Vellore Institute of Technology, Vellore

Final Project Report

On

Community Forest Algorithm Implementation

Organization:

Reliance Jio Infocomm Limited

Ghansoli, Navi Mumbai – 400 701

By:

Ashutos Mahapatra

B'Tech: EEE | Semester: VI

Duration: 2nd June 2019 – 4th July 2019

AKNOLEDGEMENT

Mr. George Cherian (Industry Mentor, Head of Data Science Team)

Mr. George Cherian has been my project guide and mentor in my time at Jio. I have learnt a lot under him while working on the Community Forest Algorithm. He has been extremely generous and a great teacher sharing salient lessons and deeper insights of the corporate world and influencing me push my limits further in order to achieve excellence.

Dr. Shailesh Kumar (Chief Data Scientist at Jio)

Dr. Shailesh Kumar is the chief data scientist at Reliance Jio and The Community Forest Algorithm was theorised by him that I have been working on. He has been extremely approachable in helping me understand the algorithm and the essence of it.

Mr. Anurag Sahoo (AI CoE)

Mr. Anurag Sahoo is part of the Artificial Intelligence Centre of Excellence at Hyderabad. He works closely with Dr. Shailesh. It was with support that I have implemented the Community Forest Algorithm. He has been immensely supportive in explaining how code must be oriented.

ABOUT THE COMPANY

Reliance Jio Infocomm Limited, d/b/a Jio, is an Indian mobile network operator. Owned by Reliance Industries and headquartered in Mumbai, Maharashtra, it operates a national LTE network with coverage across all 22 telecom circles. Jio does not offer 2G or 3Gservice, and instead uses voice over LTE to provide voice service on its network.

Jio soft launched on 27 December 2015 (the eve of what would have been the 83rd birthday of Reliance Industries founder Dhirubhai Ambani), with a beta for partners and employees, and became publicly available on 5 September 2016. As of 31 March 2019, it is the second largest mobile network operator in India and the sixth largest mobile network operator in the world with over 306 million subscribers.

On 5 July 2018, fixed line broadband service named Gigafiber, was launched by the Reliance Industries Limited's chairman Mukesh Ambani, during the company's Annual General Meeting.

ABOUT THE DEPARTMENT

Jio Data Science Platform

JDSP is responsible for designing and implementing processes and layouts for complex, largescale data sets used for modelling, data mining, and research purposes.

It focuses on business case development, planning, coordination / collaboration with multiple teams & project, managing the life-cycle of an analysis project, and interface with business sponsors to provide periodic updates.

Data scientists perform data analysis, develop algorithms and predictive models, and deliver reports for a broad audience. They apply algorithms and modelling tools to a variety of data assets and develop critical solutions for multiple areas of the business.

ABOUT THE PROJECT

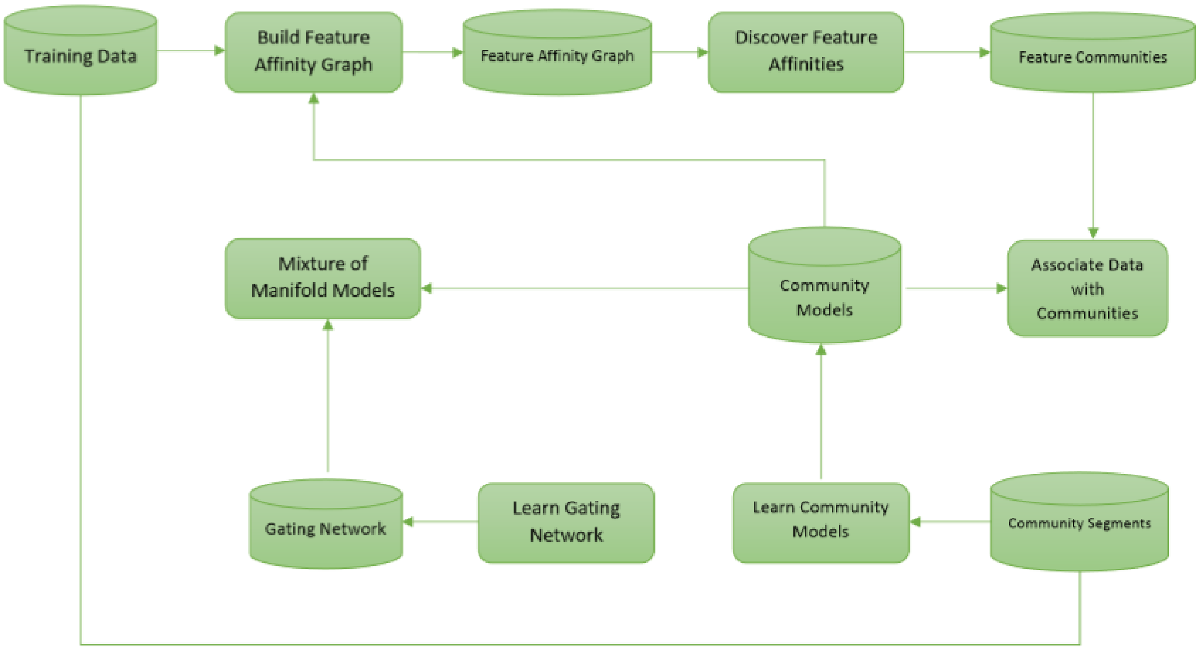
Community Forest Algorithm is an Ensemble Learning Technique in which logical communities of related features are constructed on the basis of mutual affinity of features in a dataset. It employs a Gating Mechanism in the second stage where records are clustered into community based on their likelihood of being closely associated to a specific community. This algorithm outperforms the pre-existing Supervised Machine Learning Algorithms and gives detailed insight of the data.

