ORACLE

Art of Possible with AI & Data Science

Tax Fraud Detection powered by Generative AI, Graph Analytics and Machine Learning

EMEA Data Science Team

Tax Fraud Briefly

Income Tax Evasion

- Individuals and businesses underreporting their income or inflating deductions to reduce tax liabilities.
- Data Sources: Tax Returns, Wage and Salary Data

VAT/GST Fraud

- Fraudulent schemes involving the collection of value-added tax (VAT) or goods and services tax (GST) but not remitting it to the government.
- Data Sources: Transaction data, invoices, and sales records from businesses
- Taxpayer Records

Corporate Tax Fraud

- Large corporations engaging in aggressive tax planning or transfer pricing schemes to minimize their tax obligations.
- Data Sources: Corporate Financial Statements, Transfer Pricing Documentation

Employment Tax Fraud

- Employers misclassifying employees as independent contractors, thereby avoiding payroll taxes.
- Data Sources: Payroll data

Tax Fraud Briefly

Tax Shelter Abuse

- Exploiting offshore tax shelters or abusive tax shelters to hide income from taxation.
- Data sources: Financial institution data to reveal offshore accounts and financial holdings
- Legal documents

Identity Theft and Refund Fraud

- Criminals using stolen identities to file fraudulent tax returns and claim refunds.
- Data sources: Taxpayer Identification Data,

Sales Tax Fraud

- Retailers underreporting sales or manipulating sales records to evade sales tax.
- Data sources: POS data, inventory records, sales transactions

Cryptocurrency Tax Evasion

- Analyzing blockchain data to trace cryptocurrency transactions and identify tax evasion involving digital assets.
- Cryptocurrency blockchain data, Cryptocurrency Exchange Data

Real-Time Fraud Detection with Oracle Graph Technologies

"It's not only about fraud. We can analyze our payments network, our device agent networks, we can feed graph topology and machine learning algorithms. They help us uncover how much more we can do"

Stanka Dalekova

Chief Technical Lead, Paysafe

Paysafe Group

- Online Payments provider, headquartered in Canada
- Annualized transactional volume $100bn (2020)
- Processing up to 500000 payments per day

Business requirements

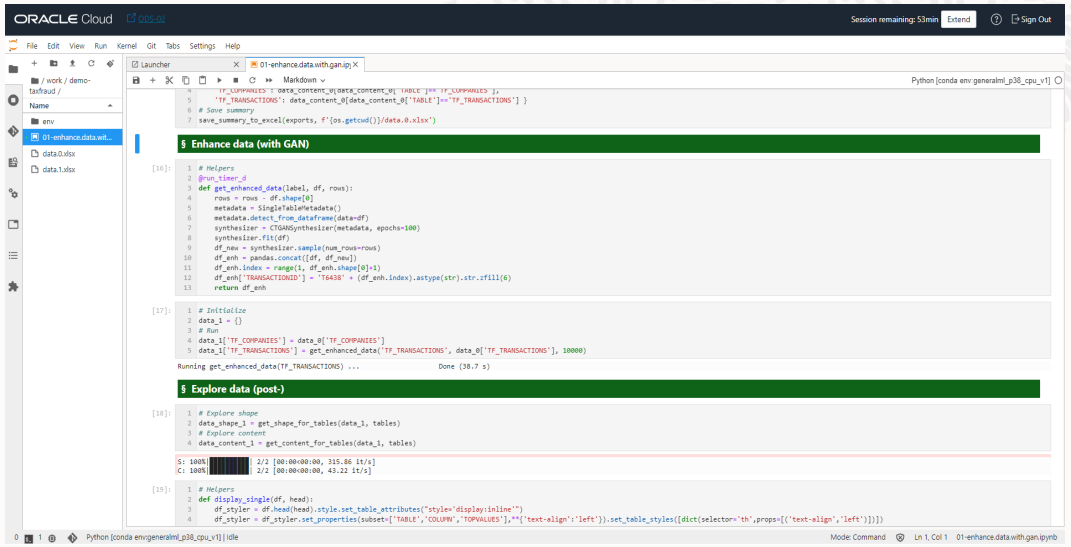
- Improve accuracy of fraud detection
- minimize chargebacks from Credit Card providers while not impacting customer experience
- Automate work of analysts in fraud department
- Integrate with operational payment systems (Oracle, MSFT)

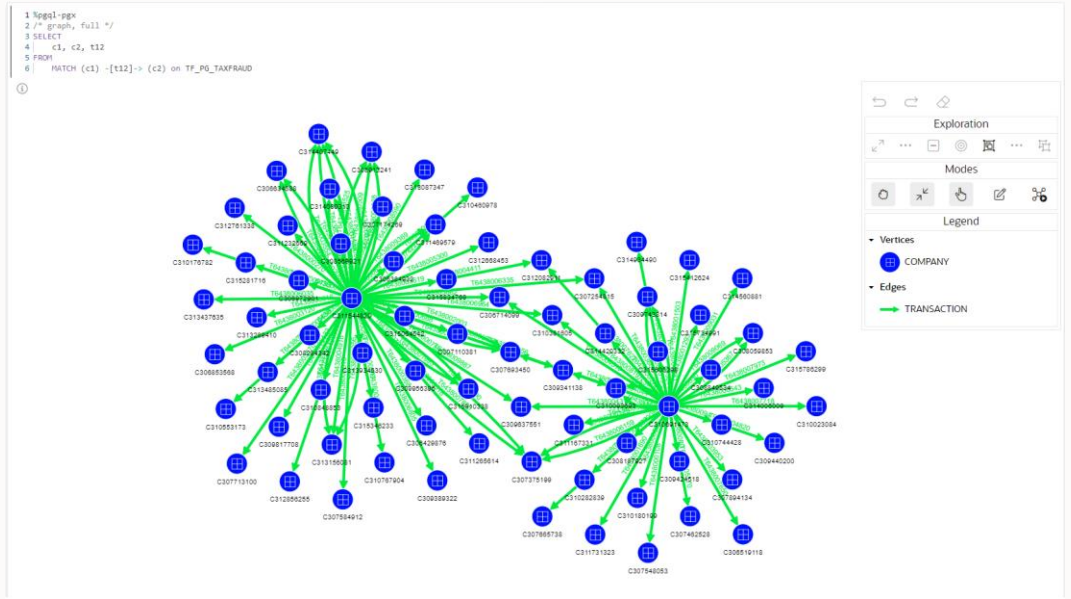
Fraud analytics platform based on Oracle Graph

- In-memory analytics engine with millisecond response time enables real-time transaction monitoring
- Integrated interactive visualization of transactions network eliminates manual work of fraud analysts
- Inclusion of account and device-related data in combination with ML improves prediction accuracy
- Seamless integration with payment systems through Oracle GoldenGate

Demo Flow

1. Demo Inspiration & Target Personas
2. How have we achieved this?
   - Business Understanding
   - Data Enhancement with Generative AI
   - Data Exploration
   - Data Preparation (for Graph)
   - Graph Model
   - Graph Metrics
   - Data Preparation (for Machine Learning)
   - Machine Learning Model
   - Business Insights
3. Behind the Scenes
   - Oracle Data Platform
4. Next Steps





Demo Inspiration & Target Personas

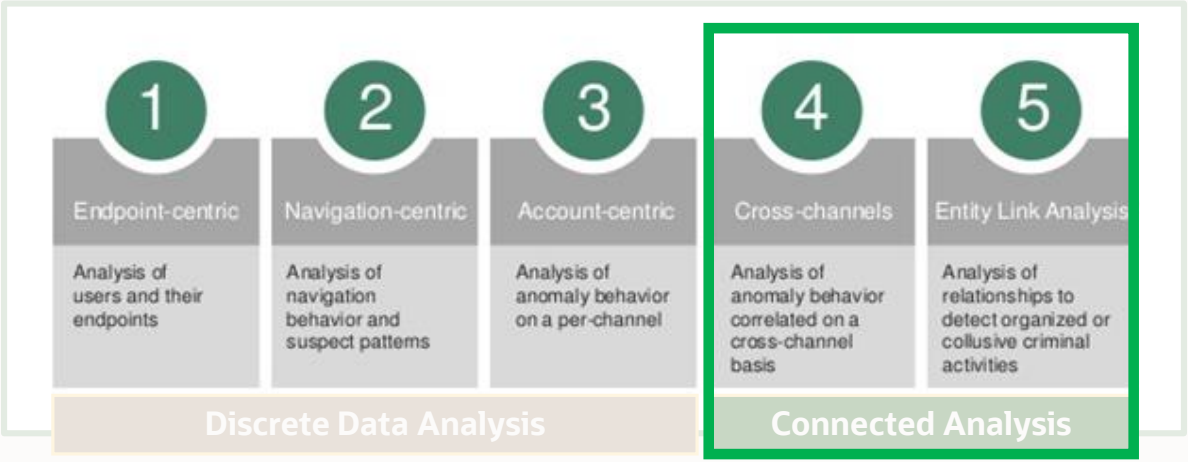
Demo Inspiration

This demo is going to showcase Oracle AI/ML platform capabilities to identify fraudulent behavior in companies through analysis of interactions.

