University of Sheffield

COM6012 Assignment

Jagpreet Jakhar

Department of Computer Science

May 5, 2023

Contents

List of Figures

List of Tables

Chapter 1

Question 1. Log Mining and Analysis

1.1 Find out the total number of requests for :

1.1.1 All hosts from Germany ending with ".de" : All hosts from Canada ending with ".ca" : All hosts from Singapore ending with ".sg",Report these three numbers and visualise them using a graph of your choice :

Table 1.1: *Hosts by country*

Hosts ending with .de	Hosts ending with .ca	Hosts ending with .sg
21345	58290	1057

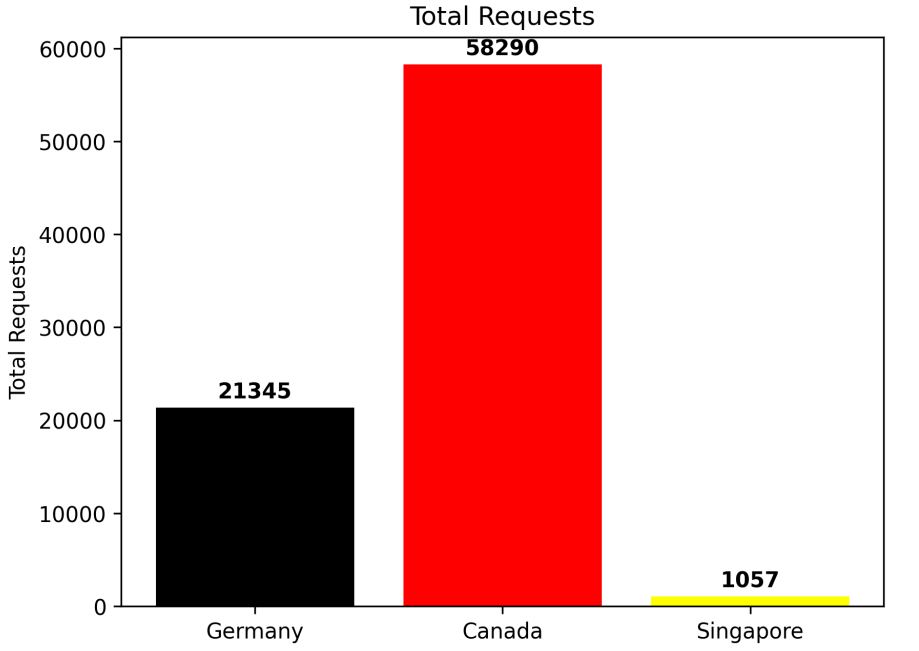

Figure 1.1: *Number of Requests*

1.2 For each of the three countries in Question A (Germany, Canada, and Singapore), find the number of unique hosts and the top 9 most frequent hosts among them. You need to report three numbers and 3 x 9 = 27 hosts in total.

Germany	count
host62.ascend.interop.eunet.de	832
aibn32.astro.uni-bonn.de	642
ns.scn.de	523
www.rrz.uni-koeln.de	423
ztivax.zfe.siemens.de	387
sun7.lrz-muenchen.de	280
relay.ccs.muc.debis.de	275
dws.urz.uni-magdeburg.de	244
relay.urz.uni-heidelberg.de	239

Canada	count
ottgate2.bnr.ca	1718
freenet.edmonton.ab.ca	782
bianca.osc.on.ca	511
alize.ere.umontreal.ca	479
pcrb.ccrs.emr.ca	461
srv1.freenet.calgary.ab.ca	362
ccn.cs.dal.ca	351
oncomdis.on.ca	304
cobain.arcs.bcit.bc.ca	289

Singapore	count
merlion.singnet.com.sg	308
sunsite.nus.sg	40
ts900-1314.singnet.com.sg	30
ssc25.iscs.nus.sg	30
scctn02.sp.ac.sg	25
ts900-1305.singnet.com.sg	25
ts900-406.singnet.com.sg	25
ts900-402.singnet.com.sg	24
einstein.technet.sg	23

```
Number of unique hosts from Germany: 1138
Number of unique hosts from Canada: 2970
Number of unique hosts from Singapore: 78
```

1.3 For each country, visualise the percentage (with respect to the total in that country) of requests by each of the top 9 most frequent hosts and the rest (i.e. 10 proportions in total) using a graph of your choice with the 9 hosts clearly labelled on the graph. Three graphs need to be produced.

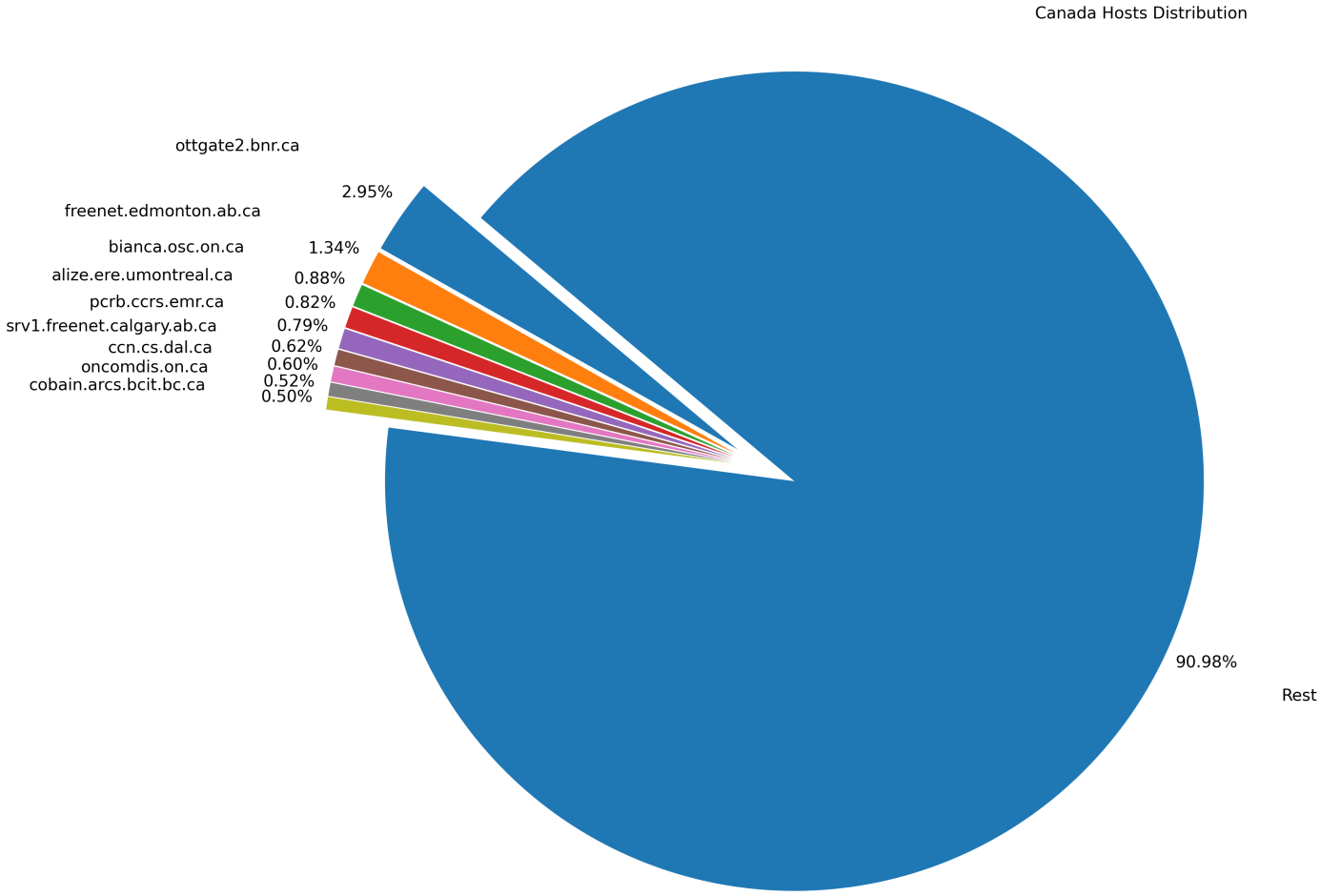

Figure 1.2: *Percentage of requests by each of the top 9 most frequent hosts and the rest in Canada*

