withColumnRenamed('_c1','text')
```

```
1 df.show(truncate=False)
```

```
+-----+-----+
|class|text
+-----+-----+
|ham  |Go until jurong point, crazy.. Available only in bugis n great world la e bufi
|ham  |Ok lar... Joking wif u oni...
|spam |Free entry in 2 a wkly comp to win FA Cup final tkts 21st May 2005. Text FA to
|ham  |U dun say so early hor... U c already then say...
|ham  |Nah I don't think he goes to usf, he lives around here though
|spam |FreeMsg Hey there darling it's been 3 week's now and no word back! I'd like sc
|ham  |Even my brother is not like to speak with me. They treat me like aids patent.
|ham  |As per your request 'Melle Melle (Oru Minnaminunginte Nurungu Vettam)' has bee
|spam |WINNER!! As a valued network customer you have been selected to receive a £900
|spam |Had your mobile 11 months or more? U R entitled to Update to the latest colour
|ham  |I'm gonna be home soon and i don't want to talk about this stuff anymore tonig
|spam |SIX chances to win CASH! From 100 to 20,000 pounds txt> CSH11 and send to 8757
|spam |URGENT! You have won a 1 week FREE membership in our £100,000 Prize Jackpot! 1
|ham  |I've been searching for the right words to thank you for this breather. I prom
|ham  |I HAVE A DATE ON SUNDAY WITH WILL!!
|spam |XXXMobileMovieClub: To use your credit, click the WAP link in the next txt mes
|ham  |Oh k...i'm watching here:)
|ham  |Eh u remember how 2 spell his name... Yes i did. He v naughty make until i v v
|ham  |Fine if that's the way u feel. That's the way its gota b
|spam |England v Macedonia - dont miss the goals/team news. Txt ur national team to 8
+-----+-----+
only showing top 20 rows
```

```
1 df.createOrReplaceTempView('t')
```

More Queries

▼ 2. How many word "Love" were in data? 167

```
1 spark.sql("SELECT count(*) FROM t \
2         WHERE text LIKE '%love%';").show()
```

```
+-----+
|count(1)|
+-----+
|      167|
+-----+
```

3. How many sentences there were where they had "I love you " in that text in the data ? 19 messages had this sentences.

```
1 spark.sql("SELECT count(*) FROM t \
2         WHERE text LIKE '%I love you%';").show()
```

```
+-----+
|count(1)|
+-----+
|      19|
+-----+
```

4. What was the class detected that "I love you" sentences were in those texts? All were observed as a Ham sms.

```
1 spark.sql("SELECT * FROM t \
2         WHERE text LIKE '%I love you%';").show()
```

```
+-----+-----+
|class|      text|
+-----+-----+
| ham|Did I forget to t...|
| ham|Hello my boytoy ....|
| ham|Geeeee ... I love...|
| ham|I wonder if your ...|
| ham|Wow ... I love yo...|
| ham|I love you. You s...|
| ham|Do have a nice da...|
| ham|I love your ass! ...|
| ham|Hey ! I want you ...|
| ham|BABE !!! I miiiii...|
| ham|I thank you so mu...|
| ham|Mmmmm ... It was ...|
| ham|I love you both t...|
| ham|Hey loverboy! I l...|
| ham|I love you !!! Yo...|
| ham|I miss you so muc...|
| ham|Mmmmmm ... I love...|
| ham|Don't forget thou...|
| ham|I wish things wer...|
+-----+-----+
```

5. How many text messages were Spam and also there was using the word "lick" in it? There were 4 Spam messages in total that has been using the word "lick" in it.

