GRADIENT

GRADIENT

Project Name

GRADIENT (**GRA**phical **D**rug **I**nsights for **E**ffective **T**reatment)

Contributors

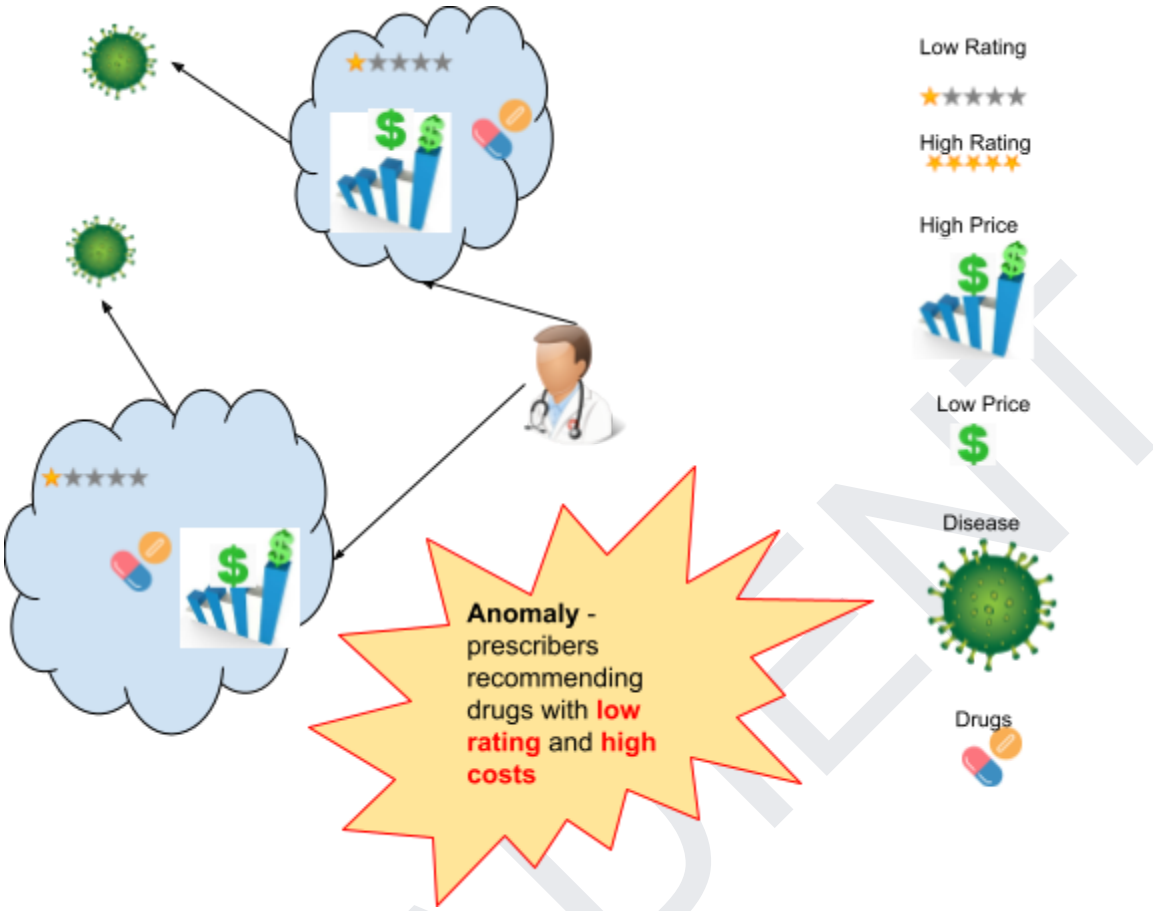
Kaniska Mandal
kaniska.mandal@gmail.com
https://www.linkedin.com/in/kaniskamandal/

Tyrone Hou
tyroneh@bu.edu
https://www.linkedin.com/in/tyronehou/

Goal

This project aims at developing functionalities using TigerGraph which can help review the quality of drugs used for treatment of diseases and identify potential anomalies like if the prescribers are using low rated drugs. Our goal is to improve healthcare by extracting useful insights from the usage of drugs, discovering anomalies and help saving lifes.