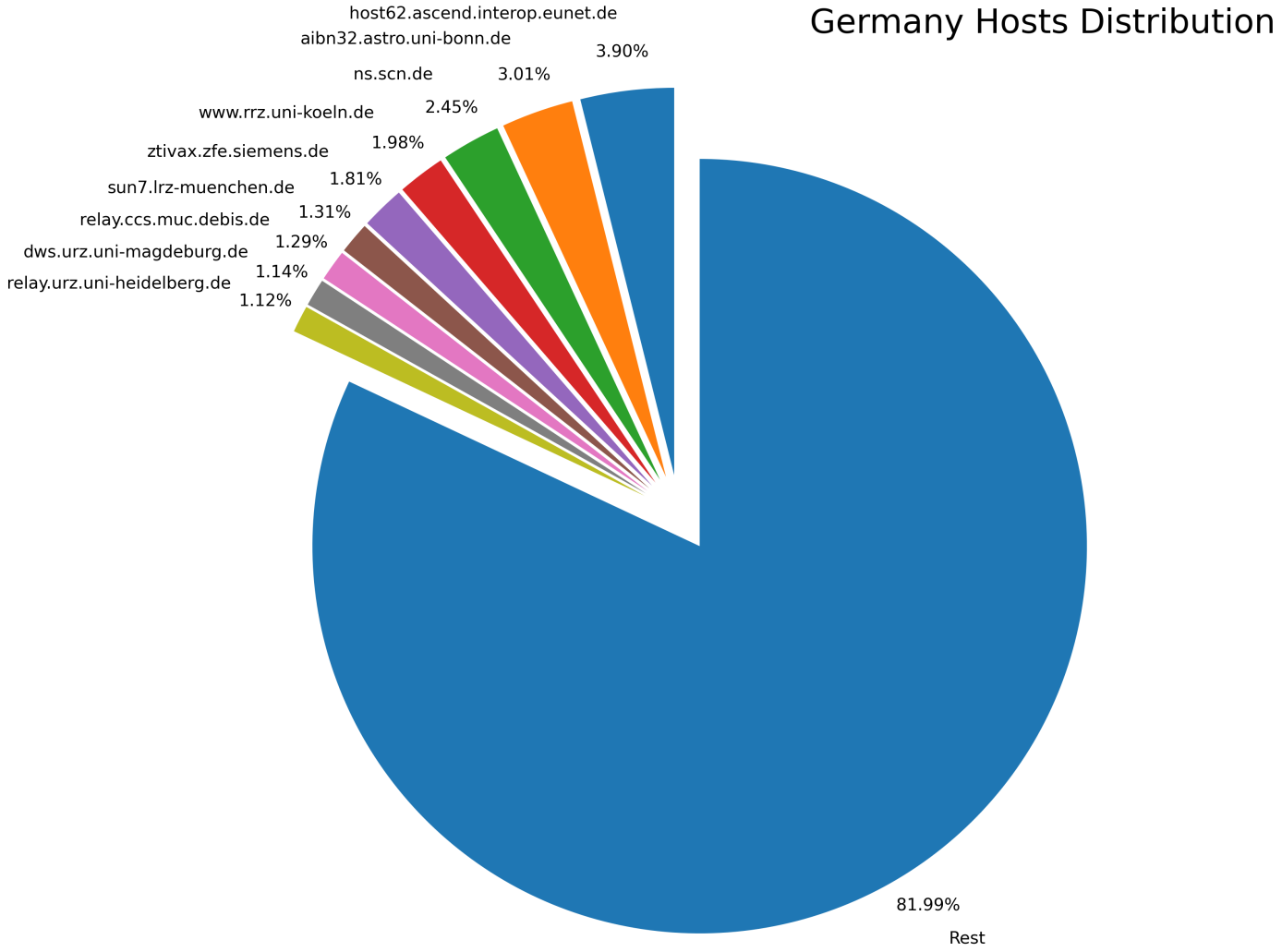

Figure 1.3: *Percentage of requests by each of the top 9 most frequent hosts and the rest in Germany*

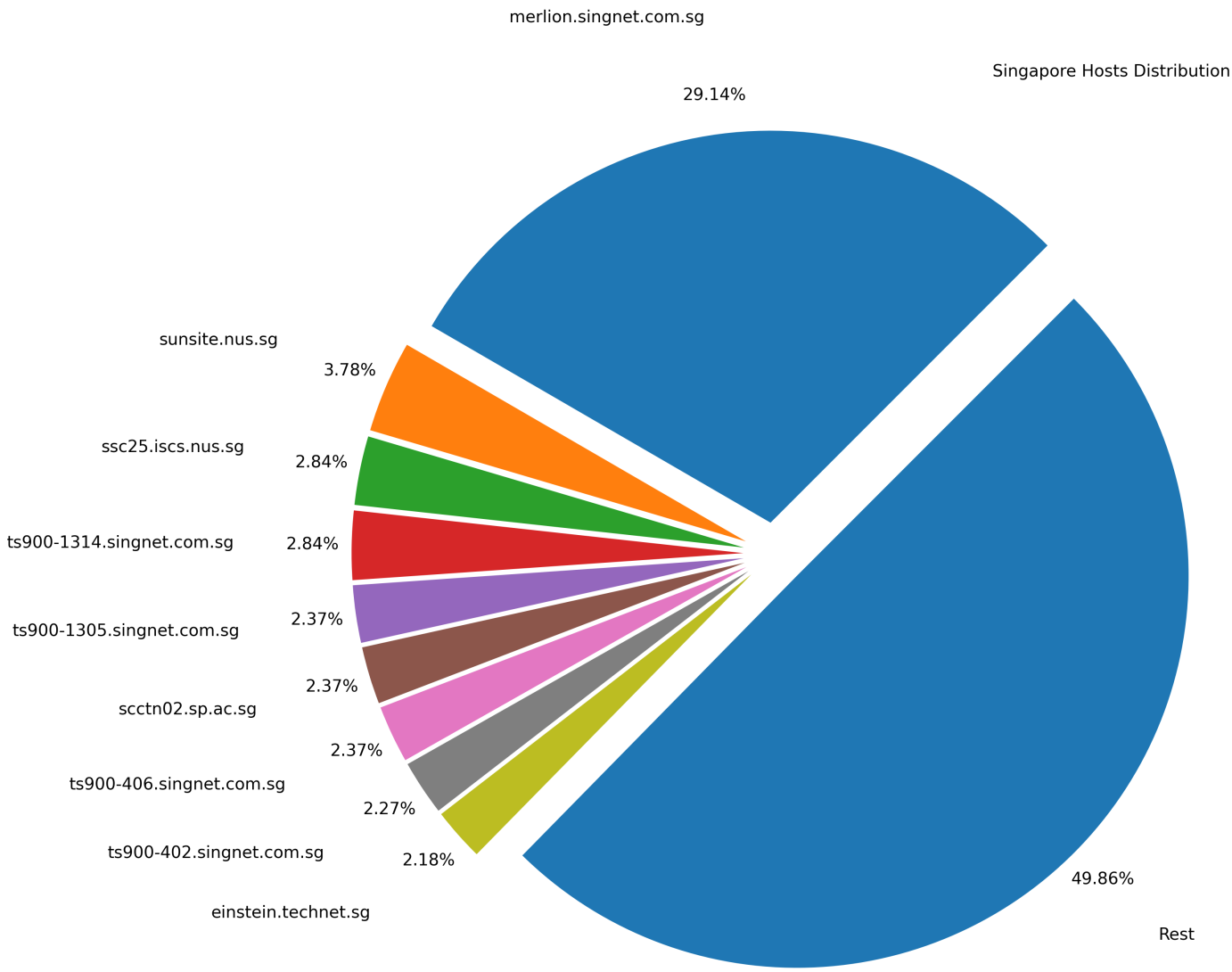

Figure 1.4: *Percentage of requests by each of the top 9 most frequent hosts and the rest in Singapore*

1.4 For the most frequent host from each of the three countries, produce a heatmap plot with day as the x-axis (the range of x-axis should cover the range of days available in the log file. If there are 31 days, it runs from 1st to 31st. If it starts from 5th and ends on 25th, it runs from 5th to 25th), the hour of visit as the y-axis (0 to 23, as recorded on the server), and the number of visits indicated by the colour. Three x-y heatmap plots need to be produced with the day and hour clearly labelled.