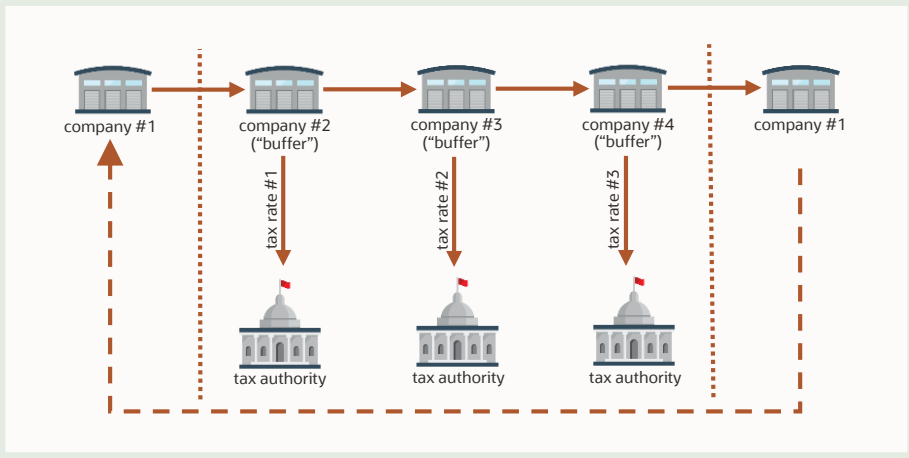
A particular type of behavior investigated includes circular money transfers, which can be indicative of abnormal interactions and can constitute sufficient cause for further analysis.

Objective is uncovering hidden fraud patterns through **graph analysis, machine learning** and **generative AI** techniques.

Gartner's 5 of Fraud Prevention / Detection



Circular money transfers



Demo Inspiration

With AI driven fraud detection model:

- Tax investigators will be able to identify companies with high fraud risks more effectively
- Tax authorities will better allocate investigator resources
- Government will promote a fair tax system and minimizing revenue losses

Tax fraud poses a significant challenge globally, with estimates suggesting that it costs governments billions of dollars in lost revenue each year. It not only affects public finances but also erodes trust in the tax system.

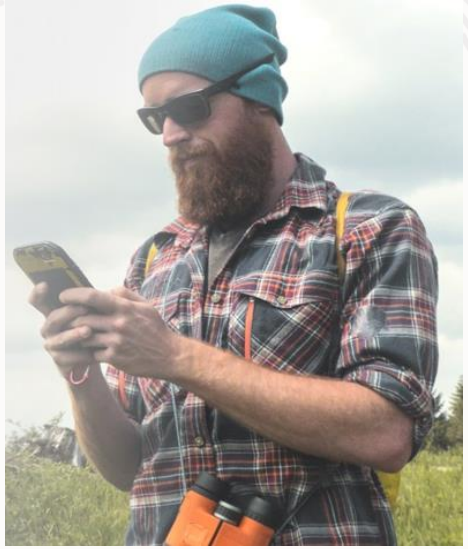
*HMRC (UK tax collecting authority) estimates that losses to tax fraud amount to **£16 billion** each year. This is nearly half of HMRC's estimate of the tax gap (£34 billion): the difference between the amount of tax HMRC should collect each year and the amount it actually collects.**

**https://www.nao.org.uk/reports/tackling-tax-fraud-how-hmrc-responds-to-tax-evasion-the-hidden-economy-and-criminal-attacks/*

Target Personas

Fraud Investigator wants to gather evidence to identify tax fraud investigating suspicious activities

Data Scientist wants to use Generative AI, machine learning and graph analytics to develop models and algorithms that can detect fraud patterns in large datasets.



Business Analyst wants to build final dashboards to create visually informative and interactive dashboards that enable fraud investigator to monitor and analyze suspicious activities

How have we achieved this?

Business Understanding

Fraud Investigator wants to understand any cyclic money transfers among companies, as well as importance of said companies.

Cyclic money transactions can be a potential signal of tax fraud because they often indicate an attempt to hide the true nature of financial transactions and income. A measure of importance would also be very useful for prioritization.

Transactions
Transaction ID
Company From ID
Company To ID
Transaction Date
Transaction Amount
...

Companies
Company ID
Company Name
...

Other Potential Data Sources
Geolocation
VAT forms/tax returns
Whistleblower Reports
Customs and Import/Export Data
Social Media data
3rd party data

tax fraud data sources

Data Enhancement with Generative AI

Data Scientist assesses available data and sees that existing data volume is very low. She uses Generative AI to increase data volume that helps understanding fraud cases better.

Companies
Company ID
Company Name
...

Transactions
Transaction ID
Company From ID
Company To ID
Transaction Date
Transaction Amount
...

Transactions
Transaction ID
Company From ID
Company To ID
Transaction Date
Transaction Amount
...

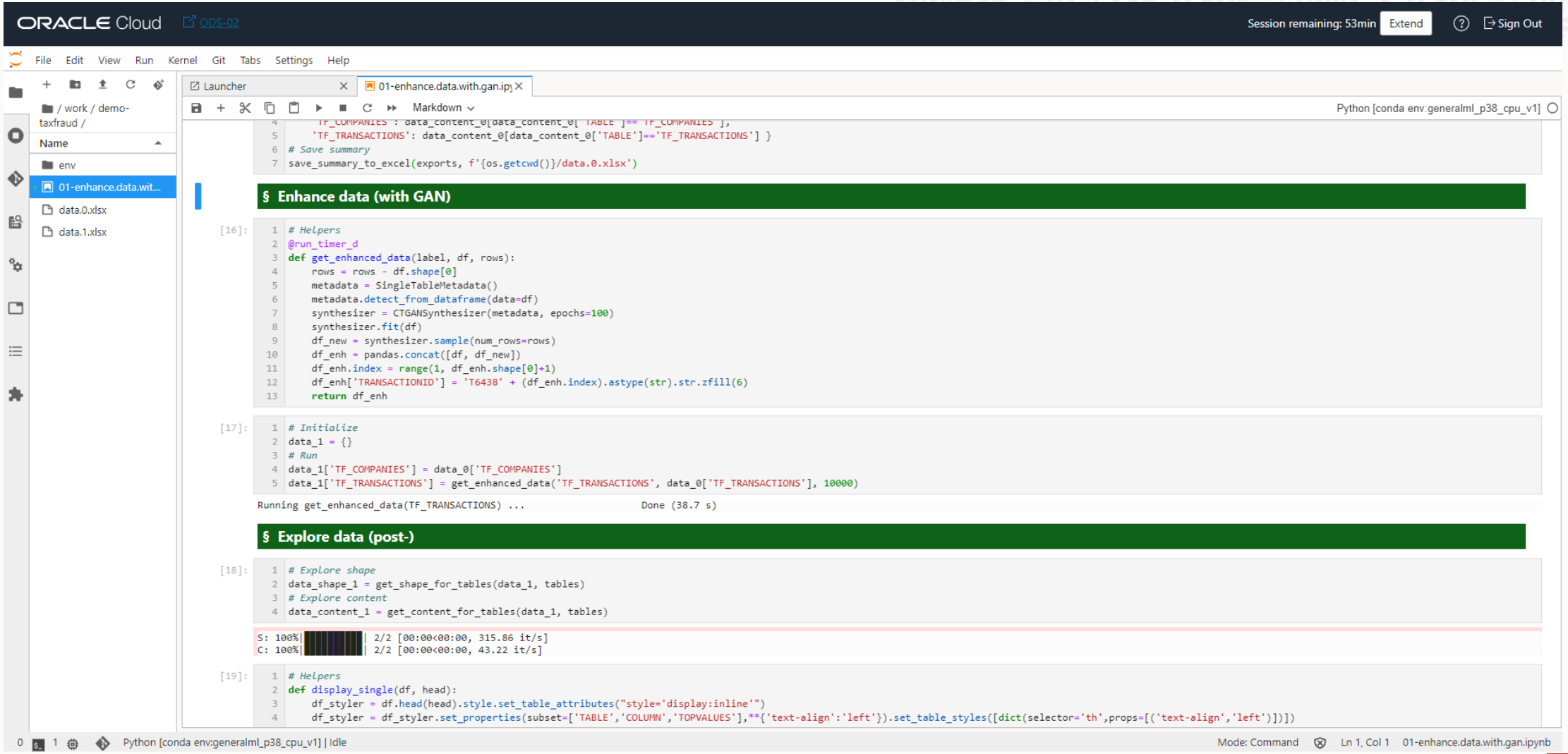
200 companies

1K transactions

10K transactions

synthetic data generation

Data Enhancement with Generative AI



Data Exploration

Data Scientist explores the data to have a good understanding before proceeding with more advanced analysis and modeling tasks.