GRADIENT

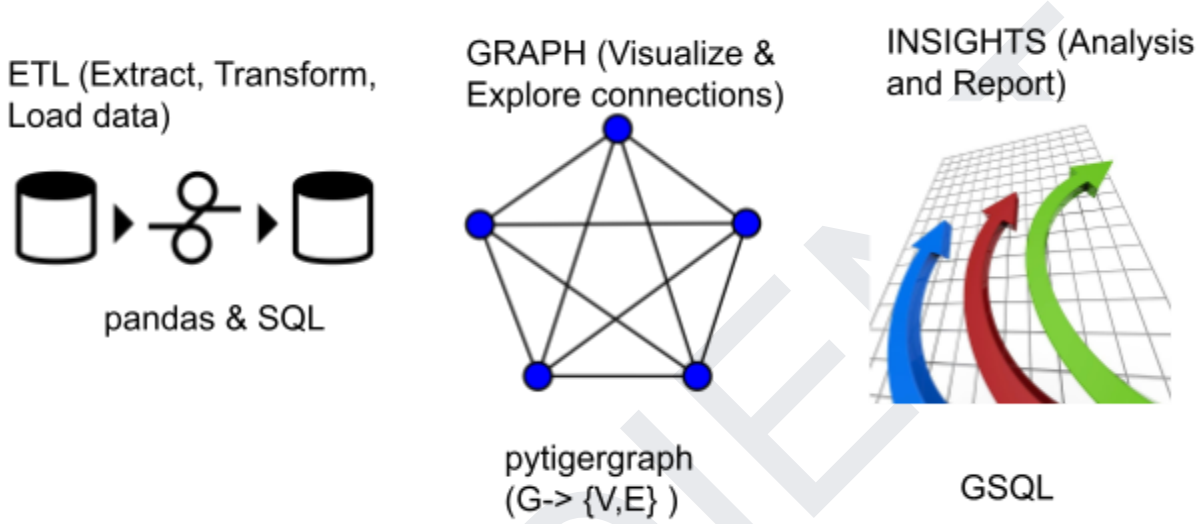


Data Set

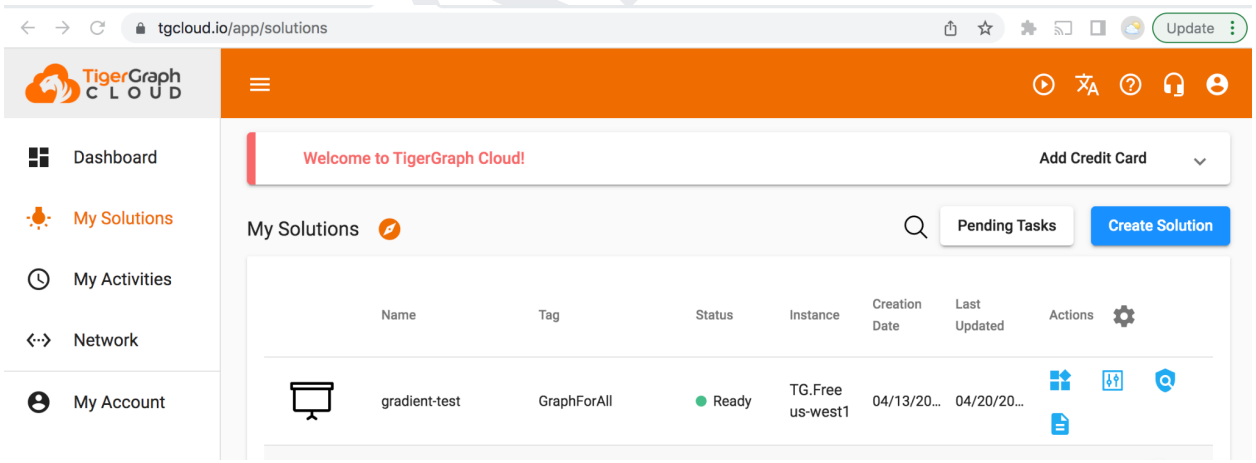
- Drug Bank Ids
  https://go.drugbank.com/releases/latest#open-data
- Stanford Biosnap databases
  Disease Synopsis
  http://snap.stanford.edu/biodata/datasets/10003/10003-D-MeshMiner.html
- Disease-Drug associations
  http://snap.stanford.edu/biodata/datasets/10004/10004-DCh-Miner.html
- Medicare Part D Prescribers - by Provider and Drug
  https://data.cms.gov/provider-summary-by-type-of-service/medicare-part-d-prescribers/medicare-part-d-prescribers-by-provider-and-drug/data