The main goal of this algorithm is to give insights to users as to what group of features lead to a certain metric or prediction. The main application of this algorithm lies in cases where the data is non-homogenous i.e. the data is a combination of multiple sources of data, and where the number of features in the data is large.

The community forest algorithm has been divided into 4 phases:

- Phase 0: Data Hygiene
- Phase 1: Feature Affinity Graph generation
- Phase 2: Community Detection of features
- Phase 3: Modelling each Community
- Phase 4: Gating Mechanism

ARCHITECTURE OF COMMUNITY FOREST



Phase 0: Data Hygiene

- The raw data goes through a data agnostic hygiene module which generates a cleaned data (csv file).
- Reading the raw data as a dataframe.
- A target list and column drop list of rows and columns having NaN values more than a certain threshold are created and eventually dropped

Phase 1: Feature Affinity Graph generation:

- This phase generates an affinity graph from the cleaned data it receives as an input.
- The affinity graph is calculated by taking all permutations and combinations of the entire cleaned feature list.
- The affinity can be calculated by using any method. This method can be given in the form of a hyperparameter.
- The Affinity Graph generated by this phase contains the features as nodes and the affinity as weights. Thus, the output is finally a weighted graph.

Phase 2: Community Detection of features:

- This algorithm receives the Affinity Graph from the previous phase as an input.
- This algorithm then performs various operations on the Affinity Graph and detects communities from it.
- The output of this phase is a whole list of unique communities

Phase 3: Modelling each Community:

- This phase is used to create a model for each and every community individually.
- The ML algorithm can be selected wither by taking it directly from the hyper parameter or by iterating over simple ML algorithms and choosing the one with the best accuracy

Phase 4: Gating Mechanism:

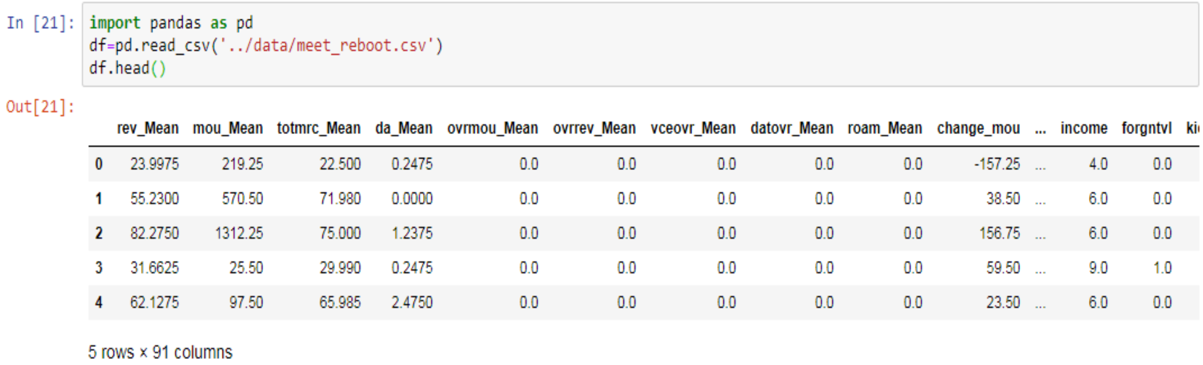
- The gating mechanisms is used to allocate new data points to each community.
- This can be done by comparing confidence scores returned from all the community models generated in phase 3 and applying a function such as ‘arg max’ to it.

- The gating mechanism is extremely crucial in the entire process as it reiteratively trains the models.

PHASE: 0

Data Cleaning

- The raw data goes through a data agnostic hygiene module which generates a cleaned data (csv file).
- Reading the raw data as a dataframe.
- A target list and column drop list of rows and columns having NaN values more than a certain threshold are created and eventually dropped.
- Then the HLDT classifier classifies the column fields as LongLat,NumCat, Categorical, ID, DateTime, IP Address, Logistic Classification and URL.
- Non-Required column fields (for eg: ID fields) are dropped.
- Day of Week and Month of Year from Date Time objects are extracted.
- Imputing Categorical Data by taking Mode and Numerical Data by computing Mean/Median.
- Performing Encoding and Binning.
- Exporting the cleaned data as a new csv file.



```
In [21]: import pandas as pd
         df=pd.read_csv('../data/meet_reboot.csv')
         df.head()
```

Out[21]:

	rev_Mean	mou_Mean	totmrc_Mean	da_Mean	ovrmou_Mean	ovrrev_Mean	vceovr_Mean	datovr_Mean	roam_Mean	change_mou	...	income	forgntvl	ki
0	23.9975	219.25	22.500	0.2475	0.0	0.0	0.0	0.0	0.0	-157.25	...	4.0	0.0	
1	55.2300	570.50	71.980	0.0000	0.0	0.0	0.0	0.0	0.0	38.50	...	6.0	0.0	
2	82.2750	1312.25	75.000	1.2375	0.0	0.0	0.0	0.0	0.0	156.75	...	6.0	0.0	
3	31.6625	25.50	29.990	0.2475	0.0	0.0	0.0	0.0	0.0	59.50	...	9.0	1.0	
4	62.1275	97.50	65.985	2.4750	0.0	0.0	0.0	0.0	0.0	23.50	...	6.0	0.0	