Companies
Company ID
Company Name
...

Transactions
Transaction ID
Company From ID
Company To ID
Transaction Date
Transaction Amount
...

200 companies

10K transactions

Data Exploration

```
%python

z.show(TF_WIP_SHAPE)
```

FINISHED

settings

TABLE	ROWS	COLUMNS
TF_COMPANIES	200	2
TF_TRANSACTIONS	10000	5

```
%python

z.show(TF_WIP_CONTENT)
```

FINISHED

settings

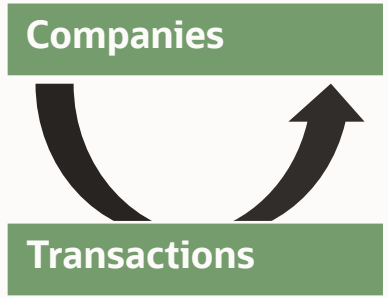
TABLE	COLUMN	VALUES	NULLS	NOTNULLS	DISTINCT	TOPVALUES
TF_COMPANIES	COMPANYID	200	0	200	200	C310023084 : 1 (0.5%), C307110381 : 1 (0.5%), C306592341 : 1 (0.5%), C306634688 : 1 (0.5%), C306714099 : 1 (0.5%),
TF_COMPANIES	COMPANYNAME	200	0	200	200	Company #0079 : 1 (0.5%), Company #0016 : 1 (0.5%), Company #0006 : 1 (0.5%), Company #0007 : 1 (0.5%), Company #0008 : 1 (0.5%),
TF_TRANSACTIONS	TRANSACTIONID	10000	0	10000	10000	T6438000001 : 1 (0.0%), T6438006671 : 1 (0.0%), T6438006664 : 1 (0.0%), T6438006665 : 1 (0.0%), T6438006666 : 1 (0.0%),
TF_TRANSACTIONS	TRANSACTIONDATE	10000	0	10000	31	13.04.2017 : 390 (3.9%), 12.04.2017 : 388 (3.9%), 23.04.2017 : 384 (3.8%), 07.04.2017 : 372 (3.7%), 14.04.2017 : 363 (3.6%),
TF_TRANSACTIONS	COMPANYFROMID	10000	0	10000	200	C315346233 : 143 (1.4%), C309440200 : 138 (1.4%), C307110381 : 124 (1.2%), C309113624 : 116 (1.2%), C306429876 : 105 (1.1%),
TF_TRANSACTIONS	COMPANYTOID	10000	0	10000	200	C313789101 : 104 (1.0%), C313862631 : 100 (1.0%), C315734991 : 98 (1.0%), C310308434 : 97 (1.0%), C315664164 : 96 (1.0%),
TF_TRANSACTIONS	TRANSACTIONAMOUNT	10000	0	10000	6465	49 : 3158 (31.6%), 2948 : 5 (0.1%), 10335 : 4 (0.0%), 882 : 3 (0.0%), 622 : 3 (0.0%),

Data Preparation (for Graph)

Data Scientist transforms tabular data into a format suitable for representing companies as nodes and transaction amount as edges in a graph.

Companies
Company ID
Company Name
...

Transactions
Transaction ID
Company From ID
Company To ID
Transaction Date
Transaction Amount
...



200 companies

10K transactions

200 nodes
10K edges

Data Preparation (for Graph)

```python
%python

# imports
import warnings
import oml
import pandas

# settings
warnings.filterwarnings('ignore')
pandas.set_option('display.width', 2000)
```

FINISHED

```python
%python

# sync proxy objects
TF_COMPANIES = oml.sync(table="TF_COMPANIES")
TF_TRANSACTIONS = oml.sync(table="TF_TRANSACTIONS")

# pull data
data = {}
data['TF_COMPANIES'] = TF_COMPANIES.pull()
data['TF_TRANSACTIONS'] = TF_TRANSACTIONS.pull()
```

FINISHED

```python
%python

# init
data_pg = {}

# transform data 'COMPANIES'
data_pg['TF_PGV_COMPANIES'] = data['TF_COMPANIES']

# transform data 'TRANSACTIONS'
data_pg['TF_PGE_TRANSACTIONS'] = data['TF_TRANSACTIONS']

# check
print(f'COMPANIES data now has {data_pg["TF_PGV_COMPANIES"].shape[0]} rows and {data_pg["TF_PGV_COMPANIES"].shape[1]} columns.')
print(f'TRANSACTIONS data now has {data_pg["TF_PGE_TRANSACTIONS"].shape[0]} rows and {data_pg["TF_PGE_TRANSACTIONS"].shape[1]} columns.')
```

```
COMPANIES data now has 200 rows and 2 columns.
TRANSACTIONS data now has 10000 rows and 5 columns.
```

FINISHED

```python
%python

# init
data_s = data_pg

# push data
try:
    oml.drop(table='TF_PGV_COMPANIES')
    oml.drop(table='TF_PGE_TRANSACTIONS')
except:
    pass
TF_PGV_COMPANIES = oml.create(data_s['TF_PGV_COMPANIES'], table='TF_PGV_COMPANIES',
    dbtypes={'COMPANYID':'VARCHAR2(100)','COMPANYNAME':'VARCHAR2(100)'})
TF_PGE_TRANSACTIONS = oml.create(data_s['TF_PGE_TRANSACTIONS'], table='TF_PGE_TRANSACTIONS',
    dbtypes={'TRANSACTIONID':'VARCHAR2(100)','TRANSACTIONDATE':'VARCHAR2(100)','COMPANYFROMID':'VARCHAR2(100)','COMPANYTOID':'VARCHAR2(100)','TRANSACTIONAMOUNT':'NUMBER(10)'})
```

FINISHED

```
%script

alter table TF_PGE_TRANSACTIONS drop constraint FK_TF_PGE_TRANSACTIONS_FROM;
alter table TF_PGE_TRANSACTIONS drop constraint FK_TF_PGE_TRANSACTIONS_TO;
alter table TF_PGE_TRANSACTIONS drop constraint PK_TF_PGE_TRANSACTIONS;
alter table TF_PGV_COMPANIES drop constraint PK_TF_PGV_COMPANIES
```