GRADIENT

Design & Development Approach



Create a Tigergraph Cloud Account



Use Jupyter Notebook

GRADIENT

- We have used Jupyter Notebook to fetch and preprocess data (ETL) and create graphs by connecting it with TigerGraph Cloud.

Use TigerGraph Studio

- TigerGraph Studio was used for executing GSQL queries to extract insights.

Implementation Process

Setup Github Project

Codebase: https://github.com/kaniska/gradient

GSQL Queries: https://github.com/kaniska/gradient/tree/main/core/graph

Notebook: https://github.com/kaniska/gradient/tree/main/notebooks

Fetch dataset

Dataset of Drug Rating Per Review

review_id	drugName	condition	review	rating	date	Gnrc_Name
0	Valsartan	Left Ventricular Dysfunction	"It has no side effect, I take it in combination of Bystolic 5 Mg and Fish Oil"	9.0	May 20, 2012	Valsartan
1	Cialis	Benign Prostatic Hyperplasia	"2nd day on 5mg started to work with rock hard erections however experianced headache, lower bowel preassure. 3rd day erections would wake me up & hurt! Leg/ankles aches severe lower bowel preassure like you need to go #2 but can't! Enjoyed the initial rockhard erections but not at these side effects or $230 for months supply! I'm 50 & work out 3Xs a week. Not worth side effects!"	2.0	November 28, 2015	Tadalafil

Dataset of Drugs prescribed by Doctors

```
In [34]: filtered_prescriber_drug_names_df.iloc[:3]
```

Out[34]:

drug_id	Prscrbr_NPI	Prscrbr_Last_Org_Name	Prscrbr_First_Name	Prscrbr_Type	Brnd_Name	Gnrc_Name	Tot_Drug_Cst	GE65_Tot_Drug_Cst
3	1003000126	Enkeshafi	Ardalan	Internal Medicine	Azithromycin	Azithromycin	132.57	-1.00
4	1003000126	Enkeshafi	Ardalan	Internal Medicine	Carvedilol	Carvedilol	118.53	118.53
5	1003000126	Enkeshafi	Ardalan	Internal Medicine	Cefdinir	Cefdinir	203.03	-1.00