5 rows × 91 columns

Implementation: Output

PHASE: 1

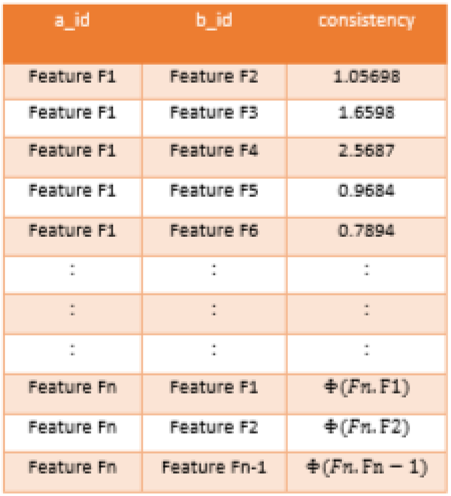
(Feature Affinity Graph Generation)

- The cleaned csv file is then read and passes through the affinity graph generator module.
- Selection of a model with respect to a given target to use on the features and the target pair. The model is set as a hyper parameter:
  a. Decision Tree
  b. Random Forest
  c. Logistic Regression
  d. XG Boost
- Goodness score is calculated given a model. Goodness is also hyper parameterized:
  i. Accuracy
  ii. F1 score
  iii. AUC
  iv. Precision
- Now calculating goodness of individual feature **G(i|M).**
- Then calculating the goodness of pairs of features **G(i,j|M).**
- Now calculating the strength between the pair of features:

$$\boldsymbol{\varphi}(\mathbf{i}.\mathbf{j}) = \frac{\mathbf{G}(\mathbf{i},\mathbf{j}|\mathbf{M})}{\sqrt{\mathbf{G}(\mathbf{i}|\mathbf{M}) * \mathbf{G}(\mathbf{j}|\mathbf{M})}}$$

- Hence creating an affinity graph csv file consisting of the all the pair of features and the subsequent weights between them.

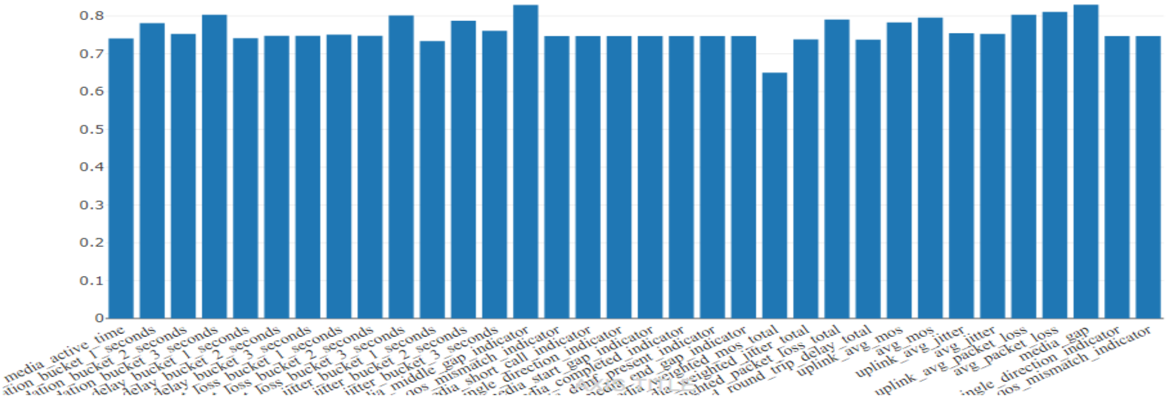
This mapping must then be stored for future reference. Thus, the output of this phase would be a csv file that would look like this:



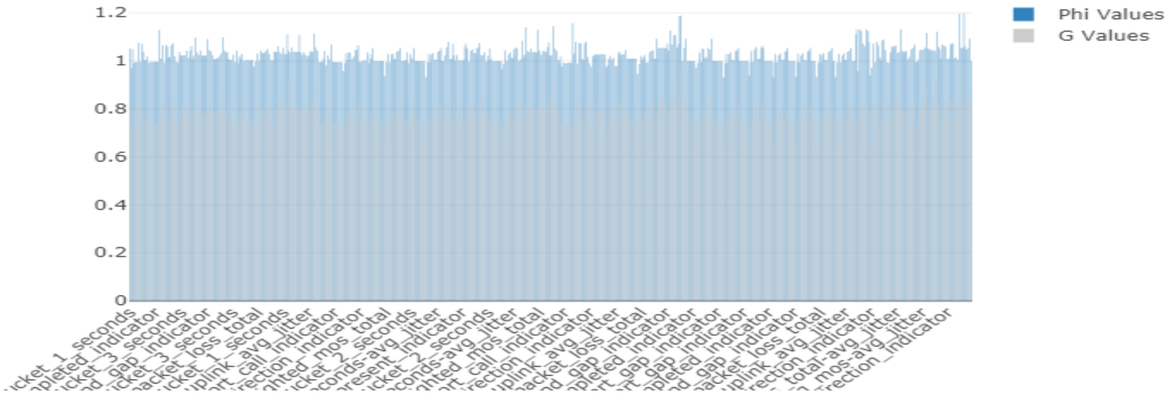
a_id	b_id	consistency
Feature F1	Feature F2	1.05698
Feature F1	Feature F3	1.6598
Feature F1	Feature F4	2.5687
Feature F1	Feature F5	0.9684
Feature F1	Feature F6	0.7894
:	:	:
:	:	:
:	:	:
Feature Fn	Feature F1	$\Phi(Fn.F1)$
Feature Fn	Feature F2	$\Phi(Fn.F2)$
Feature Fn	Feature Fn-1	$\Phi(Fn.Fn-1)$