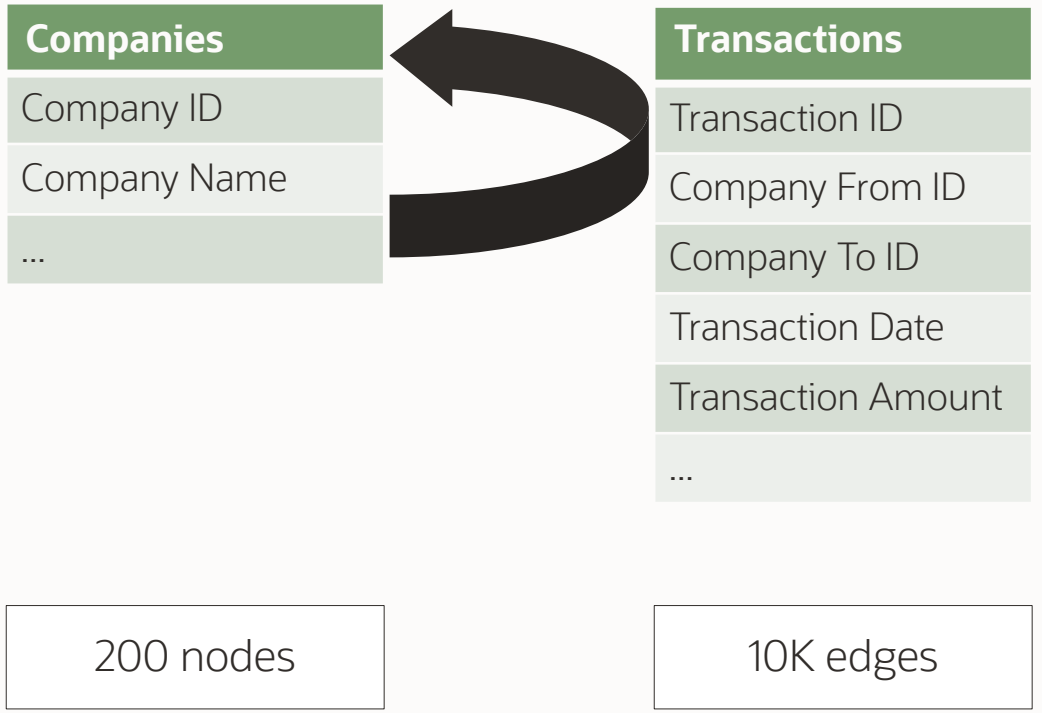
FINISHED

```
%script

alter table TF_PGV_COMPANIES add constraint PK_TF_PGV_COMPANIES PRIMARY KEY (COMPANYID);
alter table TF_PGE_TRANSACTIONS add constraint PK_TF_PGE_TRANSACTIONS PRIMARY KEY (TRANSACTIONID);
alter table TF_PGE_TRANSACTIONS add constraint FK_TF_PGE_TRANSACTIONS_FROM FOREIGN KEY (COMPANYFROMID) references TF_PGV_COMPANIES(COMPANYID);
alter table TF_PGE_TRANSACTIONS add constraint FK_TF_PGE_TRANSACTIONS_TO FOREIGN KEY (COMPANYTOID) references TF_PGV_COMPANIES(COMPANYID)
```

FINISHED

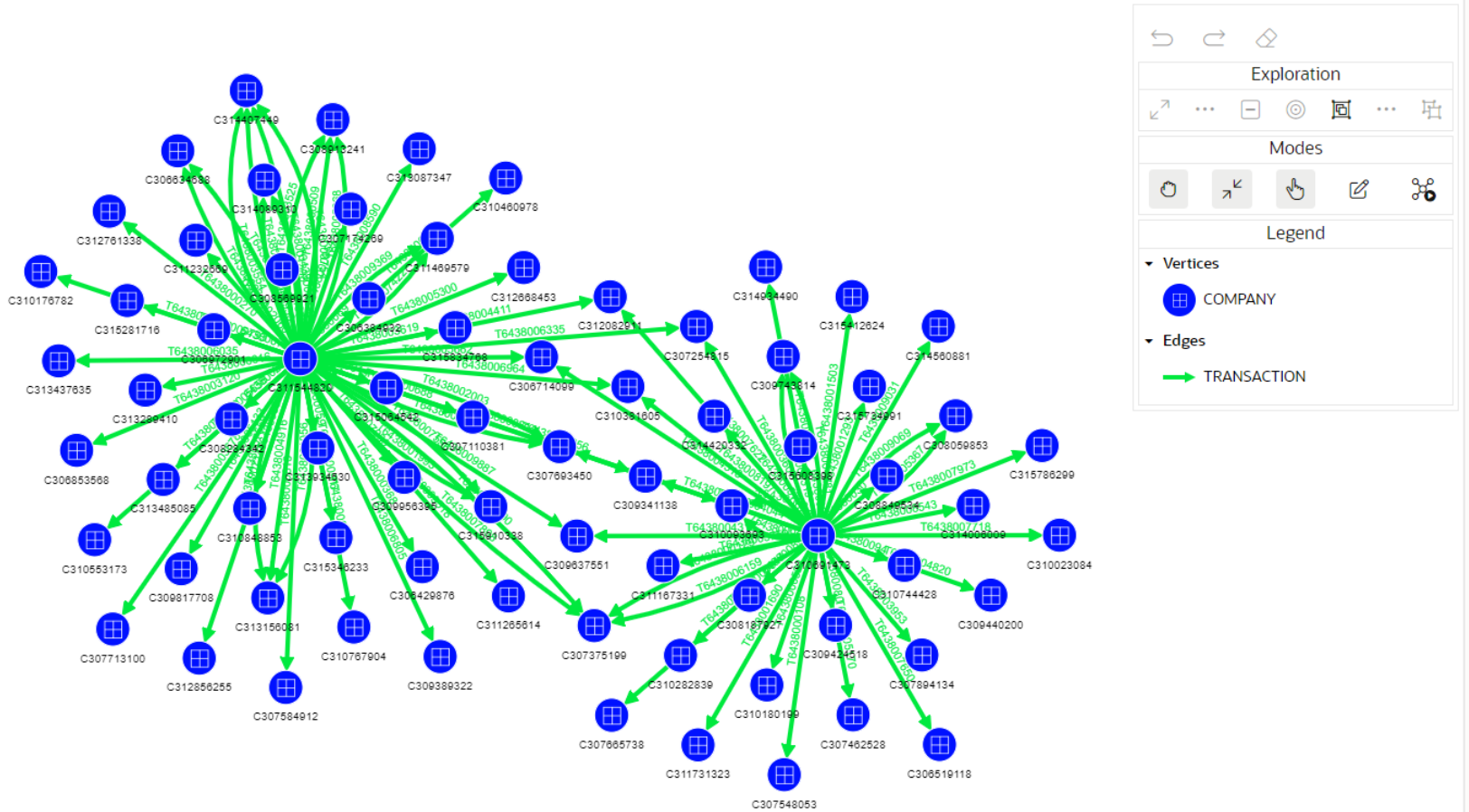
Graph Model

Data Scientist builds a Graph model to investigate cyclic money transfers and companies who are at the center of these transactions.



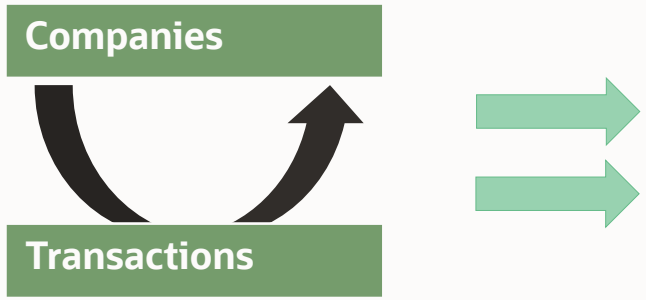
Graph Model

```
%pgql-pgx
/* graph, full */
SELECT
    c1, c2, t12
FROM
    MATCH (c1) -[t12]-> (c2) on TF_PG_TAXFRAUD
```



Graph Metrics

Data Scientist identifies 1-, 2- and 3-step cycles of transactions as well as calculates importance scores of the companies.



fraud patterns

importance

1-step cycles
2-step cycles
3-step cycles

200 nodes
10K edges

betweenness centrality
closeness centrality

Graph Metrics

```
%pgql-pgx
/* 1-step cycle, (A)->(B)->(A) */
SELECT
    a.COMPANYID as "ID",
    a.COMPANYNAME as "NAME",
    count(*) as "SCORE"
FROM
    MATCH (a) -[t1]-> (b) on TF_PG_TAXFRAUD,
    MATCH (b) -[t2]-> (a) on TF_PG_TAXFRAUD
GROUP BY a
ORDER BY count(*) desc
```

ID	NAME	SCORE
C309440200	Company #0064	59
C313862631	Company #0154	45
C312947258	Company #0135	42
C306429876	Company #0002	41
C308753692	Company #0048	36

```
%pgql-pgx
/* 2-step cycle, (A)->(B)->(C)->(A) */
SELECT
    a.COMPANYID as "ID",
    a.COMPANYNAME as "NAME",
    count(*) as "SCORE"
FROM
    MATCH (a) -[t1]-> (b) on TF_PG_TAXFRAUD,
    MATCH (b) -[t2]-> (c) on TF_PG_TAXFRAUD,
    MATCH (c) -[t3]-> (a) on TF_PG_TAXFRAUD
GROUP BY a
ORDER BY count(*) desc
```

ID	NAME	SCORE
C309440200	Company #0064	2733
C313862631	Company #0154	2088
C306429876	Company #0002	1721
C312947258	Company #0135	1665
C313934530	Company #0155	1582

```
%pgql-pgx
/* 3-step cycle, (A)->(B)->(C)->(D)->(A) */
SELECT
    a.COMPANYID as "ID",
    a.COMPANYNAME as "NAME",
    count(*) as "SCORE"
FROM
    MATCH (a) -[t1]-> (b) on TF_PG_TAXFRAUD,
    MATCH (b) -[t2]-> (c) on TF_PG_TAXFRAUD,
    MATCH (c) -[t3]-> (d) on TF_PG_TAXFRAUD,
    MATCH (d) -[t4]-> (a) on TF_PG_TAXFRAUD
GROUP BY a
ORDER BY count(*) desc
```