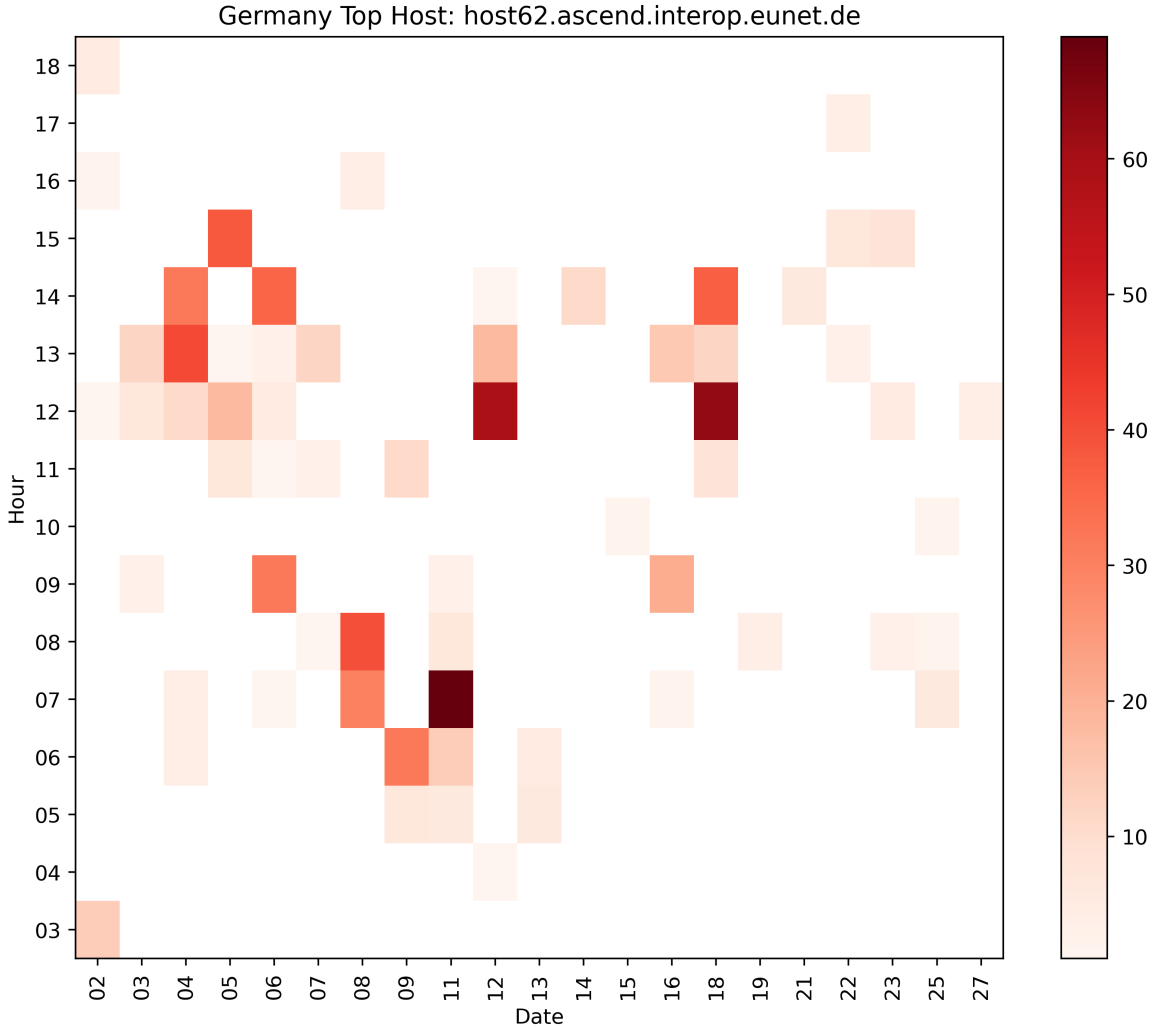

Figure 1.5: *Heatmap Germany*

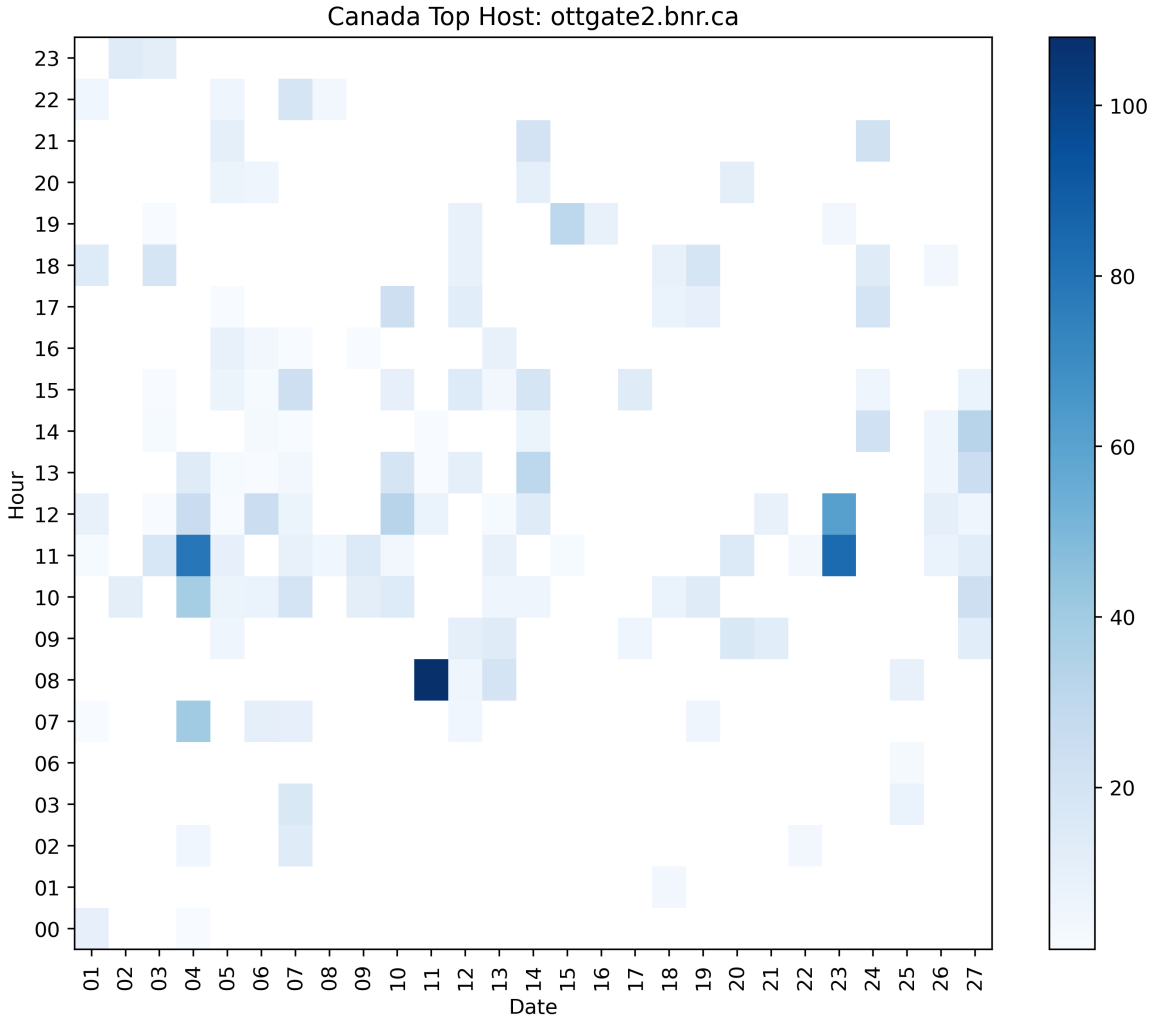

Figure 1.6: *Heatmap Canada*

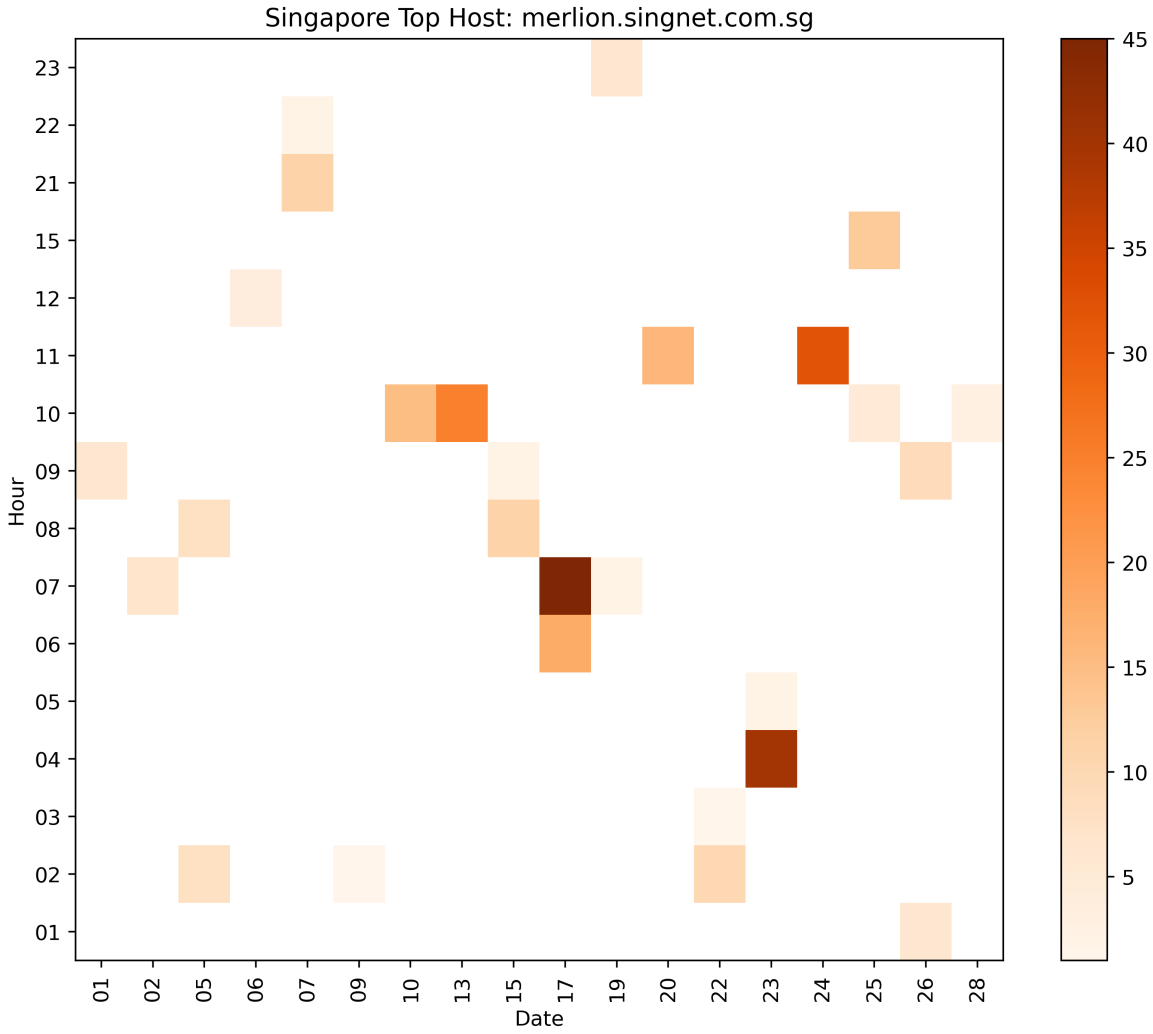

Figure 1.7: *Heatmap Singapore*

1.5 Discuss two most interesting observations from A to D above, each with three sentences:

1) What is the observation?

