## Name: Jui Shaligram

## Implement Machine Learning Concepts for Finding Spam and Non Spam Emails

**# Import different libraries for the purpose of creating a logistic regression model and test query data**

```
>>> from pyspark import SparkContext

>>> from pyspark.mllib.regression import LabeledPoint

>>> from pyspark.mllib.classification import LogisticRegressionWithSGD

>>> from pyspark.mllib.feature import HashingTF

>>> import numpy

>>> import os

>>> import pickle
```

**# Create a spark context**

```
>>> sc = spark.sparkContext
```

**# Read spam data from the given input file. Input file is stored in shared folder common to Windows and Ubuntu**

```
>>> spamData = sc.textFile("/media/sf_Downloads/Data For
Spark/Assignment4/emails_spam.txt")

18/03/08 14:32:37 WARN SizeEstimator: Failed to check whether UseCompressedOops is
set; assuming yes
```

**# Read non-spam data from given input file.**

```
>>> nonspamData = sc.textFile("/media/sf_Downloads/Data For
Spark/Assignment4/emails_nospam.txt")
```

**# Read query data from the given input file**

```
>>> queryData = sc.textFile("/media/sf_Downloads/Data For Spark/Assignment4/query.txt")
```

**# A HashingTF instance tfhash is created to map the text given into 100 different features. Based on these features we will classify data as spam and nonspam.**

```
>>> tfhash = HashingTF(numFeatures = 100)
```

**# Spam data is taken, split by space and then each word is mapped into a feature.**

```
>>> spamDataFeatures = spamData.map(lambda email: tfhash.transform(email.split(" ")))
```

**# Nonspam data is taken, split by space and then each word is mapped into a feature.**

```
>>> nonspamDataFeatures = nonspamData.map(lambda email:
tfhash.transform(email.split(" ")))
```

**# LabeledPoints are created for spam and nonspam data. Spam data here is having positive samples and non-spam data is having negative samples.**

```
>>> positiveSamples = spamDataFeatures.map(lambda features: LabeledPoint(1, features))
>>> negativeSamples =
    nonspamDataFeatures.map(lambda features: LabeledPoint(0, features))
```

**# Union of positive and negative samples is taken to form training data on which our training model is developed.**

```
>>> trainingdata = positiveSamples.union(negativeSamples)
```

**# Cache the training data**

```
>>> trainingdata.cache()
UnionRDD[6] at union at NativeMethodAccessorImpl.java:0
```

**# Regression model is developed with training data.**

```
>>> trainingmodel = LogisticRegressionWithSGD.train(trainingdata)
```

**# Using the model to predict query data for spam and non-spam data**

```
>>> queryPrediction =
queryData.map(lambda email: (email, trainingmodel.predict(tfhash.transform(email.split("
")))))
>>> queryPrediction.collect()
```

**Output:**

[(u"this is a year of promotion for Galaxy End of YearPromo You have 1 week remaining to retrieve your won prize for the Samsung Galaxy Xmas Promo 'C' draw category winning prize of Seven Hundred and Fifty Thousand Euros each and a Samsung Galaxy S6 EDGE. Winning Ticket Number:WIN-707-COS.  We advise you to keep this winning notification confidential and away from public notice to avoid double claim/mistransfer or impersonation until after remittance/payment to you.", 1),

(u"you are the lucky one: We've picked out 10 new matches for you. Meet them now and then check out all the singles in your area! you might win a prize too", 1),

(u'Do not miss your chances: Get Viagra real cheap!  Send money right away to ...', 1),

(u'Get real money fast: With my position in the office i assure you with 100% risk free that this transaction is not a childish game play and i want you to indicate your full interest with assurance of trust that you will not betray me once the fund is transfer into your nominated bank account, while i look forward for your urgent reply.', 1),

(u'Dear Spark Learner, Thanks so much for attending the Spark Summit 2014!  Check out videos of talks from the summit at ...', 0),

(u'Hi Mom, Apologies for being late about emailing and forgetting to send you the package. I hope you and bro have been ...', 0),

(u'Wow, hey Fred, just heard about the Spark petabyte sort.  I think we need to take time to try it out immediately ...', 0),

(u'Hi Spark user list, This is my first question to this list, so thanks in advance for your help! I tried running ...', 0)]