```
1 spark.sql("SELECT count(class) FROM t \
2         WHERE t.class=='spam' AND t.text LIKE '%lick%' ").show()
```

```
+-----+
|count(class)|
+-----+
|              4|
+-----+
```

```
1 from pyspark.sql.functions import length
```

- ▼ Applying the Length Function on Each Text

```
1 df=df.withColumn('length',length(df['text']))
```

```
1 df.show()
```

```
+-----+-----+-----+
|class|          text|length|
+-----+-----+-----+
| ham|Go until jurong p...| 111|
| ham|Ok lar... Joking ...| 29|
| spam|Free entry in 2 a...| 155|
| ham|U dun say so earl...| 49|
| ham|Nah I don't think...| 61|
| spam|FreeMsg Hey there...| 147|
| ham|Even my brother i...| 77|
| ham|As per your reque...| 160|
| spam|WINNER!! As a val...| 157|
| spam|Had your mobile 1...| 154|
| ham|I'm gonna be home...| 109|
| spam|SIX chances to wi...| 136|
| spam|URGENT! You have ...| 155|
| ham|I've been searchi...| 196|
| ham|I HAVE A DATE ON ...| 35|
| spam|XXXMobileMovieClu...| 149|
| ham|Oh k...i'm watchi...| 26|
| ham|Eh u remember how...| 81|
| ham|Fine if that's th...| 56|
| spam|England v Macedon...| 155|
+-----+-----+-----+
only showing top 20 rows
```

Clearly the Ham Messages Have a Lower Length than the Spam Messages

```
1 df.groupby('class').mean().show()
```

```
+-----+-----+
|class|      avg(length)|
+-----+-----+
|  ham|71.45431945307645|
| spam|138.6706827309237|
+-----+-----+
```

Importing the Needed Libraries

```
1 # Needed libraries
2 import matplotlib.pyplot as plt
3 import pandas as pd
4 import numpy as np
5 import seaborn as sns
6 %matplotlib inline
7
```

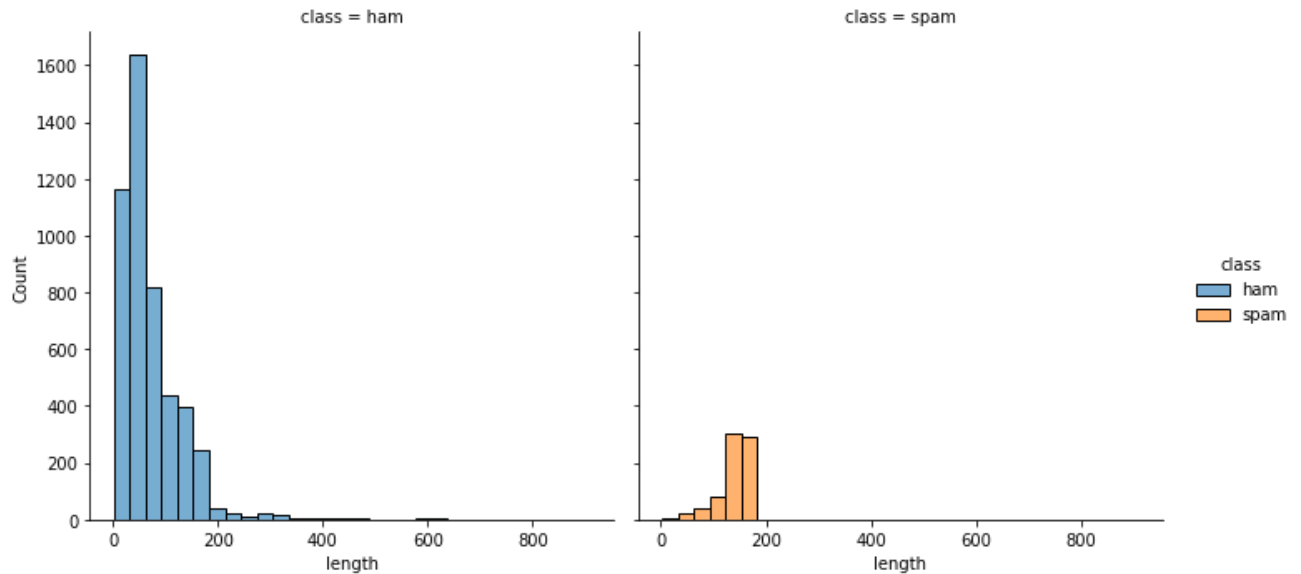
```
1 df_pandas=df.toPandas()
```

Clearly it is showing in both histograms that the Spams mostly are more than 150 words in case the most Ham texts are less than 150 words.

Histograms