2) What are the possible causes of the observation?

3) How useful is this observation to NASA?

- Canada Accounts for more than twice the requests from Germany and Singapore combined.Possible cause may be Proximity and awareness and interest towards NASA.Canadians have a rich history of participation in NASA and NASA may invest in the Human resources in Canada to get more participation.

- The distribution of requests by hosts is more equitable for Canada, whereas for Singapore some unique hosts dominate the requests, It may be due to the small size of Singapore.NASA may benefit from focusing on more awareness about its activites in regions with small populace and far away from USA.

Chapter 2

Question 2 Liability Claim Prediction

2.1 Provide RMSE or accuracy, and model coefficients for each of the predictive models obtained from the following tasks

1.Model the number of claims (ClaimNb) conditionally on the input features via Poisson regression.

```
For Poisson Regression
RMSE = 0.247971
Model Coefficients:
[0.005057972508974127,-0.0010617099596318925,-0.0006420225090499234,
-0.0026403332535199784,0.005425916695717338,-0.001313778783670871,
-0.0029290452708882835,0.0001915070605640519,0.004618634937963245,
-0.0009230232983699072,0.012783975626582742,0.012563324980146204,
-0.0016204448366736884,0.000436841293493627,0.016037417431419537,
0.0016248072291138468,-0.0045462474355981684,-0.008417998933085882,
-0.008790698021585986,9.496731660541077e05,-0.008197231152832416,
-0.011081436568077595,0.004965862070833389,0.0001108541757780317 8,
-0.008991423598088069,0.006285034777604171,-0.008742035138219556,
0.0053044556858856335,0.003153203994716013,0.0004638436311752013,
0.000545422107288944,-0.00253979465233204,0.006198501107485032,
0.00915966427710716,-0.010495646492162418,-0.004752921567334665,
0.008419852891907506,-0.10159198170614923,0.008101109235657095,
-4.0094361262874094e05,0.009728952192964114,-0.007438649892752686,
-0.01362552713963212,0.0067516295936443,0.012350295405749667,
0.0019428325977281563,-0.004456554586195711,-0.004381561898551594,
-0.007128238628384552,0.03289832064251561,-0.015320659962800316,
0.0025618030172044415,0.0024680192192399796]
```

2.Model the relationship between LogClaimNb and the input features via Linear regression, with L1 and L2 regularisation respectively.

```
    RMSE Linear Regression_L1 = 0.255

Coefficients:  [0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,
0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,
0.0,0.0,0.0,0.0,0.0,-0.016581673766405677,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,
0.0,0.0,0.0,0.008097874053047245,0.0,0.0,0.0]

    RMSE Linear Regression_L2 = 0.254
Coefficients:  [0.0034733119104192675,-0.0005571401414144276,9.596406020987964e06,
-0.0011453077352969316,0.004110841256549953,-0.00030336597713209355,
-0.0013221623460423536,0.0009239355945296598,0.004747531529637773,
-0.0004208944748230224,0.006773420329904437,0.006413630721841841,
-0.0028330404087949894,-0.0014412311198636514,0.00983210319899552,
0.0001315863200300 4947,-0.006203676942293502,-0.007257293845590484,
-0.008144170252672719,-0.0010084030397925286,-0.007982861182917411,
-0.010114330950503169,0.0036356010048807586,-0.0010145786298292228,
-0.010425552703212122,0.005226908607911314,-0.0122075577134902,
0.00630849879525538,0.0012400383877276704,-0.0008250009094772366,
-0.0009497519591859382,-0.0022405371452436616,0.0025115319920377033,
0.004696154666792484,-0.007101994528436635,-0.003666895012212548,
0.004794042021631531,-0.07246495078392932,0.004304762750661181,
-0.0006220118130317035,0.005333621862785282,-0.005629329950126414,
-0.009406015920729918,0.003998989613299877,0.007706166687900909,
0.0008898011526206624,-0.005127437527660577,-0.008306588548390198,
-0.013503811362767692,0.018749847606745494,-0.008787600812341086,
0.0020723090983900626,0.0011623353875302041]
```

3.Model the relationship between NZClaim and the input features via Logistic regression, with L1 and L2 regularisation respectively

```
Logistic Regression With L1 Regularisation Accuracy : 0.947

Coefficients:  [-0.14120436  0.1239572 ]

    Logistic Regression With L2 Regularisation Accuracy : 0.947

Coefficients:  [ 4.95267003e-02 -2.25228847e-02 -8.96585405e-03
-2.29570667e-02
5.58986978e-02 -1.50781592e-02 -2.85202832e-02 -7.25037306e-03
  5.60568034e-02 -2.28319423e-02  1.09476740e-01  9.96401277e-02
 -5.63098452e-02 -1.59624798e-02  1.54702722e-01 -6.41793012e-05
 -1.46184979e-01 -1.44769522e-01 -1.59839855e-01 -2.02103170e-02
```

```
-1.64154661e-01 -1.40073111e-01  6.50261155e-02 -2.08128076e-02
-2.41216528e-01  8.03814357e-02 -2.31890297e-01  8.55956660e-02
-8.67344720e-03 -2.68118002e-02 -1.61673533e-02 -2.76434771e-02
 4.60639531e-02  7.72018581e-02 -9.98958735e-02 -4.64551458e-02
 7.56554395e-02 -9.18444886e-01  6.46487548e-02 -1.24544132e-02
 8.38019927e-02 -9.38421491e-02 -1.77690011e-01  5.93318938e-02
 1.11564796e-01  9.44844595e-03 -9.05495091e-02 -1.31380627e-01
-2.13250292e-01  2.99771096e-01 -1.37917815e-01  3.57173810e-02
 1.95302277e-02]
```

2.1.1 Determine the values of regParam (in [0.001, 0.01, 0.1, 1, 10]) for the above tasks automatically using a small subset of the training set (e.g. 10). Plot the validation curves to files for the five models (one figure per model) with respect to the values of regParam.