ID	NAME	SCORE
C309440200	Company #0064	134808
C313862631	Company #0154	103939
C306429876	Company #0002	90206
C312947258	Company #0135	82454
C313934530	Company #0155	81816

```
%pgql-pgx
/* betweenness centrality */
SELECT
    a.COMPANYID as "ID",
    a.COMPANYNAME as "NAME",
    a.BETWEENNESS as "SCORE"
FROM
    MATCH (a) -[t]-> (b) on TF_PG_TAXFRAUD
GROUP BY a
ORDER BY a.BETWEENNESS desc
```

Type to search

ID	NAME	SCORE
C309440200	Company #0064	712.1449272244015
C313862631	Company #0154	537.4907683695352
C306429876	Company #0002	444.29669081790325
C315346233	Company #0186	423.10032538396234
C312947258	Company #0135	401.4715513258675

```
%pgql-pgx
/* closeness centrality */
SELECT
    a.COMPANYID as "ID",
    a.COMPANYNAME as "NAME",
    a.CLOSENESS as "SCORE"
FROM
    MATCH (a) -[t]-> (b) on TF_PG_TAXFRAUD
GROUP BY a
ORDER BY a.CLOSENESS desc
```

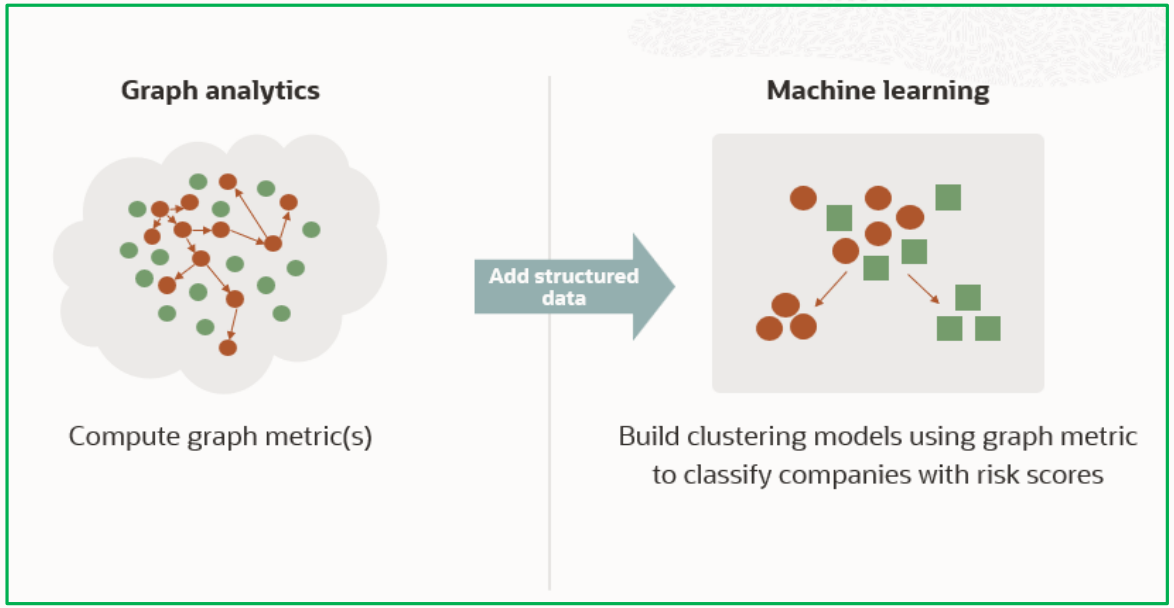
Type to search

ID	NAME	SCORE
C309440200	Company #0064	0.0038461538461538464
C313862631	Company #0154	0.0035587188612099642
C306429876	Company #0002	0.0035211267605633804
C315346233	Company #0186	0.0034965034965034965
C307110381	Company #0016	0.003484320557491289

Detecting Companies with High Fraud Risk

Fraud Investigator would like to understand the list of companies with high fraud potential.

Data Scientist uses Graph metrics to segment companies to provide higher risk companies to fraud investigator.



Data Preparation (for Machine Learning)

Data Scientist prepares data for segmenting companies in terms of fraud risks.

Companies

1-step cycles
2-step cycles
3-step cycles

betweenness centrality
closeness centrality

Companies

1-step cycles score
2-step cycles score
3-step cycles score
betweenness centrality score
closeness centrality score

Data Preparation (for Machine Learning)

```python
%python

# imports
import warnings
import oml
import pandas

# settings
warnings.filterwarnings('ignore')
pandas.set_option('display.width', 2000)
```

FINISHED

```python
%python

# sync proxy objects
TF_WIP_PG_CYCLE1 = oml.sync(table="TF_WIP_PG_CYCLE1")
TF_WIP_PG_CYCLE2 = oml.sync(table="TF_WIP_PG_CYCLE2")
TF_WIP_PG_CYCLE3 = oml.sync(table="TF_WIP_PG_CYCLE3")
TF_WIP_PG_BCENTRAL = oml.sync(table="TF_WIP_PG_BCENTRAL")
TF_WIP_PG_CCENTRAL = oml.sync(table="TF_WIP_PG_CCENTRAL")

# pull data
data = {}
data['TF_WIP_PG_CYCLE1'] = TF_WIP_PG_CYCLE1.pull()
data['TF_WIP_PG_CYCLE2'] = TF_WIP_PG_CYCLE2.pull()
data['TF_WIP_PG_CYCLE3'] = TF_WIP_PG_CYCLE3.pull()
data['TF_WIP_PG_BCENTRAL'] = TF_WIP_PG_BCENTRAL.pull()
data['TF_WIP_PG_CCENTRAL'] = TF_WIP_PG_CCENTRAL.pull()
```

FINISHED

```python
%python

# transform data 'METRICS'
data['TF_WIP_PG_METRICS'] = pandas.merge(data['TF_WIP_PG_CYCLE1'], data['TF_WIP_PG_CYCLE2'], left_on=['ID'], right_on=['ID'], how='inner')
data['TF_WIP_PG_METRICS'] = data['TF_WIP_PG_METRICS'].drop(columns=['NAME_y'])
data['TF_WIP_PG_METRICS'] = data['TF_WIP_PG_METRICS'].rename(columns={'NAME_x':'NAME','SCORE_x':'SCORE_CYC1','SCORE_y':'SCORE_CYC2'})
data['TF_WIP_PG_METRICS'] = pandas.merge(data['TF_WIP_PG_METRICS'], data['TF_WIP_PG_CYCLE3'], left_on=['ID'], right_on=['ID'], how='inner')
data['TF_WIP_PG_METRICS'] = data['TF_WIP_PG_METRICS'].drop(columns=['NAME_y'])
data['TF_WIP_PG_METRICS'] = data['TF_WIP_PG_METRICS'].rename(columns={'NAME_x':'NAME','SCORE':'SCORE_CYC3'})
data['TF_WIP_PG_METRICS'] = pandas.merge(data['TF_WIP_PG_METRICS'], data['TF_WIP_PG_BCENTRAL'], left_on=['ID'], right_on=['ID'], how='inner')
data['TF_WIP_PG_METRICS'] = data['TF_WIP_PG_METRICS'].drop(columns=['NAME_y'])
data['TF_WIP_PG_METRICS'] = data['TF_WIP_PG_METRICS'].rename(columns={'NAME_x':'NAME','SCORE':'SCORE_BCEN'})
data['TF_WIP_PG_METRICS'] = pandas.merge(data['TF_WIP_PG_METRICS'], data['TF_WIP_PG_CCENTRAL'], left_on=['ID'], right_on=['ID'], how='inner')
data['TF_WIP_PG_METRICS'] = data['TF_WIP_PG_METRICS'].drop(columns=['NAME_y'])
data['TF_WIP_PG_METRICS'] = data['TF_WIP_PG_METRICS'].rename(columns={'ID':'COMPANYID','NAME_x':'COMPANYNAME','SCORE':'SCORE_CCEN'})
for col in ['SCORE_CYC1','SCORE_CYC2','SCORE_CYC3','SCORE_BCEN','SCORE_CCEN']:
    data['TF_WIP_PG_METRICS'][col] = round(100 * (data['TF_WIP_PG_METRICS'][col] - data['TF_WIP_PG_METRICS'][col].min()) / (data['TF_WIP_PG_METRICS'][col].max() - data['TF_WIP_PG_METRICS'][col].min()), 2)

# check
z.show(data['TF_WIP_PG_METRICS'])
```