GRADIENT

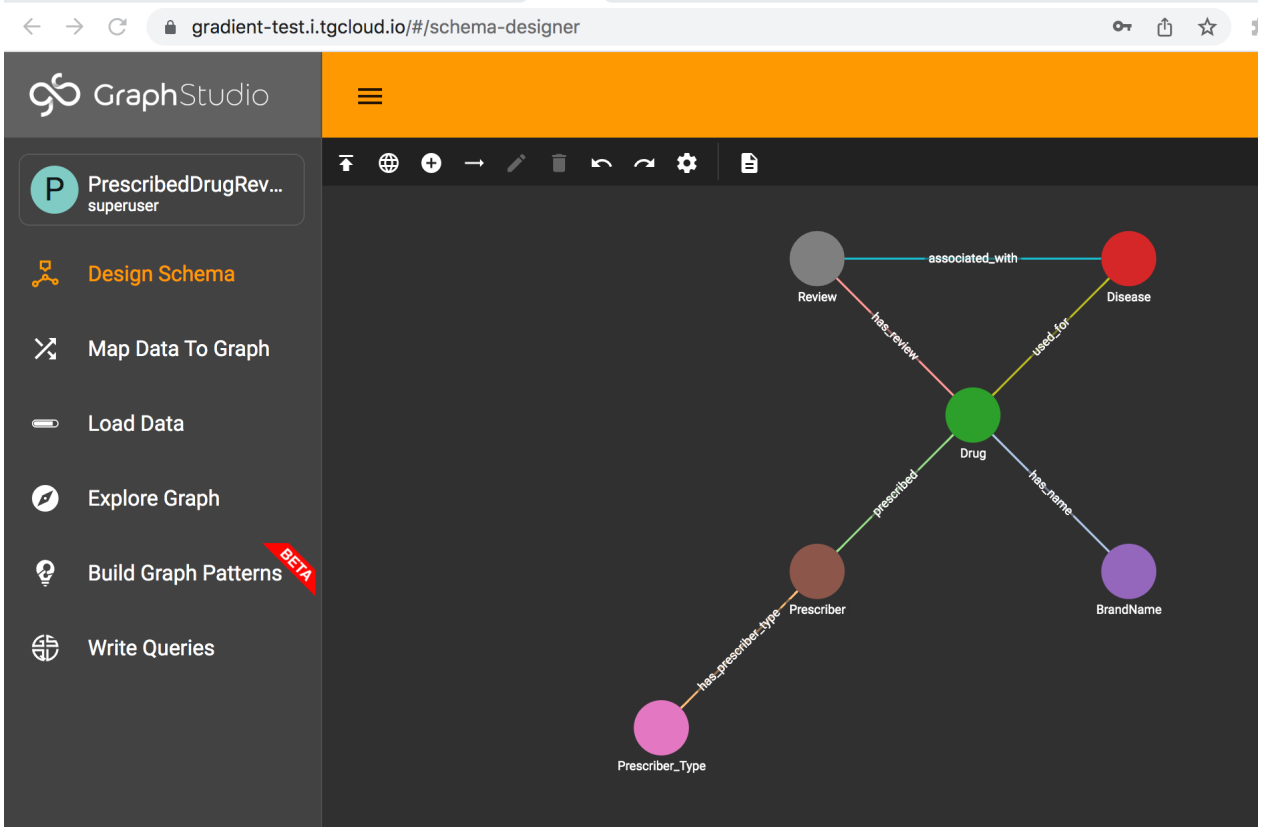
[Run this Notebook to explore and preprocess the dataset]

Once the initial exploration is done, we created 4 types of output dataset required for populating the Nodes and Edges of graph

1. reviews_df
2. drug-disease.tsv
3. drug-review.tsv
4. disease-review.tsv

Create Graph

[Run this Notebook to create Graph]



Schema information

Vertex information:

GRADIENT

Vertex: Drug
(PRIMARY ID) generic_name: STRING

Vertex: Disease
(PRIMARY ID) disease_name: STRING
description: STRING

Vertex: BrandName
(PRIMARY ID) name: STRING

Vertex: Prescriber
(PRIMARY ID) prescriber_npi: STRING
first_name: STRING
last_name: STRING

Vertex: Prescriber_Type
(PRIMARY ID) prescriber_type: STRING

Vertex: Review
(PRIMARY ID) review_id: STRING
date: DATETIME
rating: INT
review: STRING

Edge information:

Edge: used_for
Source: Drug
Target: Disease

Edge: associated_with
Source: Review
Target: Disease

Edge: has_name
Source: Drug
Target: BrandName

Edge: has_prescriber_type
Source: Prescriber
Target: Prescriber_Type

GRADIENT

Edge: prescribed
Source: Prescriber
Target: Drug
total_drug_cost: FLOAT
total_drug_cost_ge65: FLOAT

Edge: has_review
Source: Drug
Target: Review

Analysis

__Disclaimer__
- Following reports have been generated from research datasets (links specified in Reference section) and are only for demonstrative purposes.