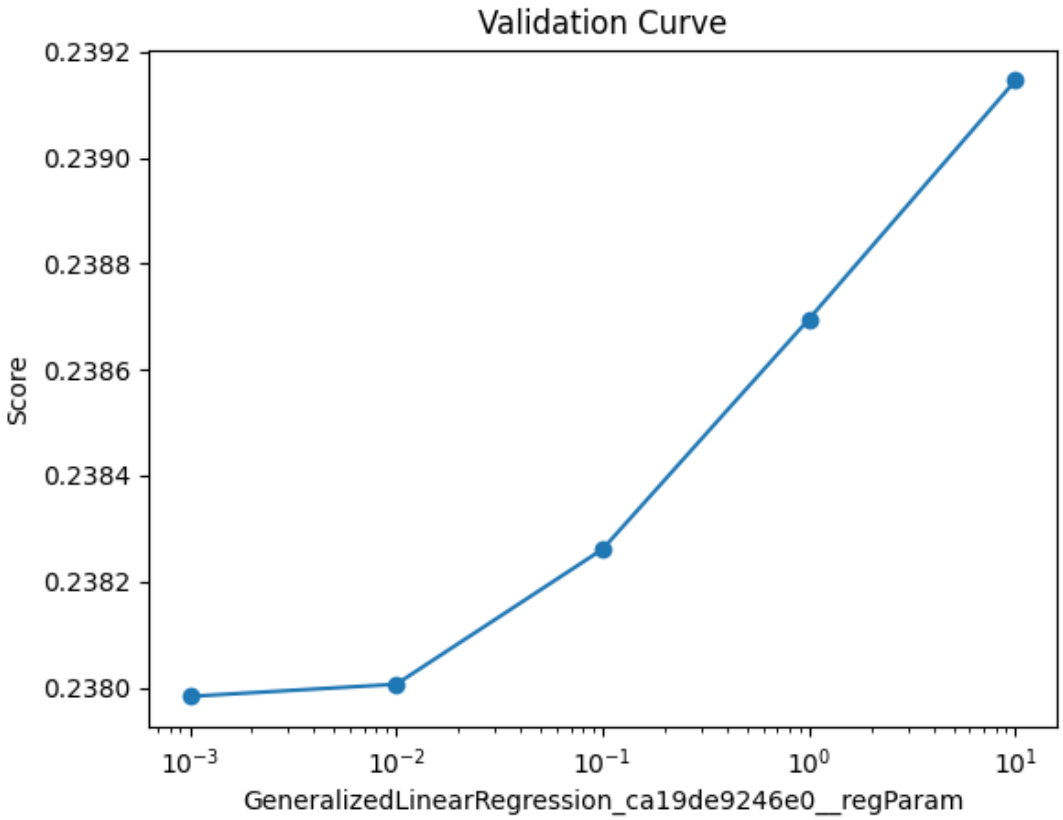

Figure 2.1: *Poisson*

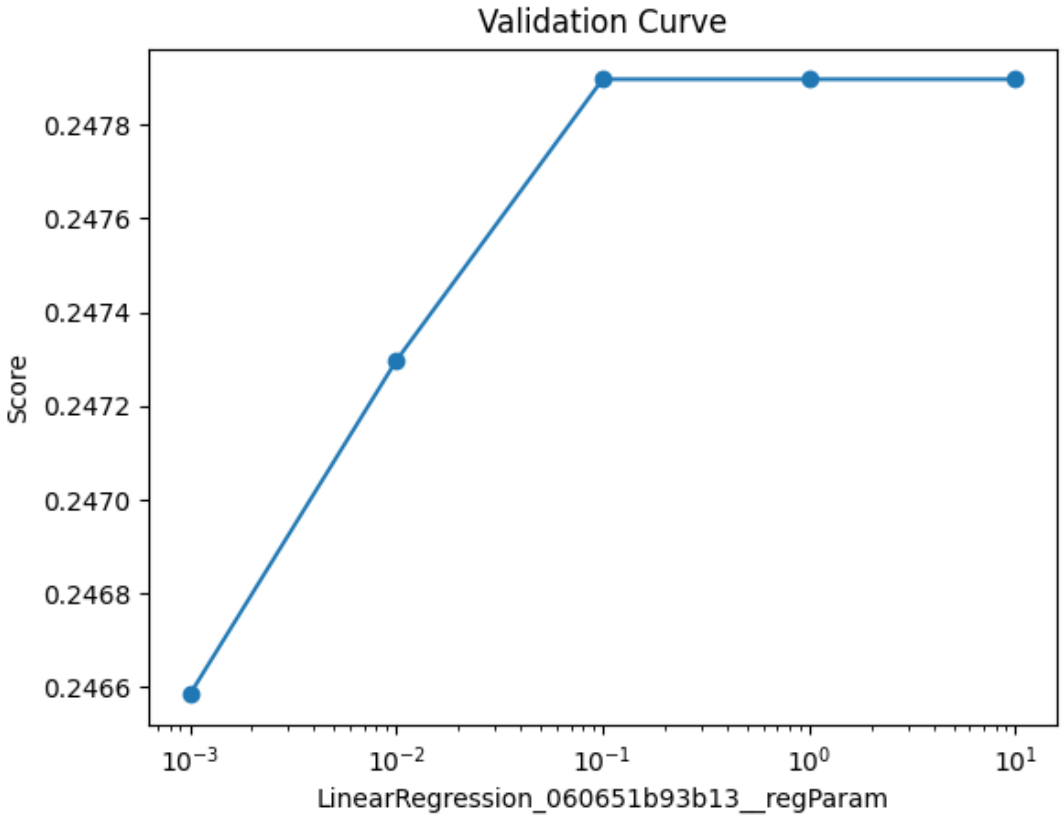

Figure 2.2: *Linear Regression L1*

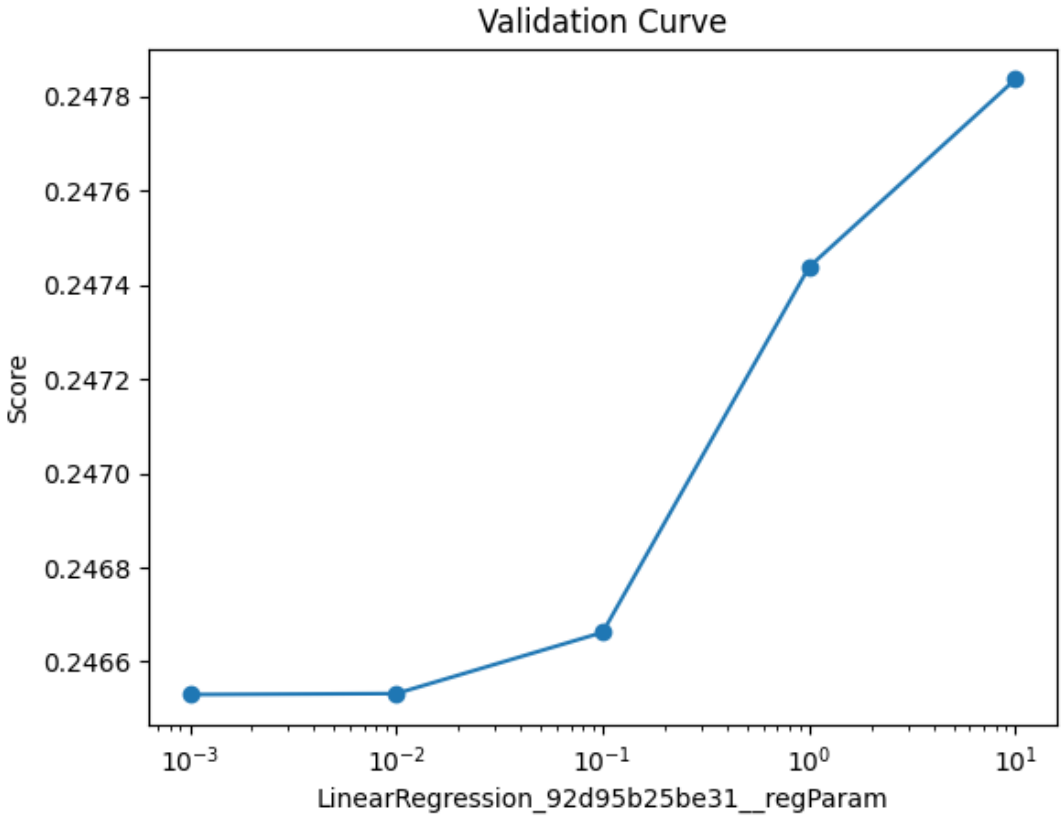

Figure 2.3: *Linear Regression L2*

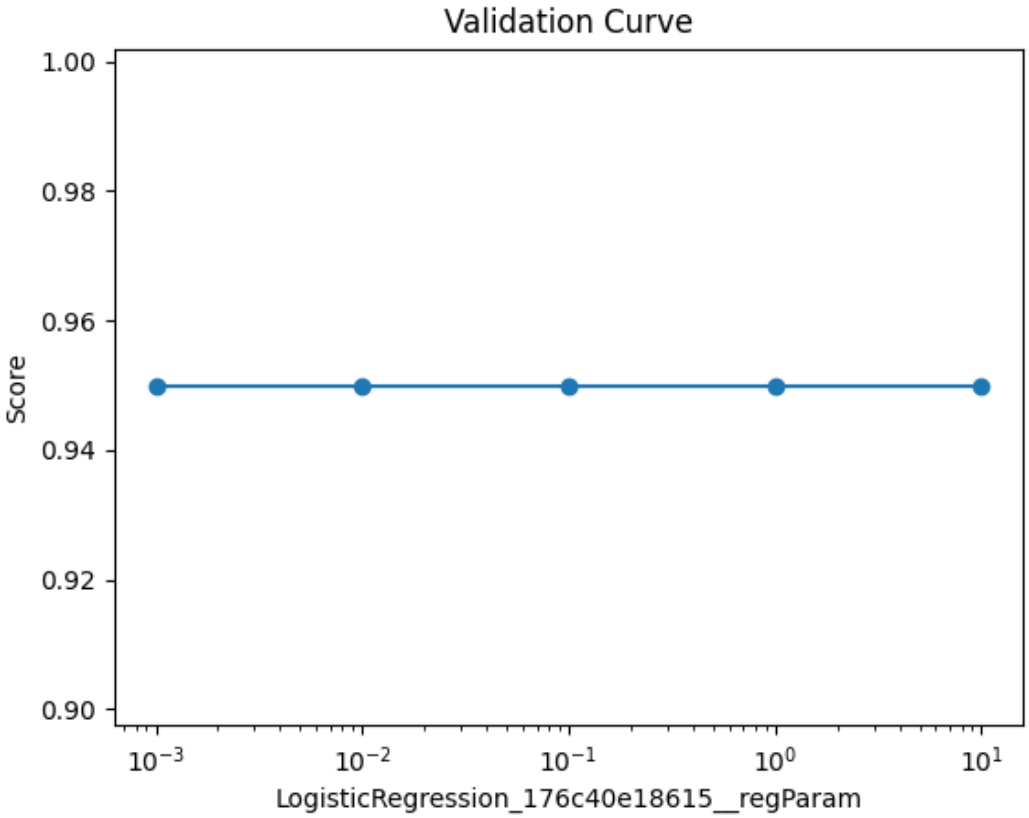

Figure 2.4: *Logistic Regression L1*

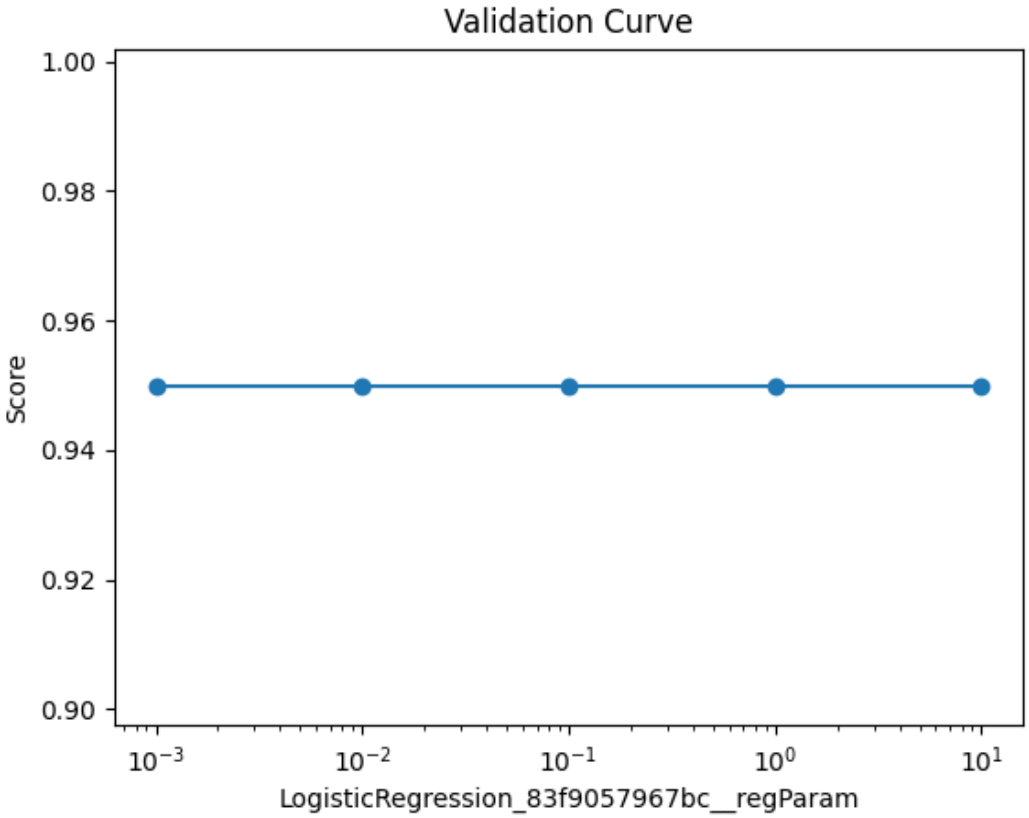

Figure 2.5: *Logistic Regression L2*

2.2 Compare the performance and coefficients obtained in Q2.B and discuss at least three observations (e.g., anything interesting), with two to three sentences for each observation. If you need to, you can run additional experiments that help you to provide these observations

- Neither RegParam's strength have any effect on the performance of Logisitic Regression Model nor the choice of L1 or L2 Regularisation.
- Linear Regression with L1 regularisation by observing its model coefficients seems to concentrate on just two input features for prediction as other model coefficients are zero. However when L2 regularisation is applied it evens out the impact more, So we can conclude in this case L2 regularisation prevents overfitting, even if not any gain in accuracy.
- The Model Coefficients of Logisitic Regression Model are more evened out and with accuracy of 94 percent, demonstrates why Logisitic regression is one of the best choices for the problems of Bianry Classification.

Chapter 3

Movie Recommendation and Cluster Analysis

3.1 Time-split Recommendation

3.1.1 Perform time-split recommendation using ALS-based matrix factorisation in PySpark on the rating data ratings.csv:

a)sort all data by the timestamp,

```
Root-mean-square error for size 40 ALS_1 = 0.8031813339236481
Root-mean-square error for size 60 ALS_1 = 0.7821724266739104
Root-mean-square error for size 80 ALS_1= 0.7963648832738655
Root-mean-square error for size 40 ALS_2 = 0.8009333019158624
Root-mean-square error for size 60 ALS_2 = 0.7796738326803554
Root-mean-square error for size 80 ALS_2= 0.7949130965849656
```