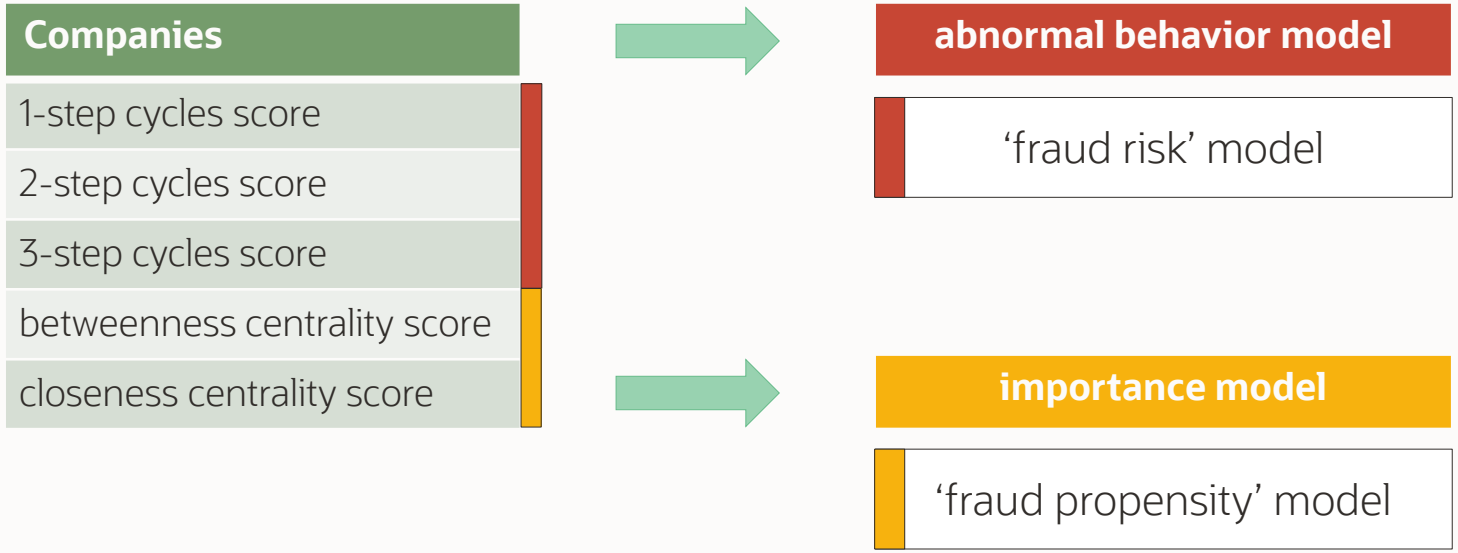
FINISHED

settings

COMPANYID	COMPANYNAME	SCORE_CYC1	SCORE_CYC2	SCORE_CYC3	SCORE_BCEN	SCORE_CCEN
C308732444	Company #0047	3.45	7.82	7.15	8.48	31.14
C309830621	Company #0074	5.17	6.41	5.33	4.49	15.3
C307940687	Company #0034	5.17	11.63	11.72	12.45	27.97
C310691473	Company #0091	6.9	6.04	6.19	3.7	16.01
C309204505	Company #0056	8.62	16.3	17.65	11.13	28.75
C314465049	Company #0171	10.34	11.04	10.89	10.5	33.57
C308125507	Company #0037	12.07	14.97	16.44	17.81	33.57
C309341138	Company #0059	12.07	15.56	15.91	17.88	35.22

Machine Learning Model

Data Scientist builds two machine learning models to detect fraud risk and propensity for fraud of the companies.



Machine Learning Model

```
%python

# prepare data
TF_WIP_CL_CYCLES_IN = oml.push(data['TF_WIP_PG_METRICS'][['COMPANYID','SCORE_CYC1','SCORE_CYC2','SCORE_CYC3']])
TF_WIP_CL_CENTRAL_IN = oml.push(data['TF_WIP_PG_METRICS'][['COMPANYID','SCORE_BCEN','SCORE_CCEN']])

# build models
try:
    oml.drop(model="TF_WIP_CL_CYCLES")
    oml.drop(model="TF_WIP_CL_CENTRAL")
except:
    pass
setting = {'KMNS_ITERATIONS':100, 'KMNS_RANDOM_SEED':6}
TF_WIP_CL_CYCLES = oml.km(n_clusters=5, **setting).fit(TF_WIP_CL_CYCLES_IN, case_id='COMPANYID', model_name="TF_WIP_CL_CYCLES")
TF_WIP_CL_CENTRAL = oml.km(n_clusters=5, **setting).fit(TF_WIP_CL_CENTRAL_IN, case_id='COMPANYID', model_name="TF_WIP_CL_CENTRAL")
```

FINISHED

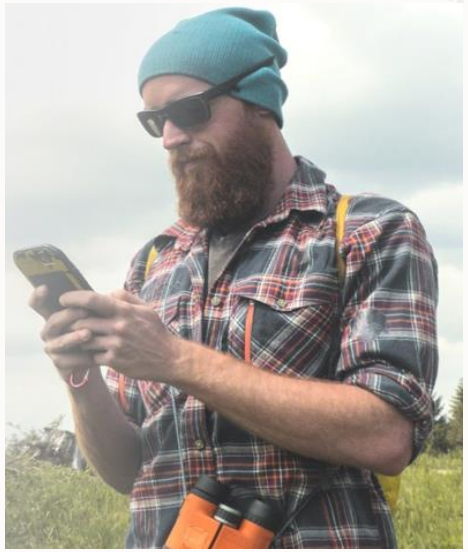
```
Model Name: TF_WIP_CL_CYCLES

Model Owner: VALENTIN

Algorithm Name: K-Means

Mining Function: CLUSTERING

Settings:
                    setting name           setting value
0                      ALGO_NAME            ALGO_KMEANS
1              CLUS_NUM_CLUSTERS                      5
2            KMNS_CONV_TOLERANCE                   .001
3                   KMNS_DETAILS   KMNS_DETAILS_HIERARCHY
4                  KMNS_DISTANCE          KMNS_EUCLIDEAN
5                 KMNS_ITERATIONS                    100
```

```
Model Name: TF_WIP_CL_CENTRAL

Model Owner: VALENTIN

Algorithm Name: K-Means

Mining Function: CLUSTERING

Settings:
                    setting name           setting value
0                      ALGO_NAME            ALGO_KMEANS
1              CLUS_NUM_CLUSTERS                      5
2            KMNS_CONV_TOLERANCE                   .001
3                   KMNS_DETAILS   KMNS_DETAILS_HIERARCHY
4                  KMNS_DISTANCE          KMNS_EUCLIDEAN
5                 KMNS_ITERATIONS                    100
```

COMPANYID	SCORE_CYC1	SCORE_CYC2	SCORE_CYC3	SCORE_BCEN	SCORE_CCEN	CLUSTER_CYC	CLUSTER_CEN
C315592736	0	0	0	0	0	Cluster #1	Cluster #1
C310769904	5	2	2	2	7	Cluster #1	Cluster #1
C312022089	2	3	2	2	11	Cluster #1	Cluster #1
C314201266	2	3	2	2	13	Cluster #1	Cluster #1
C311348313	2	4	5	6	20	Cluster #1	Cluster #1
C309639877	3	4	4	3	11	Cluster #1	Cluster #1
C315509232	3	4	4	4	17	Cluster #1	Cluster #1
C312992057	3	5	5	4	15	Cluster #1	Cluster #1

Business Insights

Data Scientist shares tax fraud segments with Business Analyst.