Exploratory Analysis: Find which diseases use most number of drugs

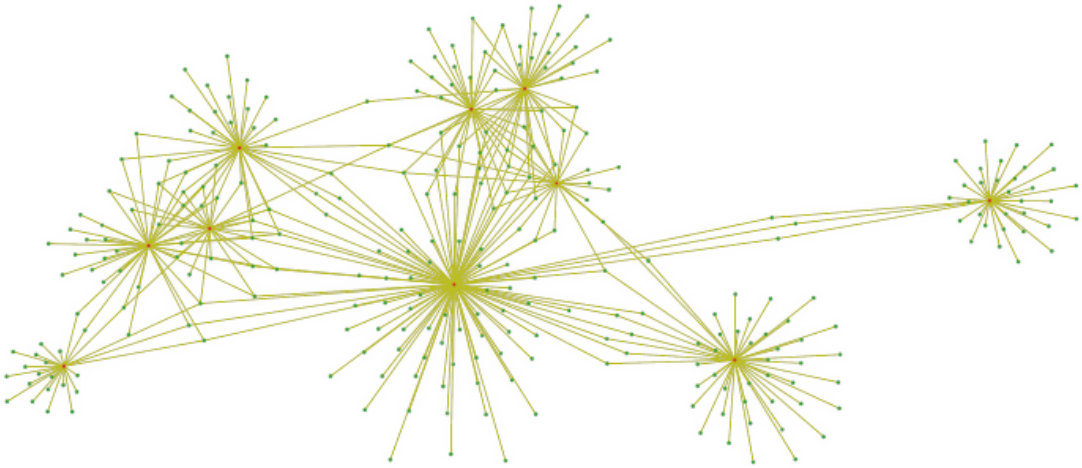
Query

```
CREATE QUERY DiseaseByDrugCount(INT topK) FOR GRAPH PrescribedDrugReview {
  TYPEDEF tuple<Vertex<Disease> disease, INT treated_by_drug_count> TreatedByDrugCount;

  SetAccum<EDGE> @@drug_disease_edges;
  HeapAccum<TreatedByDrugCount>(topK, treated_by_drug_count DESC) @@topk_diseases_accum;

  topk_diseases = SELECT ds FROM Disease:ds
  POST-ACCUM @@topk_diseases_accum += TreatedByDrugCount(ds, ds.outdegree("used_for"))
  ORDER BY ds.outdegree("used_for") DESC
  LIMIT 10
  ;

  drugs = SELECT d FROM topk_diseases:ds -(used_for:e)- Drug:d
```

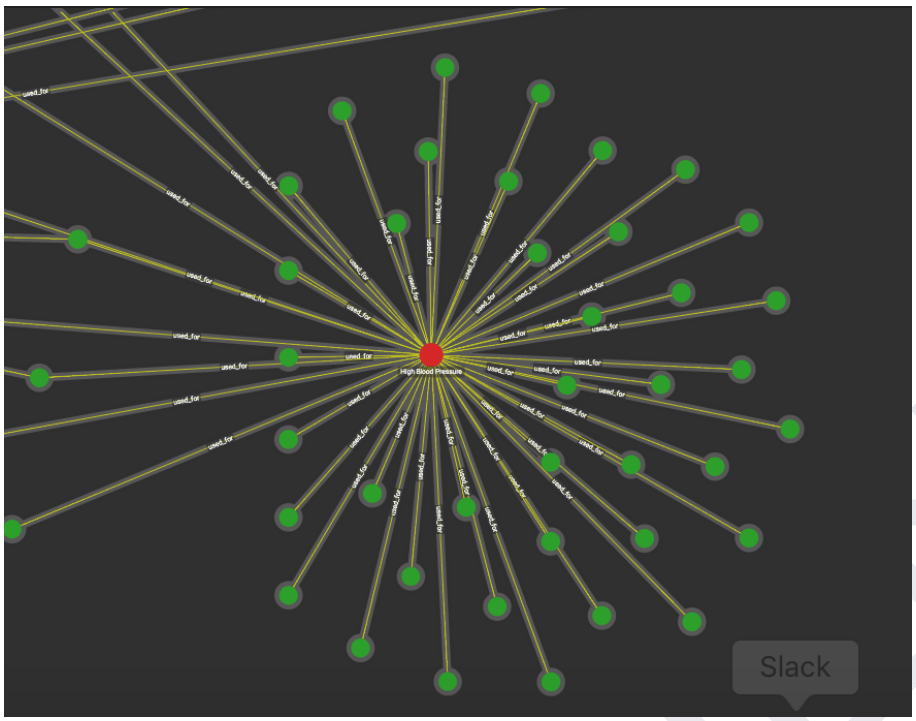
GRADIENT

```
  ACCUM @@drug_disease_edges += e
  ;

  PRINT @@topk_diseases_accum;
  PRINT topk_diseases, @@drug_disease_edges, drugs;
}
```