b)perform splitting according to the sorted timestamp. Earlier time (the past) should be used for training and later time (the future) should be used for testing, which is a more realistic setting than random split. Consider three such splits with three training data sizes: 40, 60, and 80.

3.1.2 For each of the three splits above, study two versions (settings) of ALS using your student number (keeping only the digits) as the seed as the following

Setting 1: The ALS setting used in Lab 5 except the seed

Setting 2: Based on results (see the next step 3 below) from the first ALS setting, choose another different ALS setting that can potentially improve the results. Provide at least a one-sentence justification to explain why you think the chosen setting can potentially improve the results.

The parameter to tune chosen was maxIter so as to allow for the model the opportunity to converge faster and be able to predict more accurately, However increasing maxIter too much can increase the risk of over-fitting, so a balance must be maintain.

3.1.3 For each split and each version of ALS, compute three metrics: the Root Mean Square Error (RMSE), Mean Square Error (MSE), and Mean Absolute Error (MAE). Put these RMSE, MSE and MAE results for each of the three splits in one Table for the two ALS settings in the report. You need to report 3 metrics x 3 splits x 2 ALS settings = 18 numbers. Visualise these 18 numbers in ONE single figure.

	Model	Metric	Split	Score
0	ALS1	rmse	Split 40:60	0.803181
1	ALS1	mse	Split 40:60	0.645100
2	ALS1	mae	Split 40:60	0.615886
3	ALS1	rmse	Split 60:40	0.782172
4	ALS1	mse	Split 60:40	0.611794
5	ALS1	mae	Split 60:40	0.594644
6	ALS1	rmse	Split 80:20	0.796365
7	ALS1	mse	Split 80:20	0.634197
8	ALS1	mae	Split 80:20	0.600149
9	ALS2	rmse	Split 40:60	0.800933
10	ALS2	mse	Split 40:60	0.641494
11	ALS2	mae	Split 40:60	0.611788
12	ALS2	rmse	Split 60:40	0.779674
13	ALS2	mse	Split 60:40	0.607891
14	ALS2	mae	Split 60:40	0.591294
15	ALS2	rmse	Split 80:20	0.794913
16	ALS2	mse	Split 80:20	0.631887
17	ALS2	mae	Split 80:20	0.597897