```
1 plt.figure(figsize=(20,8))
2 sns.displot(df_pandas, x='length', hue="class",bins=30,alpha=.6,multipl
3 plt.show()
4
```

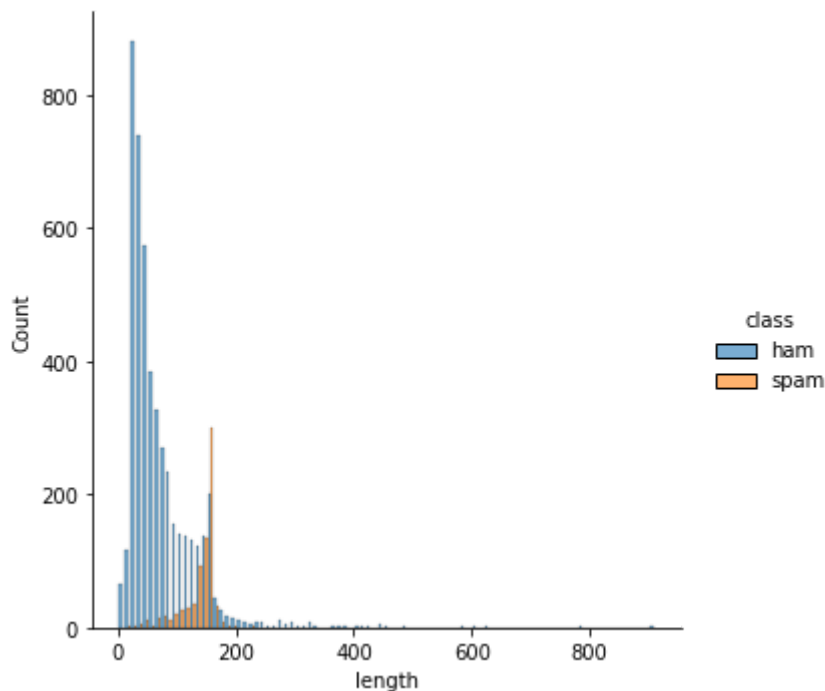
<Figure size 1440x576 with 0 Axes>



▼ Ham Vs Spam

```
1 plt.figure(figsize=(20,8))
2 sns.displot(df_pandas, x='length', hue="class",bins=50,alpha=.6,multipl
3 plt.show()
```

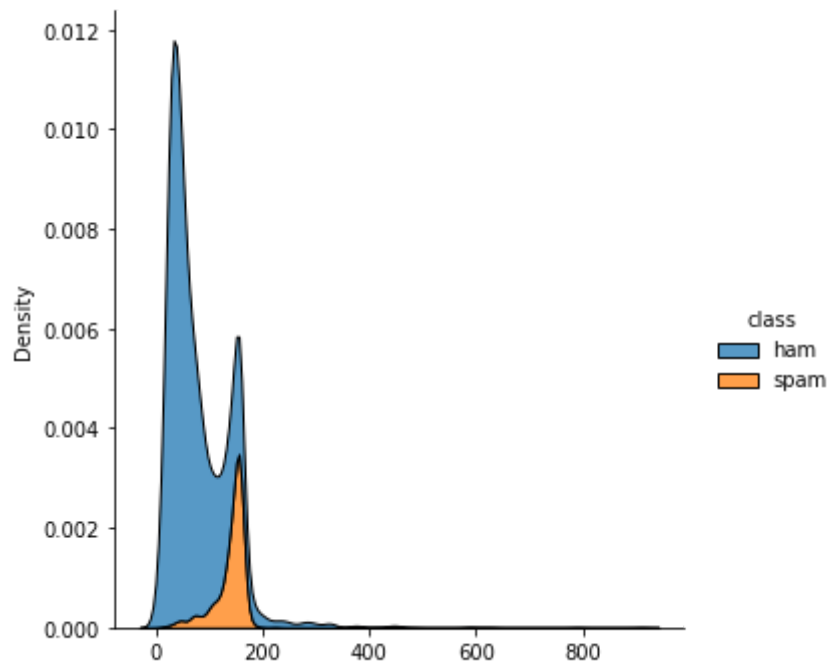
<Figure size 1440x576 with 0 Axes>



▼ Distribution of the Features