Graph



GRADIENT



Observations

Disease / Medical Condition	Drug Count
High Blood Pressure	48
Rheumatoid Arthritis	43
Pain	39
Depression	38
Diabetes, Type 2	35
Bipolar Disorde	33
Osteoarthritis	25

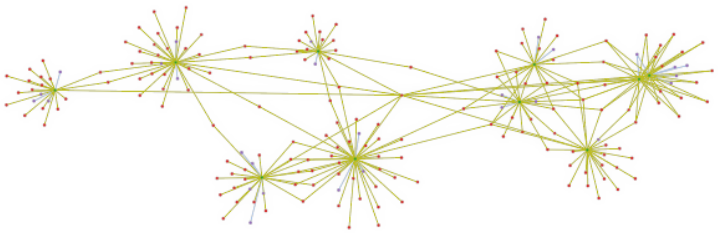
Exploratory Analysis: Find which drugs are used to treat most number of diseases

Query

```
CREATE QUERY DrugByDiseaseCount(INT topK) FOR GRAPH PrescribedDrugReview {
  TYPEDEF tuple<Vertex<Drug> drug, INT diseases_treated_count> DiseasesTreatedCount;

  SetAccum<EDGE> @@drug_disease_edges;
  SumAccum<INT> @diseases_treated_count;

  topk_drugs = SELECT d FROM Drug:d
  POST-ACCUM d.@diseases_treated_count += d.outdegree("used_for")
  ORDER BY d.outdegree("used_for") DESC
  LIMIT 10
  ;

  topk_brand_names = SELECT b FROM topk_drugs:d -(has_name:e)- BrandName:b
  ACCUM @@drug_disease_edges += e;

  diseases = SELECT ds FROM topk_drugs:d -(used_for:e)- Disease:ds
  ACCUM @@drug_disease_edges += e
  ;

  PRINT topk_drugs, @@drug_disease_edges, diseases, topk_brand_names;
}
```

GRADIENT

Graph



Observations

@diseases_treated_count	generic_name
39	Prednisone
31	Gabapentin
25	Ciprofloxacin
23	Metronidazole
23	Metronidazole/Sodium Chloride
22	Dexamethasone
20	Azithromycin
19	Clarithromycin
18	Risperidone
18	Ibuprofen

GRADIENT

Exploratory Analysis: Find the Drugs Most prescribed by Prescribers

```
CREATE QUERY DrugsByPrescriberCount(INT topK, INT botK) FOR GRAPH
PrescribedDrugReview {
  TYPEDEF tuple<Vertex<Drug> drug, INT prescriber_count> DrugPrescriberCount;
  SumAccum<INT> @prescribed_count;

  HeapAccum<DrugPrescriberCount>(topK, prescriber_count DESC)
@@most_prescribed_drugs;
  HeapAccum<DrugPrescriberCount>(botK, prescriber_count ASC)
@@least_prescribed_drugs;

  drugs = SELECT d FROM Drug:d
          POST-ACCUM
            @@most_prescribed_drugs += DrugPrescriberCount(d, d.outdegree("prescribed")),
            @@least_prescribed_drugs += DrugPrescriberCount(d, d.outdegree("prescribed"))
;

  PRINT @@most_prescribed_drugs, @@least_prescribed_drugs;
}
```