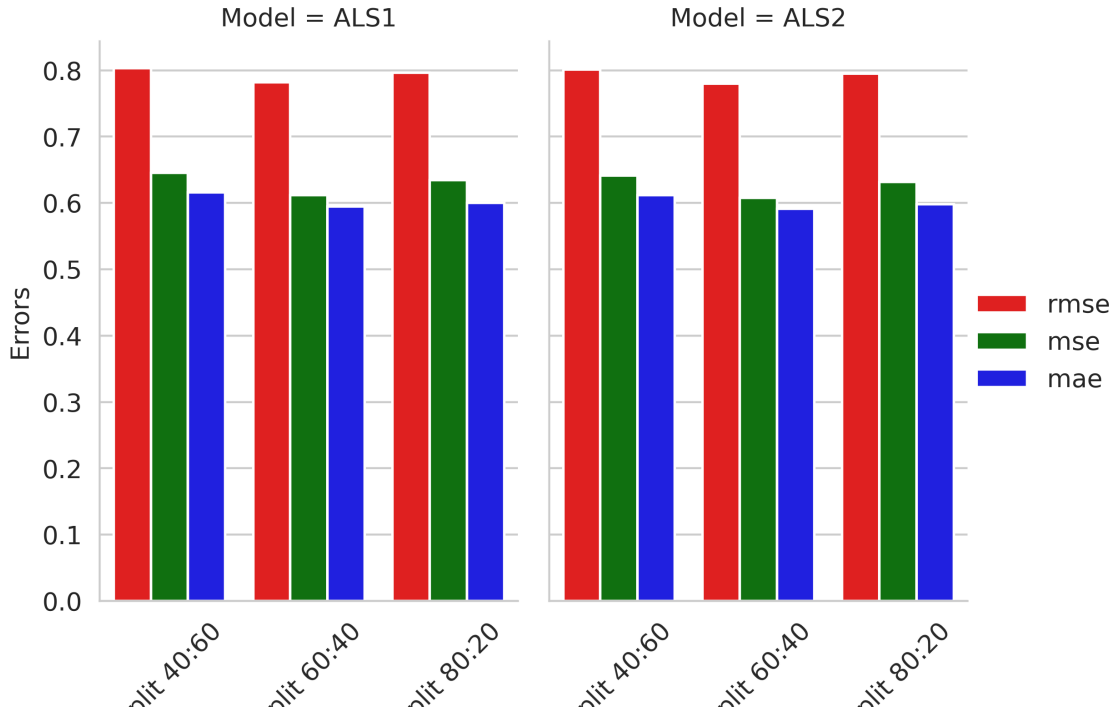

Figure 3.1: *MAE vs MSE vs RMSE*

3.2 User Analysis

3.2.1 **After ALS, each user is modelled by some factors. For each of the three time-splits, use k-means in PySpark with k=25 to cluster all the users based on the user factors learned with the ALS Setting 2 above, and find the top five largest user clusters. Report the size of (i.e. the number of users in) each of the top five clusters in one Table, in total 3 splits x 5 clusters = 15 numbers. Visualise these 15 numbers in ONE single figure.**

```
Top five largest clusters for split 1:
+----------+-----+
|prediction|count|
+----------+-----+
|         9| 4698|
|         3| 4175|
|        12| 3847|
|        21| 3750|
|         2| 3711|
+----------+-----+

Top five largest clusters for split 2:
+----------+-----+
```

```
|prediction|count|
+----------+-----+
|         7| 6841|
|         2| 6748|
|        24| 5698|
|        20| 5659|
|         5| 5351|
+----------+-----+

Top five largest clusters for split 3:
+----------+-----+
|prediction|count|
+----------+-----+
|         6| 7977|
|         7| 7312|
|        14| 7281|
|        13| 6970|
|        22| 6759|
+----------+-----+

   Split 1 Prediction  Split 1 Count  Split 2 Prediction  Split 2 Count  \
0                   9           4698                   7           6841
1                   3           4175                   2           6748
2                  12           3847                  24           5698
3                  21           3750                  20           5659
4                   2           3711                   5           5351

   Split 3 Prediction  Split 3 Count
0                   6           7977
1                   7           7312
2                  14           7281
3                  13           6970
4                  22           6759
```

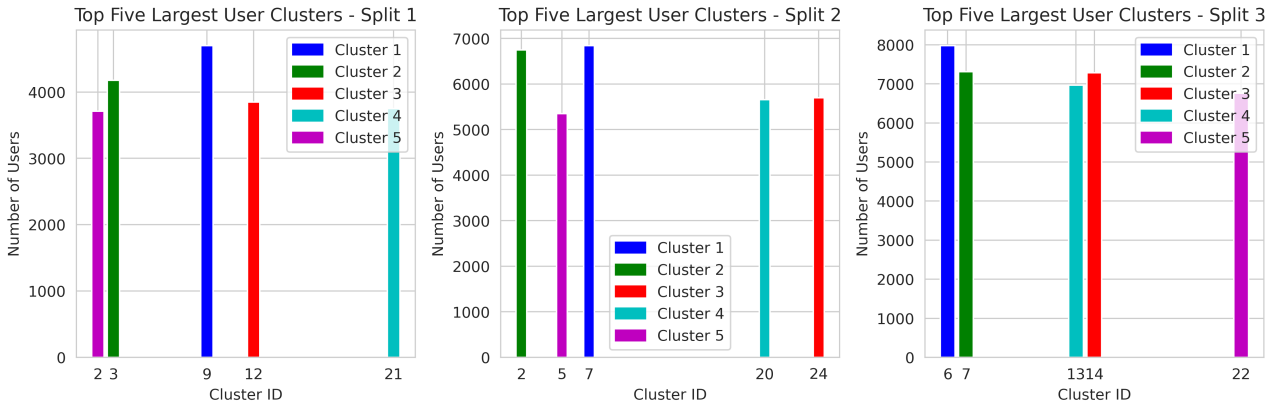

Figure 3.2: *Largest Clusters*

3.2.2 For each of the three splits in Q3 A1, consider only the largest user cluster in Q3B1 and do the following only on the training set:

a)Considering all users in the largest user cluster, find all the movies that have been rated by these users and their respective average ratings, named as movies largest cluster.

```
+-------+------------------+
|movieId|avg_rating        |
+-------+------------------+
|471    |3.718104495747266 |
|148    |3.111111111111111 |
|496    |3.6923076923076925|
|243    |1.8461538461538463|
|31     |3.073369565217391 |
+-------+------------------+
only showing top 5 rows
```

b) Find those movies in movies largest cluster with an average rating greater or equal to $4(>=4)$, named as top movies.

Using avg function movie clustered is filtered to get top movies with rating ¿ 4.

```
top_movies_1 = movies_largest_cluster_1.filter(movies_largest_cluster_1.avg_rating >= 4).select("movieId").distinct()
top_movies_2 = movies_largest_cluster_2.filter(movies_largest_cluster_2.avg_rating >= 4).select("movieId").distinct()
top_movies_3 = movies_largest_cluster_3.filter(movies_largest_cluster_3.avg_rating >= 4).select("movieId").distinct()


movies = spark.read.csv('Data/movies.csv',header=True,inferSchema=True)
```

Figure 3.3: *Top Movies*

Use movies.csv to find the genres for all the top movies and and report the top ten most popular genres (each movie may have multiple genres, separated by '—', where top refers to the number of appearances in movies). Report these 3 splits x 10 genres = 30 genres in one Table.

	Split 1 Genres	Split 1 Count	Split 2 Genres	Split 2 Count
0	Drama	281	Drama	125
1	Documentary	69	Documentary	55
2	Drama|Romance	65	Comedy	44
3	Comedy	65	Drama|War	33
4	Comedy|Drama	63	Comedy|Drama	27
5	Comedy|Drama|Romance	41	Comedy|Drama|Romance	22
6	Drama|War	39	Drama|Romance	22
7	Comedy|Romance	35	Crime|Drama	21
8	Crime|Drama	28	Comedy|Romance	18
9	Crime|Drama|Thriller	15	Crime|Drama|Thriller	12

	Split 3 Genres	Split 3 Count
0	Drama	609
1	Documentary	240
2	Drama|Romance	160
3	Comedy|Drama	134
4	Comedy	128
5	Drama|War	86
6	Crime|Drama	66
7	Comedy|Drama|Romance	64
8	Comedy|Romance	52
9	Drama|Thriller	44

3.3 Discuss two most interesting observations from A and B above, each with three sentences: 1) What is the observation? 2) What are the possible causes of the observation? 3) How useful is this observation to a movie website such as Netflix? Your report must be clearly written and your code must be well documented so that it is clear what each step is doing.

- The error values decreases slightly with increase in data for training our ALS model.This may be because with more features our ALS model better captures the implicit relationship between features more accurately. A movie website like Netflix should use this observation not only collect more data , but focus on data-points which increase implicit feedback.

- The second setting of ALS model outperforms the first setting due to being allowed to run more iterations, it is able to converge and predict more accurately. Netflix can realise the importance of using advanced models and more compute resources to get a better recommending system.

Chapter 4

Research Paper Visualisation

4.1 Use PySpark APIs to compute the top 2 principal components (PCs) on the NIPS papers. Report the two corresponding eigenvalues and the percentage of variance they have captured. Show the first 10 entries of the 2 PCs

```
First 10 entries of the 2 PCs:
 PC1:(-4.187373864219449, -13.053580670832174, -13.402833398556252, -9.518465693501682,
 -11.36210605186559, -17.27502451559686, -9.06893012471305, -14.53129493155118,
 -14.172433999569217, -11.987440697249893, -11.108393045365975)
 PC2: (0.646185617325472, 0.38433519586369336, 0.646427415343431, 0.7587197949636434,
 1.020098476527301, 0.6219632160876928, 0.85114381260986, 0.3843439207450862,
 1.023310958908378, 0.5306143092652766,
 1.06550011431261)
Eigenvalues: [0.00377015 0.00636943]
Explained_variance:  [0.05203481 0.01127954]
Retained_variance [0.82184862 0.17815138]
```

4.2 Visualise the 5811 papers using the 2 PCs, with the first PC as the x-axis and the second PC as the y-axis. Each paper will appear as a point on the figure, with coordinates determined by these top 2 PCs.

