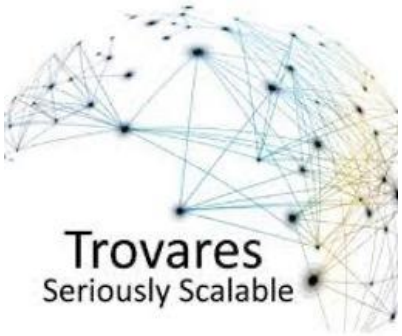
Unusual Trading Activity Detection

Team TV1 - Eric Zhang, Shruti Agarwal, Jack Wang

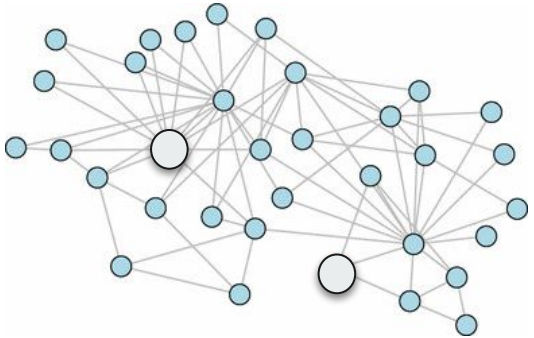
Introduction

Topic and Scope



A high-scalable graph analytics tool.

Topic and Original Scope



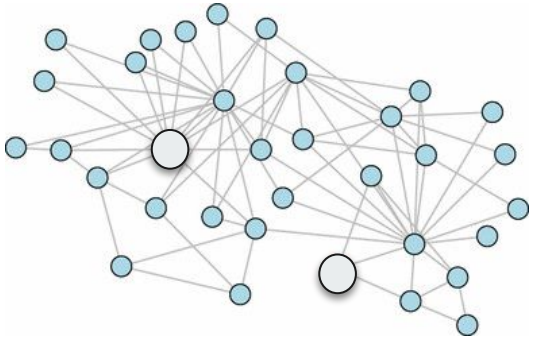
Topic: Unusual trading activities detection using financial trading datasets

Initial scope: Training graph embedding using GNN model, then perform link prediction on new data

Ideal dataset to learn a graph embedding should have the following features:

- **Node:** buyer and seller entity information (with unique entity id)
- **Edge:** trade information(eg. trade amount, location, time, etc.)
- **Other**: derived features(eg. avg trade amount, etc.)

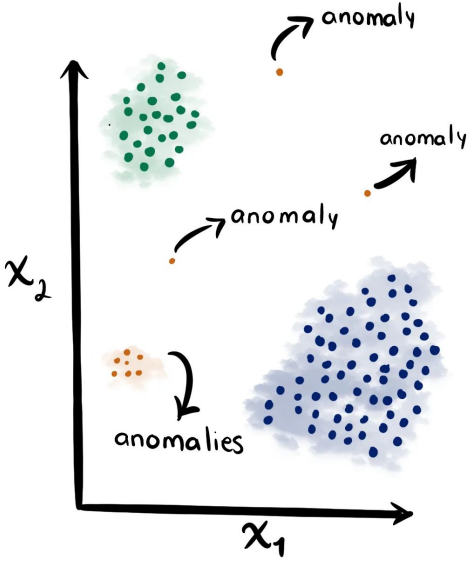
Shift in Scope



Graph Machine Learning

Difficulty

user-level financial datasets are highly private and not public accessible



Unsupervised Learning

Data

Millisecond Consolidated Trades

Millisecond Consolidated Trades contains transactions data trades for all securities listed on the New York Stock Exchange (NYSE), American Stock Exchange (AMEX), the Nasdaq National Market System (NMS), and all other U.S. equity exchanges

Data Profile

Sample of Apple Trading data(First 2 millions rows):