GRADIENT

Observations

drug	prescriber_count
Gabapentin	319849
Lisinopril	298762
Prednisone	275103
Omeprazole	269618
Furosemide	267731
Metoprolol Tartrate	234458
Hydrochlorothiazide	234005
Simvastatin	220868
Carvedilol	205977
Clopidogrel Bisulfate	198404

Advanced Analysis: Find the doctors prescribing low rated drugs for different diseases

Query

```
CREATE QUERY LowRatedPrescribers(VERTEX<Prescriber> prescriber, FLOAT threshold)
FOR GRAPH PrescribedDrugReview {
  BagAccum<FLOAT> @ratings_by_disease;
  SumAccum<FLOAT> @cum_avg_drug_rating;
  SetAccum<EDGE> @@allEdges;

  prescriber_set = {prescriber};
  start_drugs = SELECT d FROM prescriber_set:ps -(prescribed:e)- Drug:d;

  drugs = SELECT d FROM start_drugs:d -(used_for:e)- Disease:ds
  ACCUM d.@ratings_by_disease += LowRatedPrescribersSubquery(d, ds)
  POST-ACCUM d.@cum_avg_drug_rating += avg(d.@ratings_by_disease)
  HAVING d.@cum_avg_drug_rating <= threshold;
```

GRADIENT

```
 prescribers = SELECT ps FROM prescriber_set:ps -(prescribed:e)- drugs:d ACCUM
@@allEdges += e;

 diseases = SELECT ds FROM drugs:d -(used_for:e)- Disease:ds ACCUM @@allEdges += e;
 PRINT prescriber_set, drugs, diseases, @@allEdges;
}
```

Graph

Observation

This Graph shows that a Prescriber prescribed a Drug that has a low rating and it also shows the diseases that this drug is used for.

GRADIENT

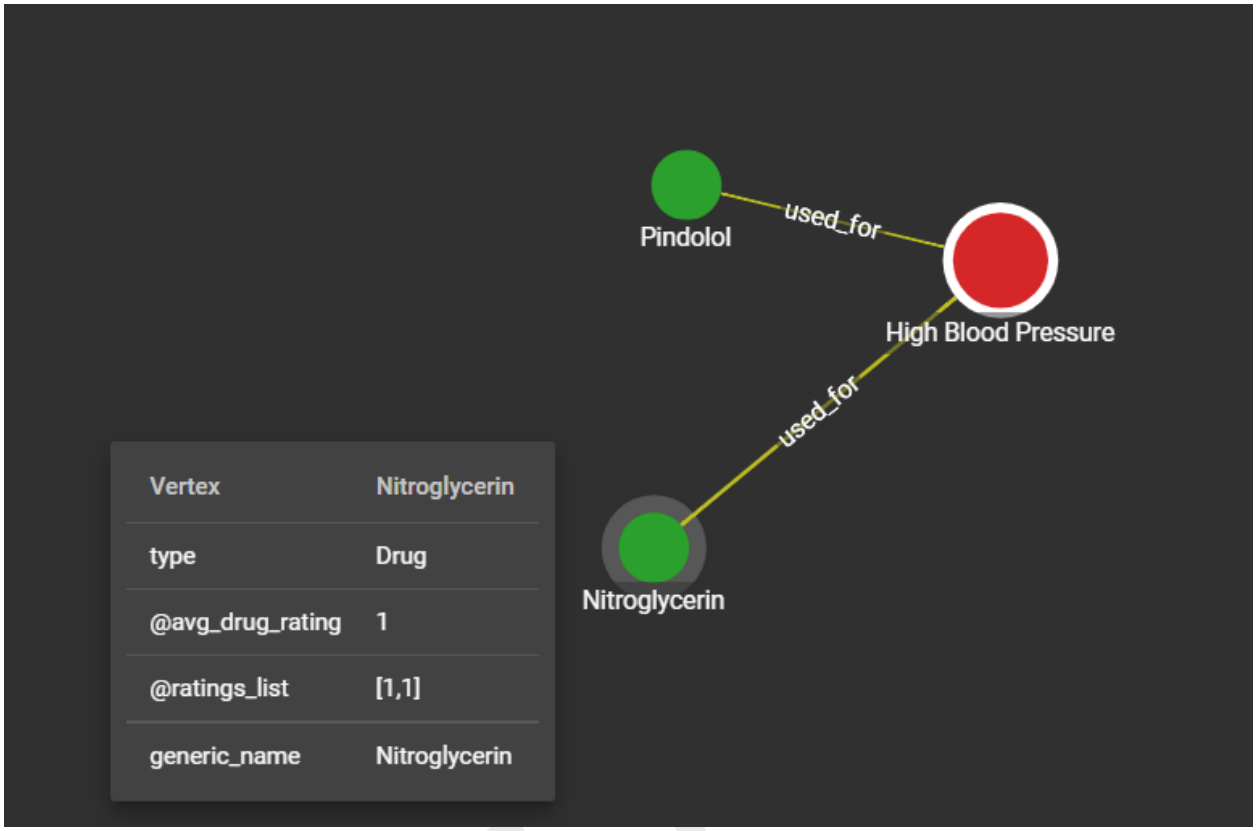
Advanced Analysis: Find the diseases being treated with low rated drugs