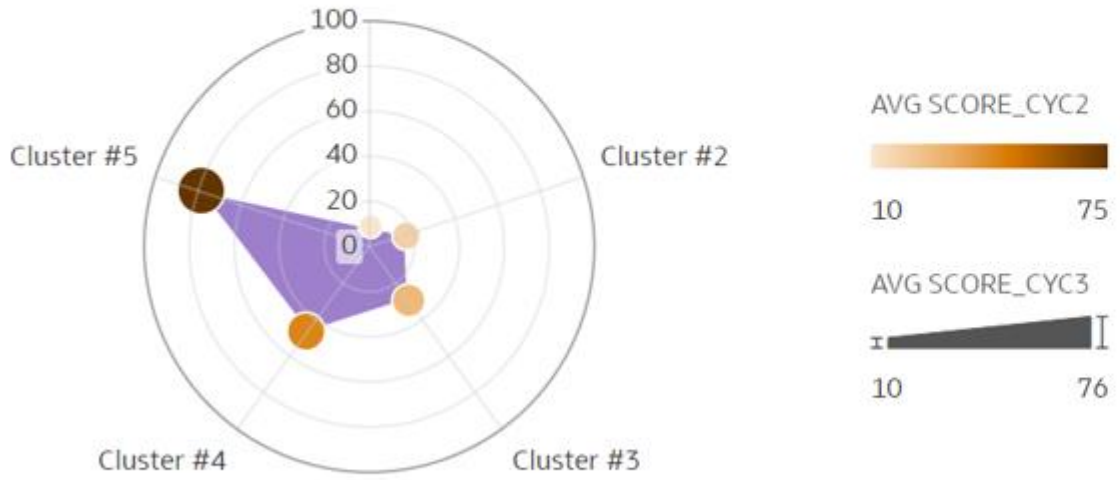


Business Analyst creates dashboards for Fraud Investigator.

Fraud Investigator accesses all high-risk companies at the click of a button.

Business Insights

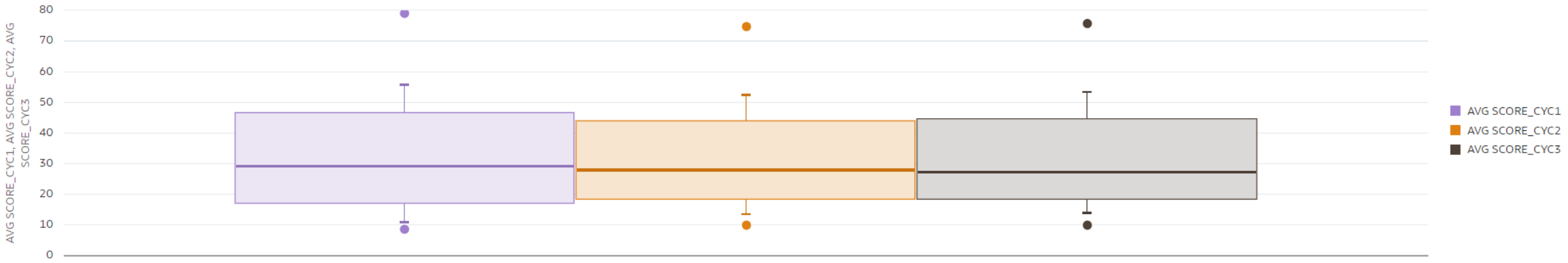
CYCLIC TRANSACTIONS



CENTRAL COMPANIES

Business Insights

EXTREME SCORES

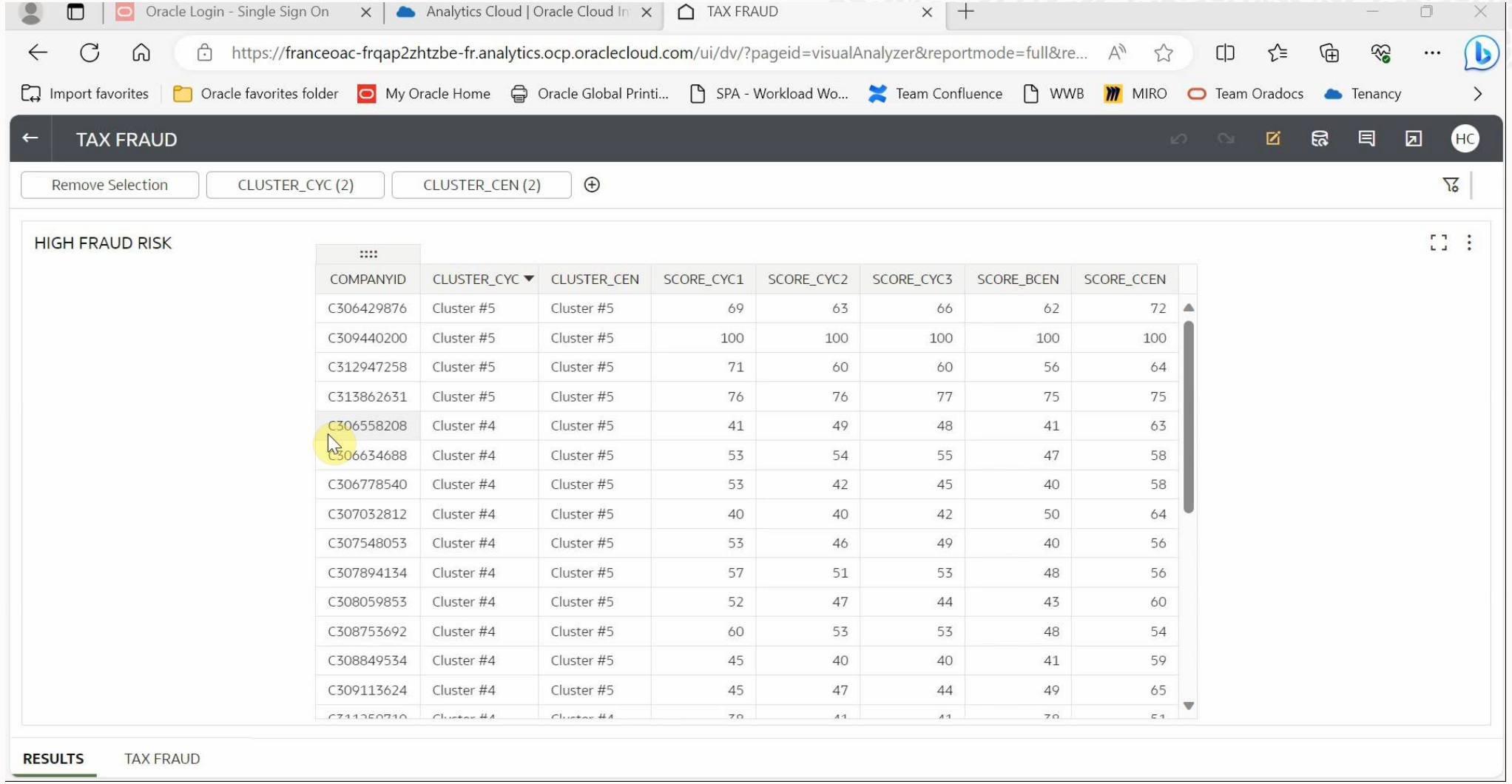


HIGH RISK COMPNIES

CLUSTER_CYC: Cluster #5 CLUSTER_CEN: Cluster #5

COMPANYID	CLUSTER_CYC	CLUSTER_CEN
C306429876	Cluster #5	Cluster #5
C309440200	Cluster #5	Cluster #5
C312947258	Cluster #5	Cluster #5
C313862631	Cluster #5	Cluster #5