```
1 sns.displot(df_pandas, x='length', kind="kde",hue='class',multiple="st
```


<seaborn.axisgrid.FacetGrid at 0x7f3b926fa898>



▼ Feature Transfromation

▼ Tokenizer,StopWordsRemover,CountVectorizer,IDF,StringIndexer

```
1 from pyspark.ml.feature import Tokenizer,StopWordsRemover,CountVectori
```

```
1 tokenizer = Tokenizer(inputCol="text", outputCol="token_text")
2 stopremove = StopWordsRemover(inputCol='token_text',outputCol='stop_tc
3 count_vec = CountVectorizer(inputCol='stop_tokens',outputCol='c_vec')
4 idf = IDF(inputCol="c_vec", outputCol="tf_idf")
5 ham_spam_to_num = StringIndexer(inputCol='class',outputCol='label')
```

```
1 from pyspark.ml.feature import VectorAssembler
2 from pyspark.ml.linalg import Vector
```

▼ Clean Up the Data

```
1 cleanedData = VectorAssembler(inputCols=['tf_idf','length'],outputCol=
```

Model

▼ Naive Bayes

```
1 from pyspark.ml.classification import NaiveBayes

1 # Use defaults
2 nb = NaiveBayes()
```

▼ Pipeline

```
1 from pyspark.ml import Pipeline

1 data_prep_pipe = Pipeline(stages=[ham_spam_to_num,tokenizer,stopremove

1 cleaner = data_prep_pipe.fit(df)