Visualization: Outputs

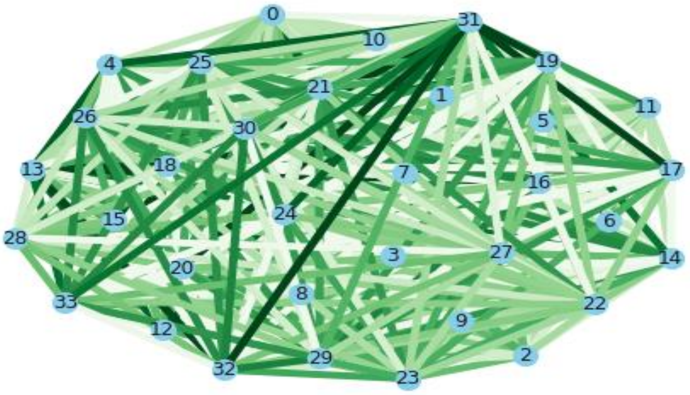
1 feature histogram:

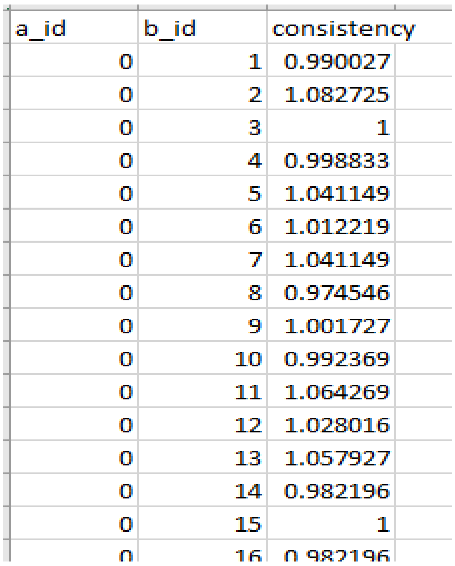


2 feature histogram:



Affinity Graph:





a_id	b_id	consistency
0	1	0.990027
0	2	1.082725
0	3	1
0	4	0.998833
0	5	1.041149
0	6	1.012219
0	7	1.041149
0	8	0.974546
0	9	1.001727
0	10	0.992369
0	11	1.064269
0	12	1.028016
0	13	1.057927
0	14	0.982196
0	15	1
0	16	0.982196

PHASE 2

(Community Detection of features):

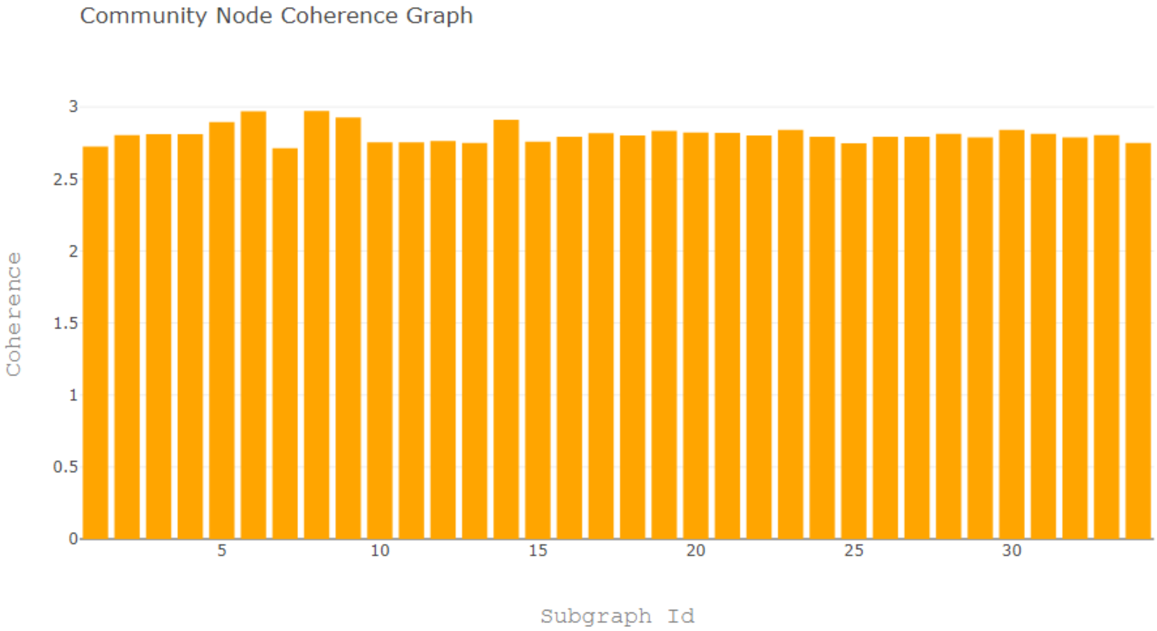
In the second phase the affinity graph is fetched into the community detection transformer which generates feature communities. The algorithm used to detect communities is the intellectual property of Reliance Jio and hence cannot be disclosed.