	DATE	TIME_M	EX	SYM_ROOT	SYM_SUFFIX	TR_SCOND	SIZE	PRICE	TR_STOP_IND	TR_CORR	TR_SEQNUM	TR_ID	TR_SOURCE
0	2022-12-01	4:00:00.008549370	K	AAPL	NaN	@ T	150	148.0800	NaN	0	1973	1	N
1	2022-12-01	4:00:00.015144352	K	AAPL	NaN	@ T	100	148.0800	NaN	0	1979	2	N
2	2022-12-01	4:00:00.015294500	K	AAPL	NaN	@ TI	20	148.0800	NaN	0	1980	3	N
3	2022-12-01	4:00:00.016654334	K	AAPL	NaN	@ TI	1	148.0800	NaN	0	1983	4	N
4	2022-12-01	4:00:00.016756830	K	AAPL	NaN	@ TI	50	148.0800	NaN	0	1984	5	N
...	...	...	...	...	...	...	...	...	...	...	...	...	...
1999995	2022-12-06	14:49:42.083800910	D	AAPL	NaN	@	100	142.4132	NaN	0	4581716	21717	N
1999996	2022-12-06	14:49:42.183898524	D	AAPL	NaN	@ I	1	142.4200	NaN	0	4581718	158343	N
1999997	2022-12-06	14:49:42.546222554	K	AAPL	NaN	@F	300	142.4100	NaN	0	4581756	34021	N
1999998	2022-12-06	14:49:42.546240497	K	AAPL	NaN	@F	300	142.4100	NaN	0	4581757	34022	N
1999999	2022-12-06	14:49:42.546245080	K	AAPL	NaN	@F	100	142.4100	NaN	0	4581758	34023	N

2000000 rows × 14 columns

Variables Description:

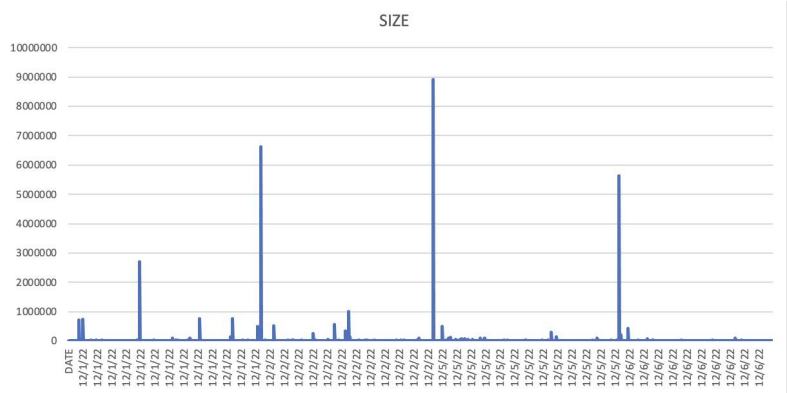
Variable Name	Type	Description
DATE	date	Date of trade (DATE)
TIME_M	double	Time of Trade or Quote with milliseconds (HHMMSSXXX) (TIME_M)
SYM_ROOT	string	Security symbol root (SYM_ROOT)
SYM_SUFFIX	string	Security symbol suffix (SYM_SUFFIX)
EX	string	Exchange that issued the trade (EX)
TR_SCOND	string	Trade Sale Condition (up to 4 codes) (TR_SCOND)
SIZE	double	Volume of trade (SIZE)
PRICE	double	Price of trade (PRICE)
TR_STOPIND	string	Trade Stop Stock Indicator (valid 2003-2015) (NYSE Only) (TR_STOPIND)
TR_CORR	string	Trade Correction Indicator (TR_CORR)
TR_SEQNUM	double	Trade Sequence Number (TR_SEQNUM)
TR_SOURCE	string	Source of Trade (TR_SOURCE)
TR_RF	string	Trade Reporting Facility (TR_RF)
TR_ID	string	Trade ID (valid Oct 2016 - present) (TR_ID)
TR_STOP_IND	string	Trade Stop Stock Indicator (valid 2016-present) (NYSE Only) (TR_STOP_IND)

Data Preprocessing:

Second-level:

	DATE	TIME_S	HOUR	EX	SYM_ROOT	TR_SCOND	TR_CORR	TR_SOURCE	PRICE	SIZE
0	2022-12-01	04:00:00	4	K	AAPL	@ T	0	N	148.080000	250
1	2022-12-01	04:00:00	4	K	AAPL	@ TI	0	N	147.951818	205
2	2022-12-01	04:00:00	4	K	AAPL	@FT	0	N	147.500000	100
3	2022-12-01	04:00:00	4	K	AAPL	@FTI	0	N	147.518000	144
4	2022-12-01	04:00:00	4	P	AAPL	@ T	0	N	147.426667	524
...	...	...	...	...	...	...	...	...	...	...
705663	2022-12-06	14:49:41	14	Q	AAPL	@ I	0	N	142.420000	69
705664	2022-12-06	14:49:41	14	Z	AAPL	@	0	N	142.420000	100
705665	2022-12-06	14:49:42	14	D	AAPL	@	0	N	142.413200	100
705666	2022-12-06	14:49:42	14	D	AAPL	@ I	0	N	142.420000	1
705667	2022-12-06	14:49:42	14	K	AAPL	@F	0	N	142.410000	700

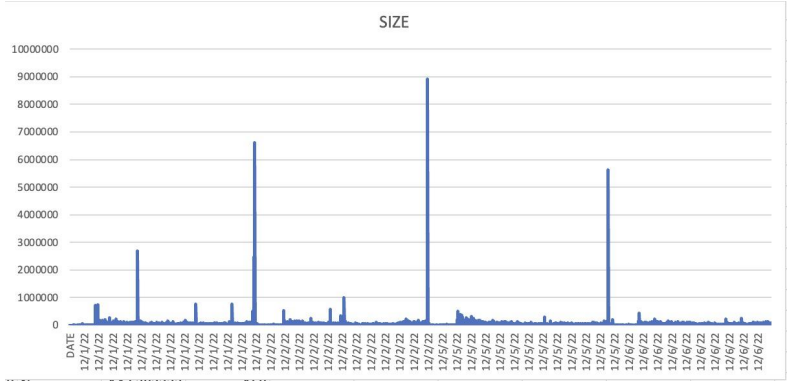
705668 rows × 10 columns



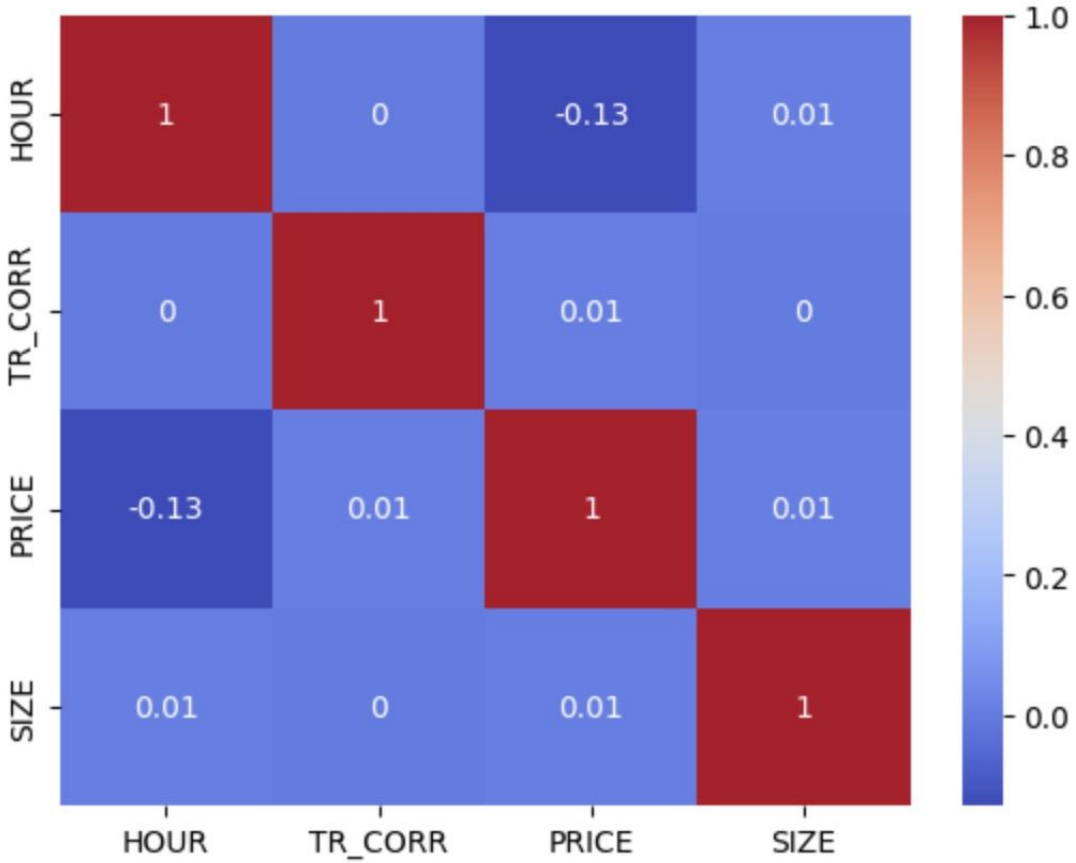
Minute-level:

	DATE	TIME_M	HOUR	EX	SYM_ROOT	TR_SCOND	TR_CORR	TR_SOURCE	PRICE	SIZE
0	2022-12-01	04:00	4	K	AAPL	@ T	0	N	148.080000	250
1	2022-12-01	04:00	4	K	AAPL	@ TI	0	N	147.855000	306
2	2022-12-01	04:00	4	K	AAPL	@FT	0	N	147.563333	1194
3	2022-12-01	04:00	4	K	AAPL	@FTI	0	N	147.643846	409
4	2022-12-01	04:00	4	P	AAPL	@ T	0	N	147.426667	524
...	...	...	...	...	...	...	...	...	...	...
80807	2022-12-06	14:49	14	X	AAPL	@F	0	N	142.370000	200
80808	2022-12-06	14:49	14	Y	AAPL	@	0	N	142.361000	100
80809	2022-12-06	14:49	14	Z	AAPL	@	0	N	142.387778	1000
80810	2022-12-06	14:49	14	Z	AAPL	@ I	0	N	142.391944	290
80811	2022-12-06	14:49	14	Z	AAPL	@F	0	N	142.340000	100

80812 rows × 10 columns



Correlation Heat Map:



Models

Unsupervised ML Models

We deployed the following models on the dataset:

1. One-Class SVM
2. Isolation Forest
3. Autoencoder

One-Class SVM

This model is a type of support vector machine that is used for anomaly detection.

It works by fitting a model to the normal data and then detecting any points that fall outside of this normal range.

Benefits:

- Unsupervised Anomaly Detection: One of the main benefits of one-class SVM is that it is able to detect anomalies in an unsupervised manner, without requiring labeled data. This makes it useful in scenarios where labeled data is not available or when the cost of labeling data is high.
- Robust to Noise: One-class SVM can deal with noisy data effectively, as it focuses on learning the underlying structure of the data rather than trying to fit the noisy points.

One-Class SVM

	DATE	TIME_M	HOUR	EX	SYM_ROOT	TR_SCOND	TR_CORR	TR_SOURCE	PRICE	SIZE	anomaly	anomaly_scores
7408	2022-12-01	10:54	10	D	AAPL	@	0	N	147.796367	104790	-1	-4.768916e-04
65873	2022-12-06	09:36	9	Z	AAPL	@	0	N	146.258846	10043	-1	-4.768339e-04
70037	2022-12-06	11:00	11	Q	AAPL	@	0	N	144.116308	16834	-1	-4.767457e-04
74908	2022-12-06	12:46	12	D	AAPL	@	0	N	144.055591	41826	-1	-4.767365e-04
7601	2022-12-01	10:58	10	D	AAPL	@	0	N	147.981436	134828	-1	-4.764295e-04
...	...	...	...	...	...	...	...	...	...	...	...	...
22331	2022-12-02	04:08	4	Q	AAPL	@FT	0	N	148.150000	150	-1	-3.206624e-07
17216	2022-12-01	14:36	14	D	AAPL	@	0	N	148.538423	36231	-1	-2.685946e-07
46523	2022-12-05	10:04	10	Z	AAPL	@F I	0	N	149.790938	1012	-1	-1.937581e-07
46098	2022-12-05	09:57	9	A	AAPL	@F I	0	N	150.555000	179	-1	-9.998197e-08
79479	2022-12-06	14:22	14	B	AAPL	@F	0	N	142.650000	320	-1	-9.641993e-08

4763 rows × 12 columns

*The average unusual trades are around 5.89%

Isolation Forest

This model is a type of anomaly detection algorithm that can be used to identify outliers in a dataset.