1 clean_data = cleaner.transform(df)
```

▼ Train and Evaluation

```
1 clean_data.show()
```

class	text	length	label	token_text	stop_tokens
ham	Go until jurong p...	111	0.0	[go, until, juron...	[go, jurong, poin... (1
ham	Ok lar... Joking ...	29	0.0	[ok, lar..., joki...	[ok, lar..., joki... (1
spam	Free entry in 2 a...	155	1.0	[free, entry, in,...]	[free, entry, 2, ... (1
ham	U dun say so earl...	49	0.0	[u, dun, say, so,...]	[u, dun, say, ear... (1
ham	Nah I don't think...	61	0.0	[nah, i, don't, t...	[nah, think, goes... (1
spam	FreeMsg Hey there...	147	1.0	[freemsg, hey, th...	[freemsg, hey, da... (1
ham	Even my brother i...	77	0.0	[even, my, brothe...	[even, brother, l... (1
ham	As per your reque...	160	0.0	[as, per, your, r...	[per, request, 'm... (1
spam	WINNER!! As a val...	157	1.0	[winner!!, as, a,...]	[winner!!, valued... (1
spam	Had your mobile 1...	154	1.0	[had, your, mobil...	[mobile, 11, mont... (1
ham	I'm gonna be home...	109	0.0	[i'm, gonna, be, ...]	[gonna, home, soo... (1
spam	SIX chances to wi...	136	1.0	[six, chances, to...	[six, chances, wi... (1
spam	URGENT! You have ...	155	1.0	[urgent!, you, ha...	[urgent!, won, 1,... (1
ham	I've been searchi...	196	0.0	[i've, been, sear...	[searching, right... (1
ham	I HAVE A DATE ON ...	35	0.0	[i, have, a, date...	[date, sunday, wi... (1
spam	XXXMobileMovieClu...	149	1.0	[xxxmobilemoviecl...	[xxxmobilemoviecl... (1
ham	Oh k...i'm watchi...	26	0.0	[oh, k...i'm, wat...	[oh, k...i'm, wat... (1
ham	Eh u remember how...	81	0.0	[eh, u, remember,...]	[eh, u, remember,... (1
ham	Fine if that's th...	56	0.0	[fine, if, that's...	[fine, that's, wa... (1
spam	England v Macedon...	155	1.0	[england, v, mace...	[england, v, mace... (1

only showing top 20 rows

- Vectorized Data, Ready to fit the Model

label	features
0.0	(13424, [7, 11, 31, 61, 72, 344, 625, 731, 1409, 1598, 4485, 6440, 8092, 8838, 11344, 12979, 13423], [2.016698353160939, 3.5761915890787823, 5.2248109173106, 7.93290041571862, 15.77561915890787823, 30.6069590559317, 60.069590559317, 120.13918011255, 240.2783602251, 480.5567204502, 961.1134409004, 1922.2268818008, 3844.4537636016, 7688.9075272032, 15377.8150544064, 30755.6301088128, 61511.2602176256, 123022.5204352512, 246045.0408705024, 492090.0817410048, 984180.1634820096, 1968360.3269640192, 3936720.5339280384, 7873440.1078560768, 15746880.2157121536, 31493760.4314243072, 62987520.8628486144, 125975040.1716972288, 251950080.3433944576, 503900160.6867889152, 1007800320.1373778304, 2015600640.2747556608, 4031201280.5495113216, 8062402560.1099022432, 16124805120.2198044864, 32249610240.4396089728, 64499220480.8792179456, 128998440960.1758358912, 257996881920.3516717824, 515993763840.7033435648, 1031987527680.1406871296, 2063975055360.2813742592, 4127950110720.5627485184, 8255900221440.1125497136, 16511800442880.2250994272, 33023600885760.4501988544, 66047201771520.9003977088, 132094403543040.1800775176, 264188807086080.3601550352, 528377614172160.7203100704, 1056755228344320.1440620384, 2113510456688640.2881240768, 4227020913377280.5762481536, 8454041826754560.1152496272, 16908083653509120.2304992544, 33816167307018240.4609985088, 67632334614036480.9219970176, 135264669228072960.1843974352, 270529338456145920.3687948704, 541058676912291840.7375897408, 1082117353824583680.1475179216, 2164234707649167360.2950358432, 4328469415298334720.5900716864, 8656938830596669440.1180143392, 17313877661193338880.2360286784, 34627755322386677760.4720573568, 69255510644773355520.9441147136, 138511021289546711040.1888294272, 277022042579093422080.3776588544, 554044085158186844160.7553177088, 1108088170316333688320.1510634176, 2216176340632667376640.3021268352, 4432352681265334753280.6042536704, 8864705362530669506560.1208507381, 17729410725061339013120.2417014762, 35458821450122678026240.4834029524, 70917642900245356052480.9668059048, 141835285800490712104960.1932118096, 283670571600981424209920.3864236192, 567341143201962848419840.7728472384, 1134682286403925696839680.1545674568, 2269364572807851393679360.3091349136, 4538729145615702787358720.6182698272, 9077458291231405574717440.1236519544, 18154916582462811149434880.2473039088, 36309833164925622298869760.4946078176, 72619666329851244597739520.9892156352, 145239332659702489195479040.1978312704, 290478665319404978390958080.3956625408, 580957330638809956781916160.7913250816, 1161914661277619913563832320.1582601632, 2323829322555239827127664640.3165203264, 4647658645110479654255329280.6330406528, 9295317290220959308510658560.1266801306, 18590634580441918617021317120.2533602612, 37181269160883837234042634240.5067205224, 74362538321767674468085268480.1013440448, 148725076643535348936170536960.2026880896, 297450153287070697872341073920.4053761792, 594900306574141395744682147840.8107523584, 1189800613148282791489364295680.1621504736, 2379601226296565582978728591360.3243009472, 4759202452593131165957457182720.6486018944, 9518404905186262331914914365440.1297203788, 19036809810372524663829828730880.2594407576, 38073619620745049327659657461760.5188815152, 76147239241490098655319314923520.1037730304, 152294478482980197310638629847040.2075460608, 304588956965960394621277259694080.4150921216, 609177913931920789242554519388160.8301842432, 1218355827863841578485109038776320.1660364464, 2436711655727683156970218077552640.3320728928, 4873423311455366313940436155105280.6641457856, 9746846622910732627880872310210560.1328291512, 19493693245821465255761744620421120.2656583024, 38987386491642930511523489240842240.5313166048, 77974772983285861023046978481684480.1062632096, 155949545966571722046093956963368960.2125264192, 311899091933143444092184193926737920.4250528384, 623798183866286888184368387853475840.8501056768, 1247596367732573776368736775706951680.1700213536, 2495192735465147552737473551413903360.3400427072, 4990385470930295105474947102827806720.6800854144, 998077094186059

11/13

▼ Split and Fitting the Data