- The affinity graph is fetched into the community detection transformer which generates feature communities.
- We start with a random node and find all its neighbouring nodes.
- Applying the grow shrink algorithm on the node.
- For each subgraph, the subgraph nodes, subgraph edges, subgraph local node centrality and the subgraph coherence score for each seed node is calculated.
- A soft maximal clique of features is constructed such that removing or adding a feature decreses its coherence.
- Coherence score for each community is calculated:

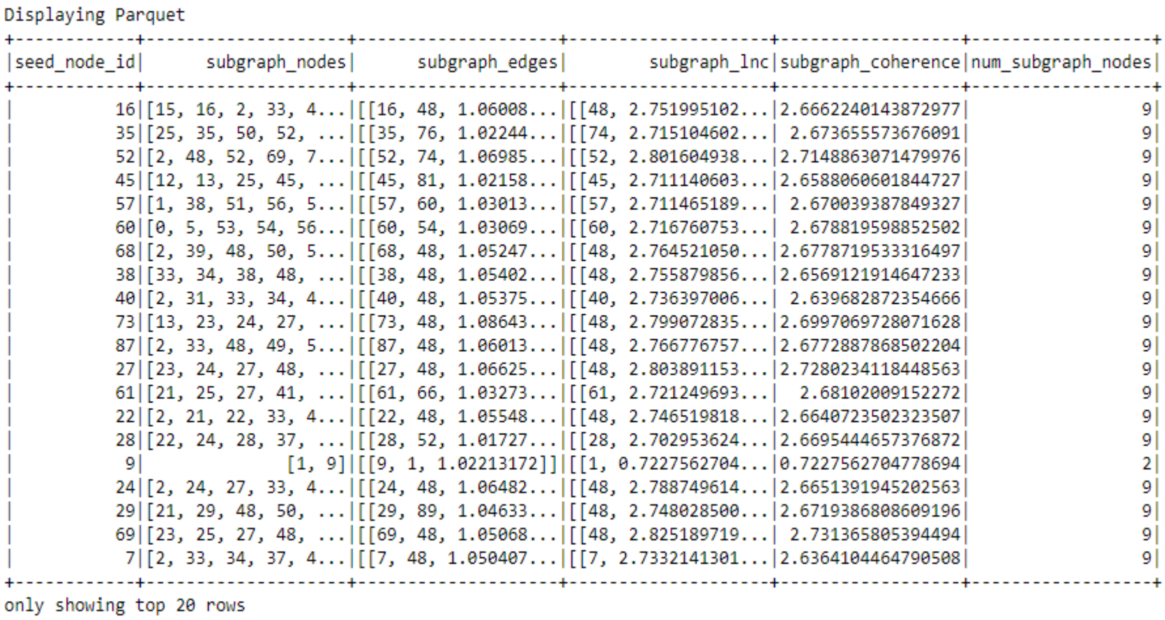
$$\boldsymbol{\pi}(\mathbf{c}) = \boldsymbol{\lambda}_{\mathbf{1}}(\mathbf{c}) * \mathbf{min}\{\mathbf{v}_{\mathbf{1}}(\mathbf{c})\}$$

- where λ1(C) is the first eigen value and v1(C) is the first eigen vector of the affinity graph submatrix.
- Thus the seed node ids having unique subgraph nodes form separate individual communities which is exported as a parquet file.

Subgraph-id vs. coherence score:



Community detection parquet:



```
Displaying Parquet
+------------+--------------------+--------------------+--------------------+------------------+------------------+
|seed_node_id|      subgraph_nodes|      subgraph_edges|        subgraph_lnc|subgraph_coherence|num_subgraph_nodes|
+------------+--------------------+--------------------+--------------------+------------------+------------------+
|          16|[15, 16, 2, 33, 4...|[[16, 48, 1.06008...|[[48, 2.751995102...|2.6662240143872977|                 9|
|          35|[25, 35, 50, 52, ...|[[35, 76, 1.02244...|[[74, 2.715104602...| 2.673655573676091|                 9|
|          52|[2, 48, 52, 69, 7...|[[52, 74, 1.06985...|[[52, 2.801604938...|2.7148863071479976|                 9|
|          45|[12, 13, 25, 45, ...|[[45, 81, 1.02158...|[[45, 2.711140603...|2.6588060601844727|                 9|
|          57|[1, 38, 51, 56, 5...|[[57, 60, 1.03013...|[[57, 2.711465189...| 2.670039387849327|                 9|
|          60|[0, 5, 53, 54, 56...|[[60, 54, 1.03069...|[[60, 2.716760753...| 2.678819598852502|                 9|
|          68|[2, 39, 48, 50, 5...|[[68, 48, 1.05247...|[[48, 2.764521050...|2.6778719533316497|                 9|
|          38|[33, 34, 38, 48, ...|[[38, 48, 1.05402...|[[48, 2.755879856...|2.6569121914647233|                 9|
|          40|[2, 31, 33, 34, 4...|[[40, 48, 1.05375...|[[40, 2.736397006...| 2.639682872354666|                 9|
|          73|[13, 23, 24, 27, ...|[[73, 48, 1.08643...|[[48, 2.799072835...|2.6997069728071628|                 9|
|          87|[2, 33, 48, 49, 5...|[[87, 48, 1.06013...|[[48, 2.766776757...|2.6772887868502204|                 9|
|          27|[23, 24, 27, 48, ...|[[27, 48, 1.06625...|[[48, 2.803891153...|2.7280234118448563|                 9|
|          61|[21, 25, 27, 41, ...|[[61, 66, 1.03273...|[[61, 2.721249693...|  2.68102009152272|                 9|
|          22|[2, 21, 22, 33, 4...|[[22, 48, 1.05548...|[[48, 2.746519818...|2.6640723502323507|                 9|
|          28|[22, 24, 28, 37, ...|[[28, 52, 1.01727...|[[28, 2.702953624...|2.6695444657376872|                 9|
|           9|              [1, 9]|[[9, 1, 1.02213172]]|[[1, 0.7227562704...|0.7227562704778694|                 2|
|          24|[2, 24, 27, 33, 4...|[[24, 48, 1.06482...|[[48, 2.788749614...|2.6651391945202563|                 9|
|          29|[21, 29, 48, 50, ...|[[29, 89, 1.04633...|[[48, 2.748028500...|2.6719386808609196|                 9|
|          69|[23, 25, 27, 48, ...|[[69, 48, 1.05068...|[[48, 2.825189719...| 2.731365805394494|                 9|
|           7|[2, 33, 34, 37, 4...|[[7, 48, 1.050407...|[[7, 2.7332141301...|2.6364104464790508|                 9|
+------------+--------------------+--------------------+--------------------+------------------+------------------+
only showing top 20 rows
```

PHASE 3

(Modelling each Community)

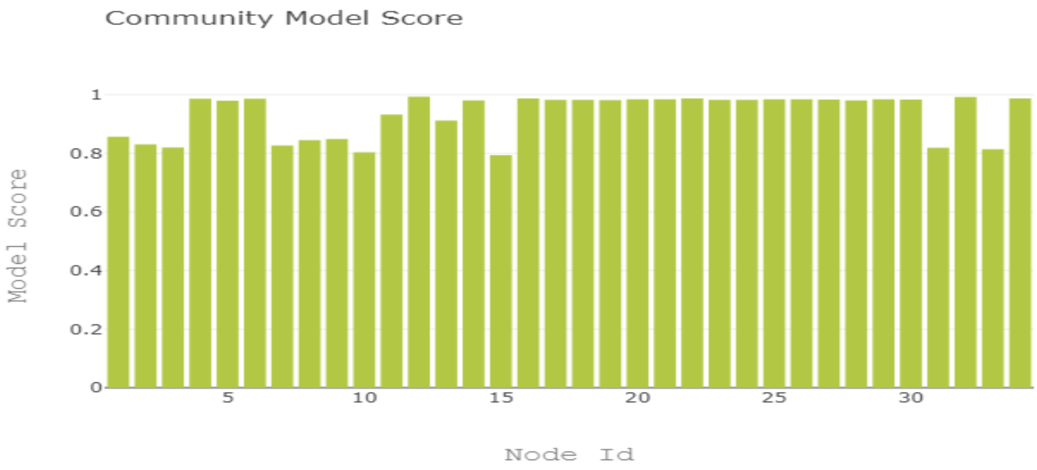
- In this phase we read the parquet file from previous phase and model each community.
- A dataframe is created for each community.
- The dataset is split into train and test dataset.
- Intializing data association into each communities: Initially all data points are equally associated with all communities.

  $$\boldsymbol{\Delta} = \boldsymbol{\delta(n, k)}$$

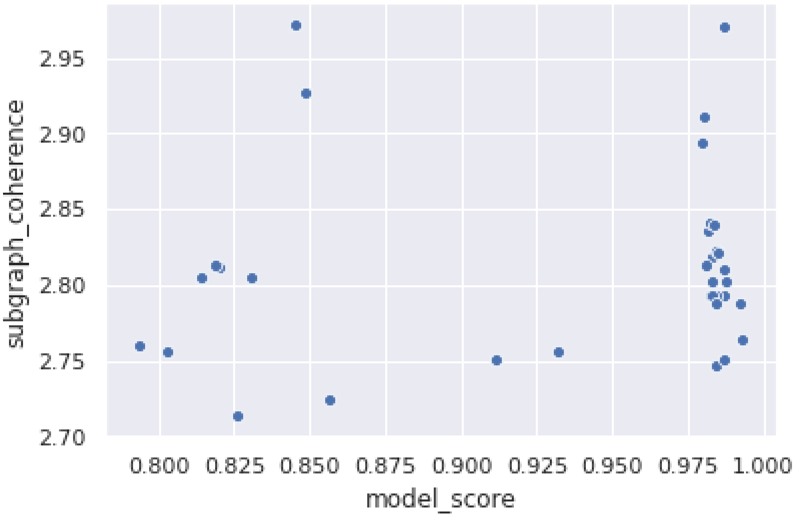
  It is the association of (X^n,Y^n) with community C_k.