It works by constructing random forests and then isolating the outliers in a different branch of the tree.

Benefits:

- Easily scalable to high dimensional and large datasets. (Like ours)
- Works when irrelevant features are included.

Isolation Forest

	DATE	TIME_M	HOUR	EX	SYM_ROOT	TR_SCOND	TR_CORR	TR_SOURCE	PRICE	SIZE	anomaly_scores	anomaly
0	2022-12-01	04:00	4	K	AAPL	@ T	0	N	148.080000	250	0.005919	1
1	2022-12-01	04:00	4	K	AAPL	@ TI	0	N	147.855000	306	0.018625	1
2	2022-12-01	04:00	4	K	AAPL	@FT	0	N	147.563333	1194	-0.007681	-1
3	2022-12-01	04:00	4	K	AAPL	@FTI	0	N	147.643846	409	0.002507	1
4	2022-12-01	04:00	4	P	AAPL	@ T	0	N	147.426667	524	0.012728	1

- Have both class(anomaly/normal) prediction outputs and anomaly scores
- Extremely fast
- Flexible

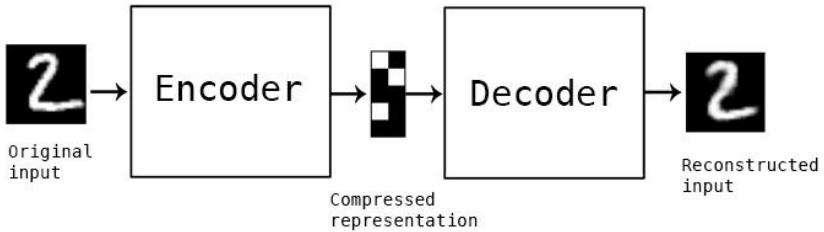
Autoencoder

This model is a type of neural network that can be used for unsupervised anomaly detection.

It works by learning the compressed representation of the input data and then reconstructing the input with this compressed representation.

Benefits:

- Dimensionality reduction.
- Anomaly detection
- Tackling unsupervised learning problems
- Image processing



Autoencoder

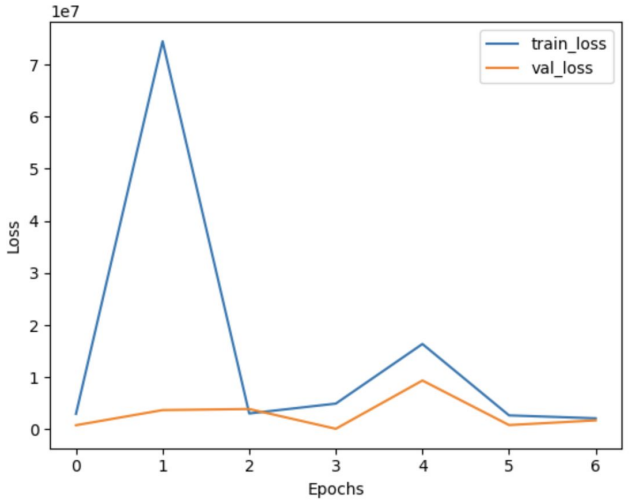
	HOUR	PRICE	SIZE	0	1	2	3	4	5	6	...	58	59	60	61	62	63	64	65	66	anomaly
48743	10	147.982524	181720	0.0	0.0	0.0	1.0	0.0	0.0	0.0	...	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	1.0	-1
46686	10	148.898230	171475	0.0	0.0	0.0	1.0	0.0	0.0	0.0	...	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	1.0	-1
3009	9	148.340000	720606	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	1.0	-1
45371	9	150.144805	175231	0.0	0.0	0.0	1.0	0.0	0.0	0.0	...	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	1.0	-1
5380	10	146.893229	176301	0.0	0.0	0.0	1.0	0.0	0.0	0.0	...	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	1.0	-1
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
37436	14	146.220000	100	0.0	0.0	0.0	1.0	0.0	0.0	0.0	...	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	1.0	1
37430	14	146.224000	11	1.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	1.0	1
37425	14	146.299545	761	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	1.0	1
37322	14	146.321429	1179	0.0	0.0	0.0	0.0	0.0	0.0	1.0	...	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	1.0	1
113892	10	143.010000	200	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	1.0	1