```
1 (training,testing) = clean_data.randomSplit([0.7,0.3])
```

```
1 spam_predictor = nb.fit(training)
```

```
1 df.printSchema()
```

```
↳ root
  |-- class: string (nullable = true)
  |-- text: string (nullable = true)
  |-- length: integer (nullable = true)
```

```
1 test_results = spam_predictor.transform(testing)
```

```
1 test_results.show()
```

```
+-----+-----+-----+-----+
|label|          features|      rawPrediction|      probability|prediction|
+-----+-----+-----+-----+
| 0.0|(13424,[0,1,14,31...|[-218.42139575785...|[1.0,4.1253117916...|      0.0|
| 0.0|(13424,[0,1,20,27...|[-972.72274973886...|[1.0,4.8058041016...|      0.0|
| 0.0|(13424,[0,1,21,27...|[-750.57452194974...|[1.0,7.5755951094...|      0.0|
| 0.0|(13424,[0,1,23,63...|[-1309.8364942103...|[1.0,1.5935943591...|      0.0|
| 0.0|(13424,[0,1,24,31...|[-356.23115975709...|[1.0,1.5083158009...|      0.0|
| 0.0|(13424,[0,1,27,88...|[-1524.4837075990...|[1.03095559490770...|      1.0|
| 0.0|(13424,[0,1,30,12...|[-600.50816776760...|[1.0,3.8787327375...|      0.0|
| 0.0|(13424,[0,1,31,43...|[-337.80543485110...|[1.0,3.1063184408...|      0.0|
| 0.0|(13424,[0,1,43,69...|[-630.13790402312...|[0.00181752980442...|      1.0|
| 0.0|(13424,[0,1,46,17...|[-1138.9889265504...|[8.47185188356784...|      1.0|
| 0.0|(13424,[0,1,72,10...|[-680.71295470136...|[1.0,1.4964461481...|      0.0|
| 0.0|(13424,[0,1,874,1...|[-97.684060707401...|[0.99999998316042...|      0.0|
| 0.0|(13424,[0,2,3,5,6...|[-2556.9300904702...|[1.0,3.8247939396...|      0.0|
| 0.0|(13424,[0,2,3,6,9...|[-3284.9514767477...|[1.0,1.78751305E-...|      0.0|
| 0.0|(13424,[0,2,4,5,7...|[-996.79951588224...|[1.0,9.2526890292...|      0.0|
| 0.0|(13424,[0,2,4,5,1...|[-2484.8726201572...|[1.0,1.8603379715...|      0.0|
| 0.0|(13424,[0,2,4,8,1...|[-1312.0648001828...|[1.0,8.8787844286...|      0.0|
| 0.0|(13424,[0,2,4,11,...|[-1126.9645557754...|[1.0,1.4277000914...|      0.0|
| 0.0|(13424,[0,2,7,11,...|[-738.60913759771...|[1.0,6.1012287779...|      0.0|
| 0.0|(13424,[0,2,7,11,...|[-1315.8949170122...|[1.0,5.0725960155...|      0.0|
+-----+-----+-----+-----+
```

only showing top 20 rows

```
1 from pyspark.ml.evaluation import MulticlassClassificationEvaluator
```

▼ Evaluation the Test Data

```
1 acc_eval = MulticlassClassificationEvaluator()  
2 acc = acc_eval.evaluate(test_results)  
3 print("Accuracy of model at predicting spam was: {:.2f}".format(acc))
```

Accuracy of model at predicting spam was: 0.92

▼ Found an accuracy of 92 % of our analysys. The model is ready for practicing on further sample messages in future.

1