Query

```
CREATE QUERY DrugReviewForDisease(VERTEX<Disease> disease, INT threshold) FOR
GRAPH PrescribedDrugReview {
  # Assign drug ratings per drug

  BagAccum<INT> @ratings_list;
  SumAccum<FLOAT> @avg_drug_rating;
  SetAccum<EDGE> @@disease_drug_edges;

  disease_set = {disease};

  drugs = SELECT d FROM Drug:d -(has_review:e1)- Review:r -(associated_with:e2)-
disease_set:ds
  PER (d, e1, r, e2, ds)
  ACCUM d.@ratings_list += r.rating
  POST-ACCUM d.@avg_drug_rating += avg(d.@ratings_list)
  HAVING d.@avg_drug_rating >= threshold;

  tmp = SELECT d FROM drugs:d -(used_for:e)- disease_set:ds
  ACCUM @@disease_drug_edges += e;

  PRINT disease_set, @@disease_drug_edges, drugs;
}
```

Graph

GRADIENT



Observations

This Graph helped discover that a very low-rated drug was used for treating high blood pressure. This presents an opportunity to check if drugs with adverse effects or bannded drugs being used for disease treatment.

Future Work

- Overall we have established a solid foundation for developing interesting features and discovering many anomalies which can improve healthcare and save lives.
- Cluster the Prescriber and providers based on drugs prescribed.
- We have already identified prescribers recommending low rated drugs which are also very costly for example related to HIV Treatments.
- We would like to create a cluster of such prescriber-drug-disease groups.
- There is a great opportunity to find if banned drugs and drugs with adverse effects are being used.
- Check if any low rated drugs are being prescribed to age group > 65

GRADIENT

- Find similar prescribers based on the drugs prescription.
- Find social sentiment and usefulness of drugs by mining the review dataset (using NLP) and then enrich the Graph.
- Find Adverse Drug Events and Key Terms from the Review Notes and link them to the Drug Node in Graph.
- We want to capture time as a node in our graph in order to show how the price and rating of drugs change over time.
- We want to leverage Data Science Algorithms like https://docs.tigergraph.com/graph-ml/current/classification-algorithms/k-nearest-neighbors and https://docs.tigergraph.com/graph-ml/current/similarity-algorithms/jaccard-similarity-of-neighborhoods-batch

References

https://docs.tigergraph.com/home/