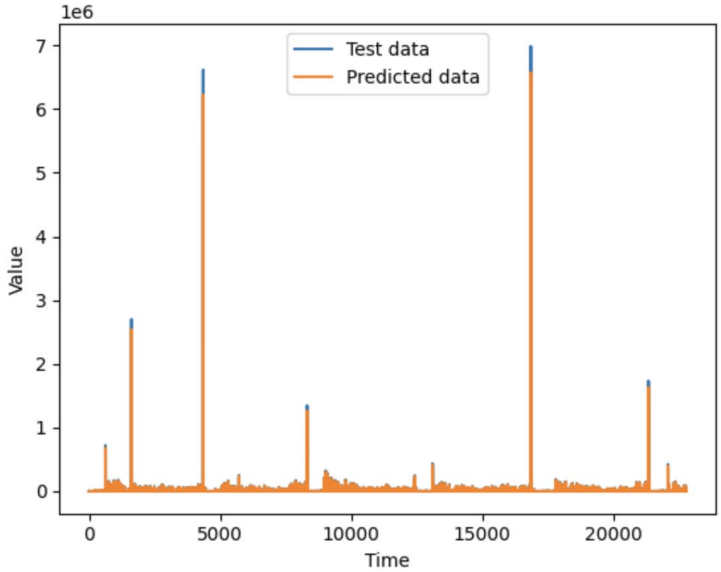
22779 rows × 71 columns

- Contains both fraud and genuine transactions.
- Classified 23 entries as fraud based on the classification parameter set to 0.1%

Autoencoder

The validation loss is close to the training loss, and that both losses decrease and plateau at a relatively low level.





The test and predicted data overlapping represent the degree of similarity or reconstruction accuracy between the original test data and the data generated by the autoencoder.

Use-case Analysis

Our company, Trovares wanted to develop an unsupervised fraud detection system to identify fraudulent activity in trading data. The data included the date, time, size, price, and the company name of the stock traded.

The use case analysis involved the following steps:

1. **Defining the business problem:** The company wants to identify fraudulent transactions in order to minimize losses due to fraudulent activity.
2. **Gathering data:** Millisecond trading data of Apple stocks.
3. **Defining the requirements:** The system should be able to identify fraudulent transactions based on patterns of activity that are unusual or anomalous, be able to handle large volumes of transaction data in real-time, and provide accurate results with minimal false positives.
4. **Developing the model:** The model would use features such as transaction price, size, date and time of day to identify unusual patterns of activity.
5. **Testing and validating the model:** The model would be tested and validated using historical transaction data, and the performance metrics would be evaluated to ensure accuracy and efficiency.
6. **Deploying the model:** Once the model is tested and validated, it would be deployed in a production environment where it can continuously monitor transactions for fraudulent activity.
7. **Evaluating and improving the model:** The company would regularly evaluate the performance of the model and identify opportunities for improvement. This could involve incorporating new data sources or refining the feature set to improve accuracy.

By implementing an unsupervised fraud detection system, the company can minimize losses due to fraudulent activity.

Applications

1. **Banking and Finance:** credit card fraud, money laundering, and identity theft.
2. **E-commerce:** fraudulent purchases and transactions
3. **Insurance:** fraudulent claims, such as fake accidents, staged thefts, or false injury claims.
4. **Healthcare**: fraudulent claims and billing practices.
5. **Government:** fraudulent activities, such as tax fraud or benefit fraud.

Limitations and Future Directions

Limitations:

Access to Data

Access to Computing Resources

Lack of Domain Knowledge

Separate Models

Future Directions:

Find data vendors if we got sponsored

Gain access cloud computing

Consult domain experts for insights

Ensemble Learning

Q&A