- Now training the model on the training dataset of each community.
  - Decision Tree
  - Random Forest
  - Logistic Regression
  - XG Boost
- Calculating the model score for each community.
  - Accuracy
  - F1 score
  - AUC
  - Precision

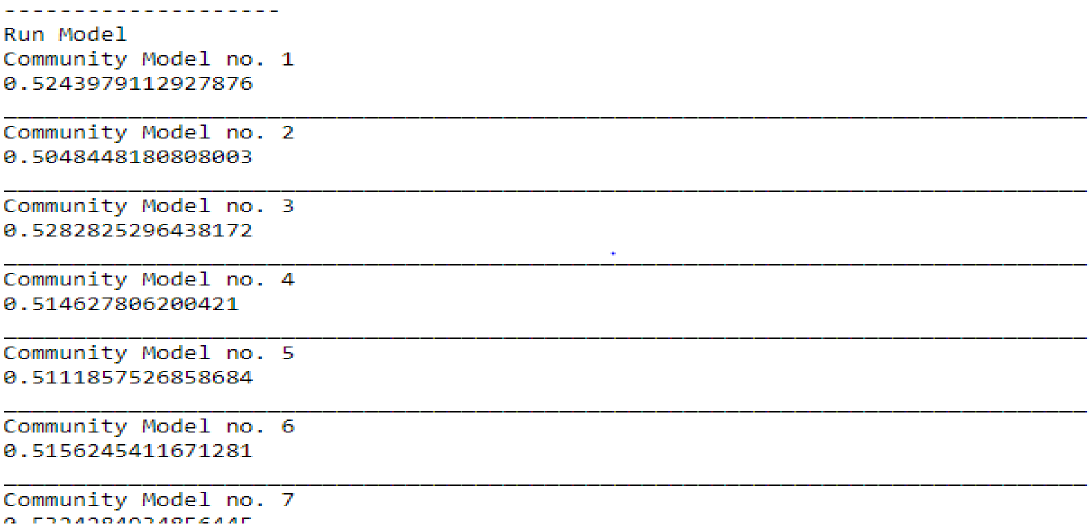
Node-id vs. model score:



Scatter plot of model score vs. subgraph coherence:



Community Model score:



```
--------------------
Run Model
Community Model no. 1
0.5243979112927876
__________________________________________________________________________
Community Model no. 2
0.5048448180808003
__________________________________________________________________________
Community Model no. 3
0.5282825296438172
__________________________________________________________________________
Community Model no. 4
0.514627806200421
__________________________________________________________________________
Community Model no. 5
0.5111857526858684
__________________________________________________________________________
Community Model no. 6
0.5156245411671281
__________________________________________________________________________
Community Model no. 7
0.5234284034856445
```

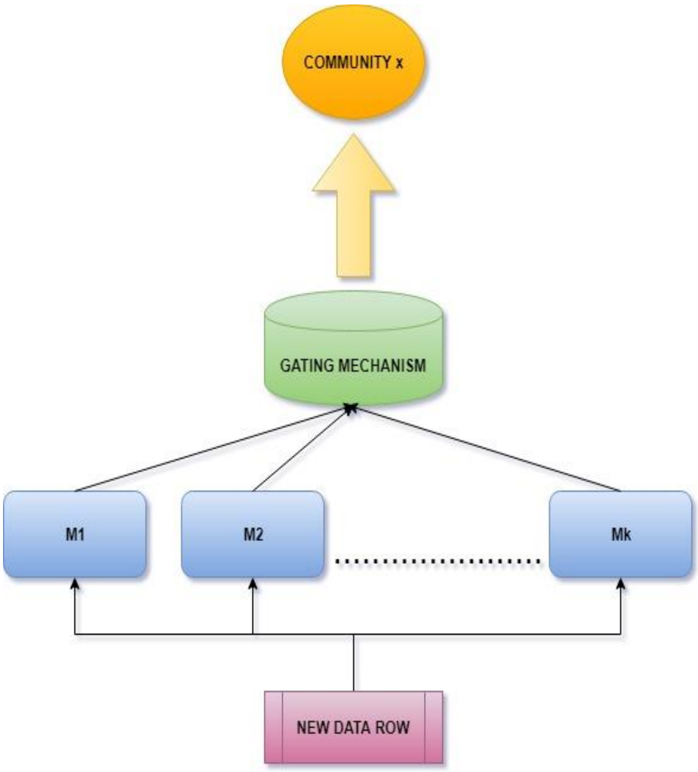
PHASE 4

(Gating Mechanism)

The gating mechanism is the last phase of the Community Forest Algorithm and also the most important. The main aim of this phase is to take a data point as an input and assign it to a community. For this we will give the data as an input to each and every community model and based on a scoring criterion we will assign the data to a community.