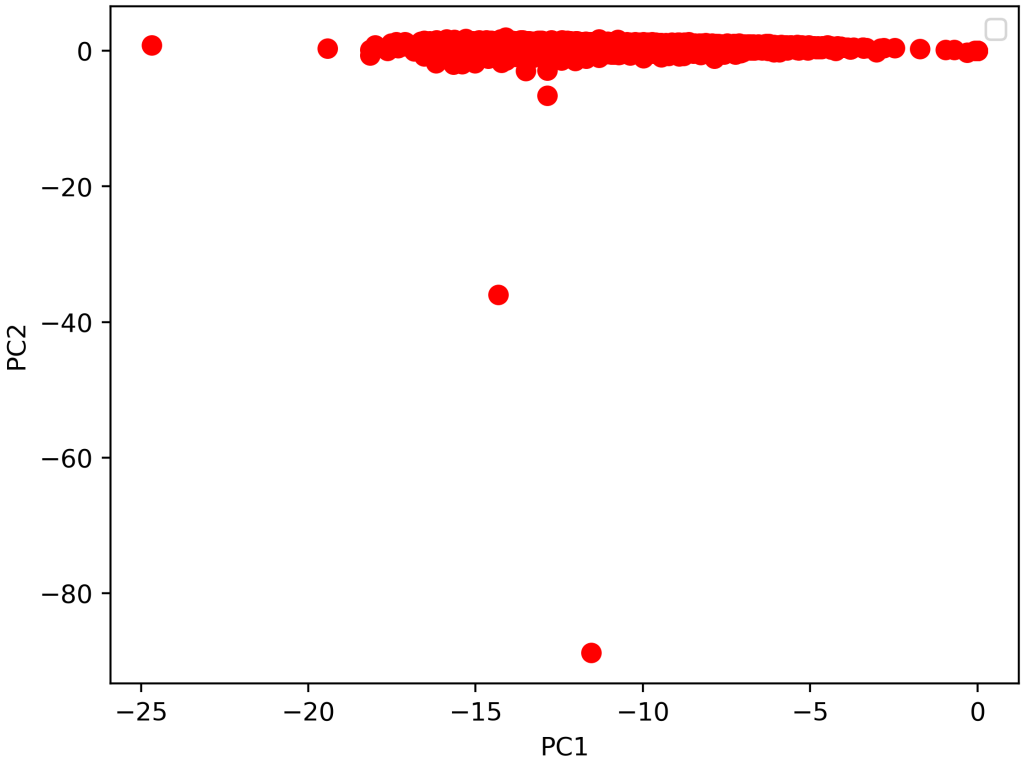

Figure 4.1: *Papers v/s PC1 and PC2*

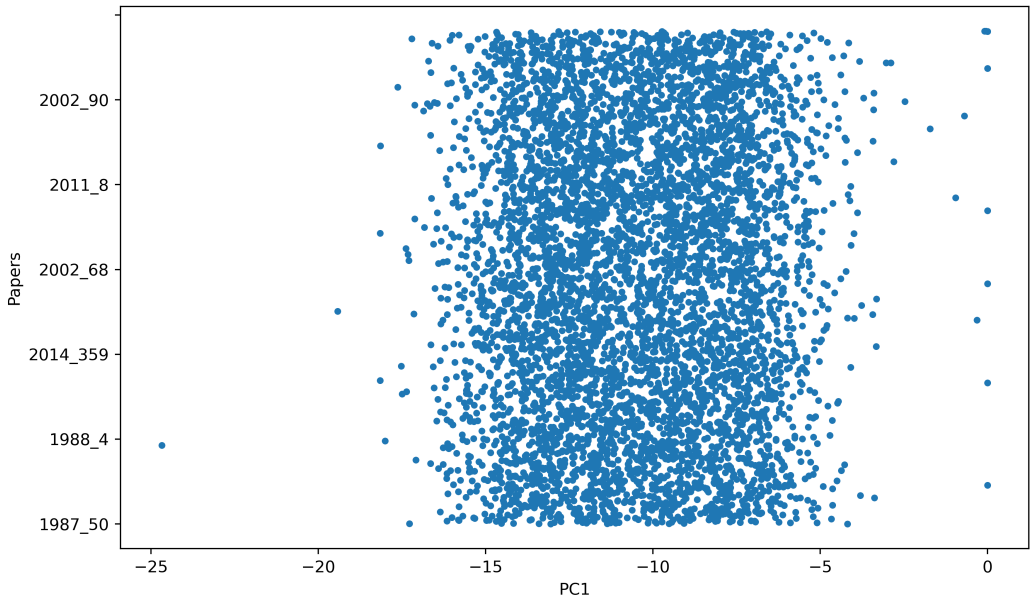

Figure 4.2: *Papers v/s PC1*

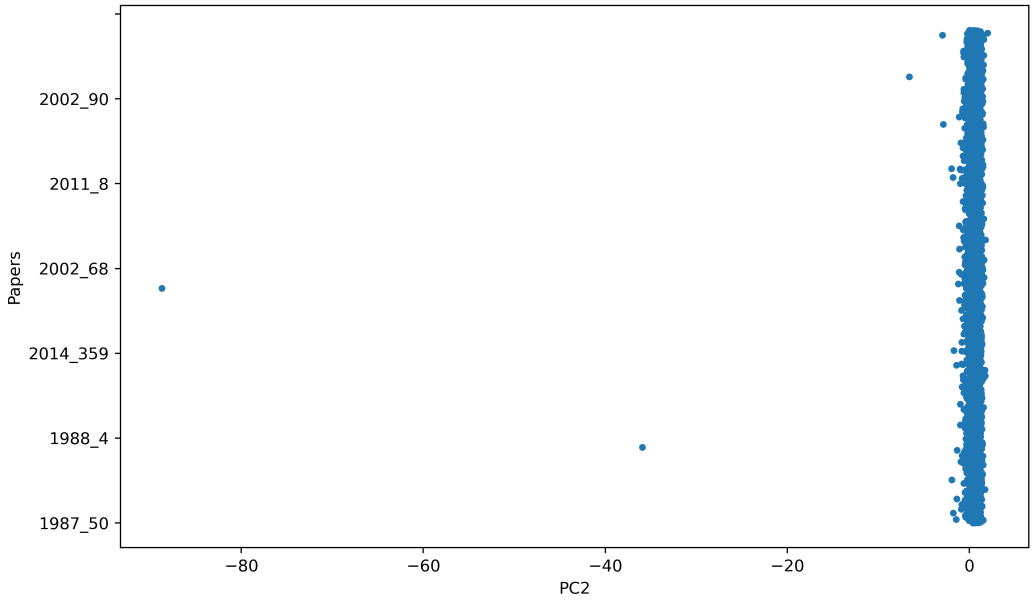

Figure 4.3: *Papers v/s PC2*

4.3 Discuss the most interesting observations from the visualisation in B, with two to three sentences. Your report must be clearly written and your code must be well documented so that it is clear what each step is doing.

- The Scatter plot is concentrated in a few points which reflects the common words in the papers which may be a result of common domain areas of knowledge, However the scatter is not discrete but continuous so the NIPS papers may cover a wide range of areas.
- PC1 accounts for most of the retained variance, so it may provide meaningful insights and demonstrates the capability of PysPark PCA model implementation as it was able to reduce the 11000 approx sparse word values to two components while still retaining most information.

Chapter 5

Searching for exotic particles in high-energy physics using ensemble methods

5.1 Use pipelines and cross validation to find the best configuration of parameters for each model

```
On a test run on 1 percent Data
GBT: {
Accuracy0.6390595212878627, AUC: 0.6397027777262763
}

RF Accuracy: 0.6391
RF AUC: 0.6939
```

5.1.1 For finding the best configuration of parameters, use 1% of the data chosen randomly from the whole set. Hint: think of proper class balancing while picking your randomly chosen subset of data. Pick three parameters for each of the two models and use a sensible grid of three options for each of those parameters

```
Best Parameters for GBT
maxDepth : 8
 MaxBins=20
 maxIter15

Best Parameters for RF
maxDepth : 12
 MaxBins=30
 numTrees30
```

5.1.2 Use the same splits of training and test data when comparing performances among the algorithms

```
    On best Hyperparameters settings the performance :
GBT: {accuracy_gbt_cv0.7046933155808394
 area_under_curve_gbt_cv0.7034034073502482}

 RF : {
 area_under_curve_rf_cv0.7883791022921786
 accuracy_rf_cv0.7134989560323176
 }
```

5.2 Working with the larger dataset. Once you have found the best parameter configurations for each algorithm in the smaller subset of the data, use the full dataset to compare the performance of the two algorithms in the cluster

5.2.1 Use the best parameters found for each model in the smaller dataset of the previous step, for the models used in this step

```
For Random Forest:
{
    "bootstrap": true,
    "cacheNodeIds": false,
    "checkpointInterval": 10,
    "featureSubsetStrategy": "auto",
    "featuresCol": "features",
    "impurity": "gini",
    "labelCol": "label",
    "leafCol": "",
    "maxBins": 30,
    "maxDepth": 12,
    "maxMemoryInMB": 256,
    "minInfoGain": 0.0,
    "minInstancesPerNode": 1,
    "minWeightFractionPerNode": 0.0,
    "numTrees": 30,
    "predictionCol": "prediction",
    "probabilityCol": "probability",
    "rawPredictionCol": "rawPrediction",
    "seed": -5387697053847413545,
    "subsamplingRate": 1.0
}
```

```
For GBTClassifier:
{
    "cacheNodeIds": false,
    "checkpointInterval": 10,
    "featureSubsetStrategy": "all",
    "featuresCol": "features",
    "impurity": "variance",
    "labelCol": "label",
    "leafCol": "",
    "lossType": "logistic",
    "maxBins": 20,
    "maxDepth": 8,
    "maxIter": 15,
    "maxMemoryInMB": 256,
    "minInfoGain": 0.0,
    "minInstancesPerNode": 1,
    "minWeightFractionPerNode": 0.0,
    "predictionCol": "prediction",
    "probabilityCol": "probability",
    "rawPredictionCol": "rawPrediction",
    "seed": 6,
    "stepSize": 0.1,
    "subsamplingRate": 0.5,
    "validationTol": 0.01
}
```

5.2.2 Once again, use the same splits of training and test data when comparing performances between the algorithms

```
Accuracy_final_GBT:0.7167746722940492
AUC_final_GBT: 0.7150005934208362
Accuracy_final_RF: 0.7163
AUC_final_RF: 0.7922
```

Bibliography