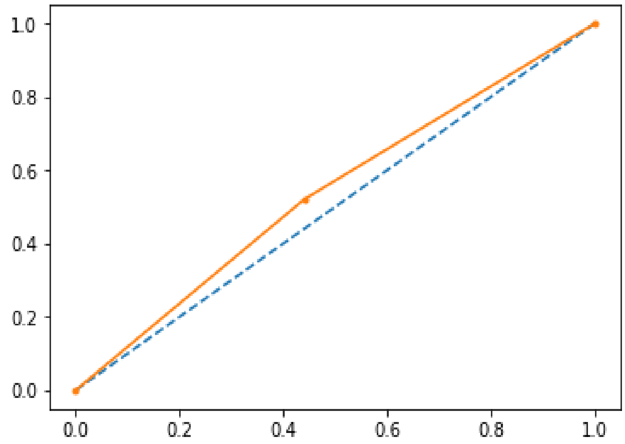
Currently for implementation purpose we will be choosing the community with the highest confidence score to be assigned the data. Thus, this Gating mechanism will help the user understand the data belongs to which community. With this module you can have real time implementation of various functions that will help transform businesses.

- The new data row is sent to each model corresponding to each community (M1, M2, M3,………,Mk).
- Prediction for each row is calculated given the community.
  x= new data
  Mi= model for a community i
  prediction score = P(x|Mi)
- Max vote count is used to predict the community
- Max_count(P(x|M1), P(x|M2),….P(x|Mk))

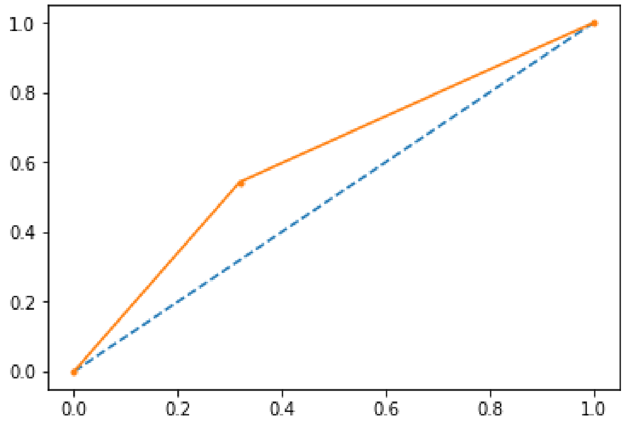


Model	Decision Tree	Random Forest	XG boost	Community Forest
ROC-AUC score	0.540	0.612	0.626	0.650

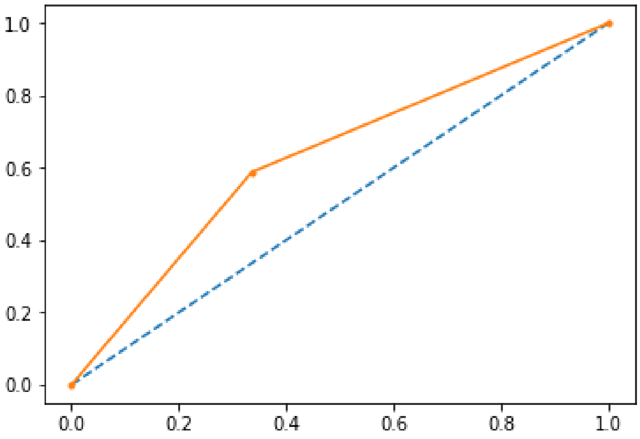
DECISION TREE



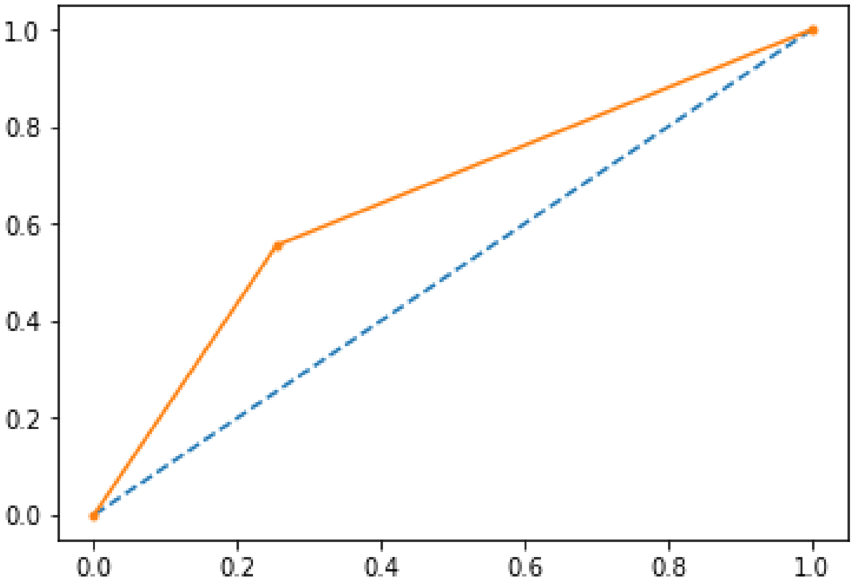
RANDOM FOREST



XG BOOST



COMMUNITY FOREST



CONCLUSION:

We propose a novel method to create logical cluster of features (called communities) and classifying data into them. It serves two purposes.

Firstly, we are able to interpret the communities and derive knowledge which may be helpful while making business decisions.

Secondly, in datasets with high dimensionality (>50), Community Forest has proven to classify records more accurately than XGBoost and Random Forest algorithms. Although the evaluations are done mainly for Telecom related data, the generality of the process enables it to be applied for any business use case where the dimensionality is large.

REFERENCES

1. https://stackoverflow.com/
2. https://github.com/
3. https://www.tutorialspoint.com/
4. https://www.getpostman.com/
5. https://jsonlint.com/