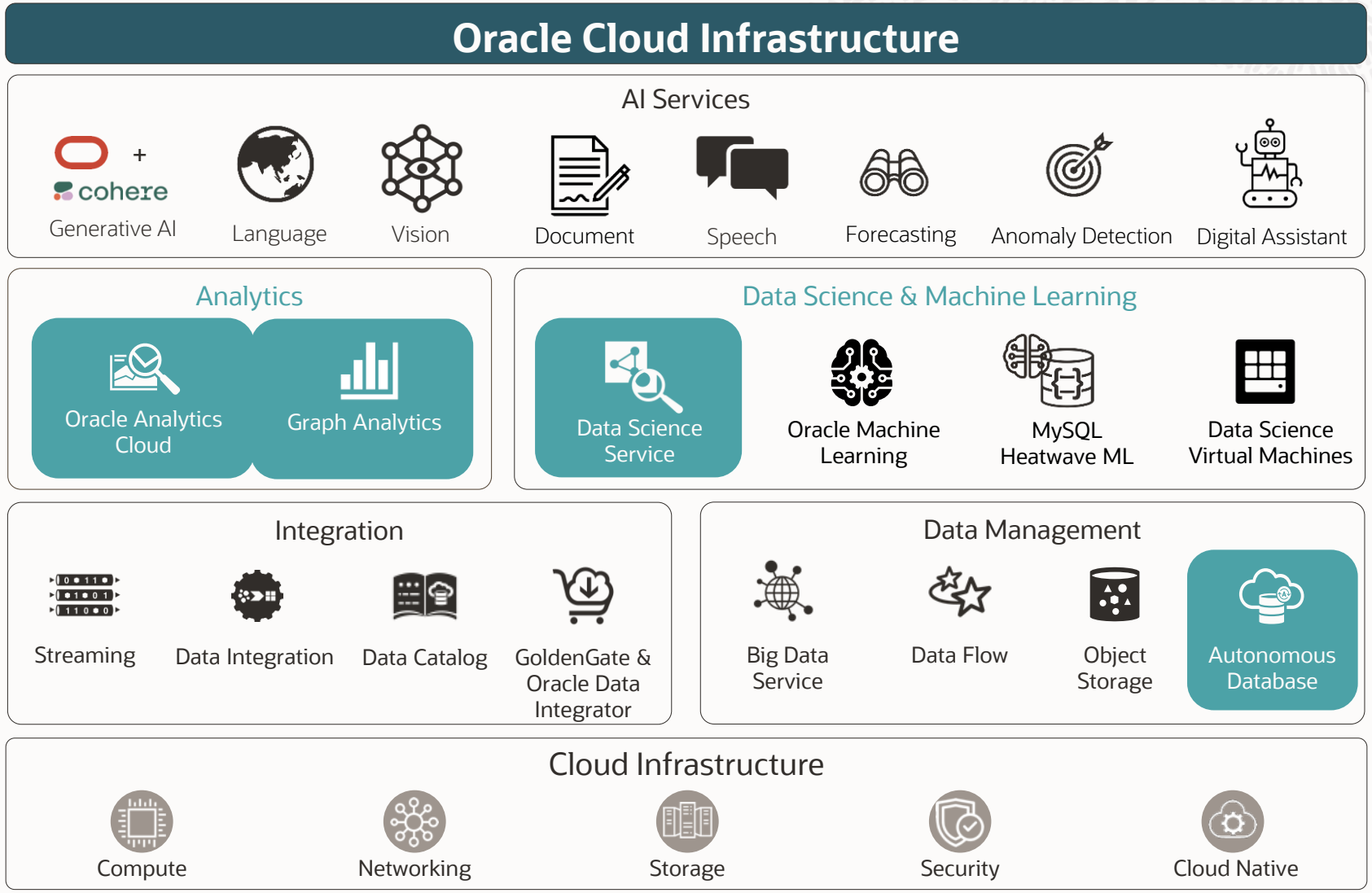
Business Insights



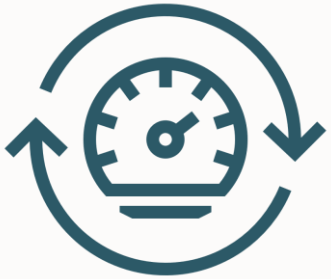
COMPANYID	CLUSTER_CYC	CLUSTER_CEN	SCORE_CYC1	SCORE_CYC2	SCORE_CYC3	SCORE_BCEN	SCORE_CCEN
C306429876	Cluster #5	Cluster #5	69	63	66	62	72
C309440200	Cluster #5	Cluster #5	100	100	100	100	100
C312947258	Cluster #5	Cluster #5	71	60	60	56	64
C313862631	Cluster #5	Cluster #5	76	76	77	75	75
C306558208	Cluster #4	Cluster #5	41	49	48	41	63
C306634688	Cluster #4	Cluster #5	53	54	55	47	58
C306778540	Cluster #4	Cluster #5	53	42	45	40	58
C307032812	Cluster #4	Cluster #5	40	40	42	50	64
C307548053	Cluster #4	Cluster #5	53	46	49	40	56
C307894134	Cluster #4	Cluster #5	57	51	53	48	56
C308059853	Cluster #4	Cluster #5	52	47	44	43	60
C308753692	Cluster #4	Cluster #5	60	53	53	48	54
C308849534	Cluster #4	Cluster #5	45	40	40	41	59
C309113624	Cluster #4	Cluster #5	45	47	44	49	65

Behind the Scenes

Behind the Scenes



Autonomous Database



Self-Driving

- Scale-out database with fault-tolerance and DR
- Runs on enterprise-proven Exadata platform
- Full compatibility with existing enterprise databases

Self-Securing

- Automatically applies security updates online
- Secure configuration with full database encryption
- Sensitive data hidden from Oracle or customer admins

Self-Repairing

- Recovers automatically from any failure
- 99.995% uptime including maintenance
- Elastically scales compute or storage as needed

Oracle Graph

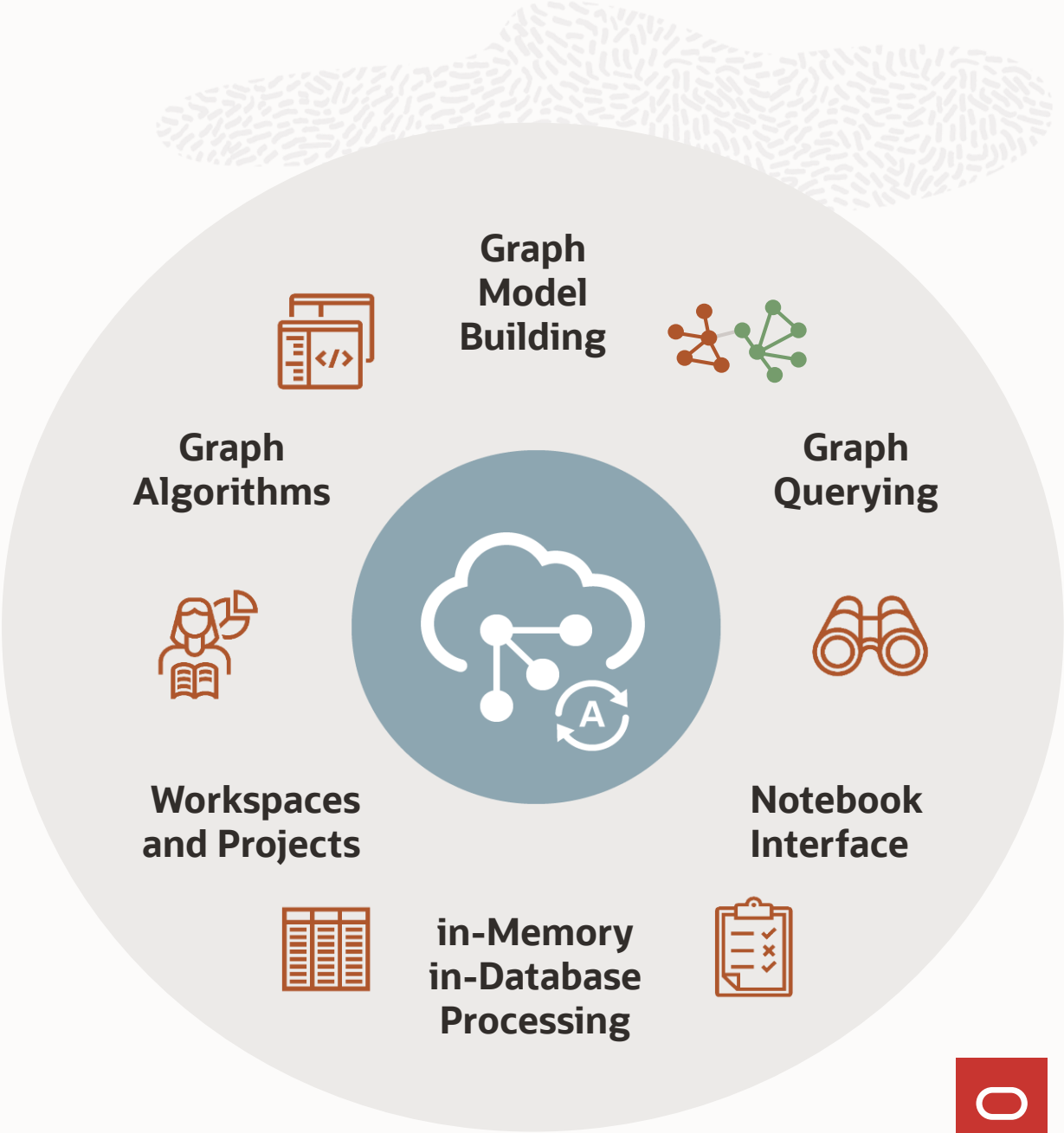
Model entities and relationships using a graph approach

Leverage comprehensive tooling and automation for graph modelling

Benefit from 50+ pre-built graph algorithms, graph querying language and visualization layer

Collaborate with teammates on shareable and reproducible graph assets

Run scalable and highly performant workloads on a managed environment



Oracle Machine Learning

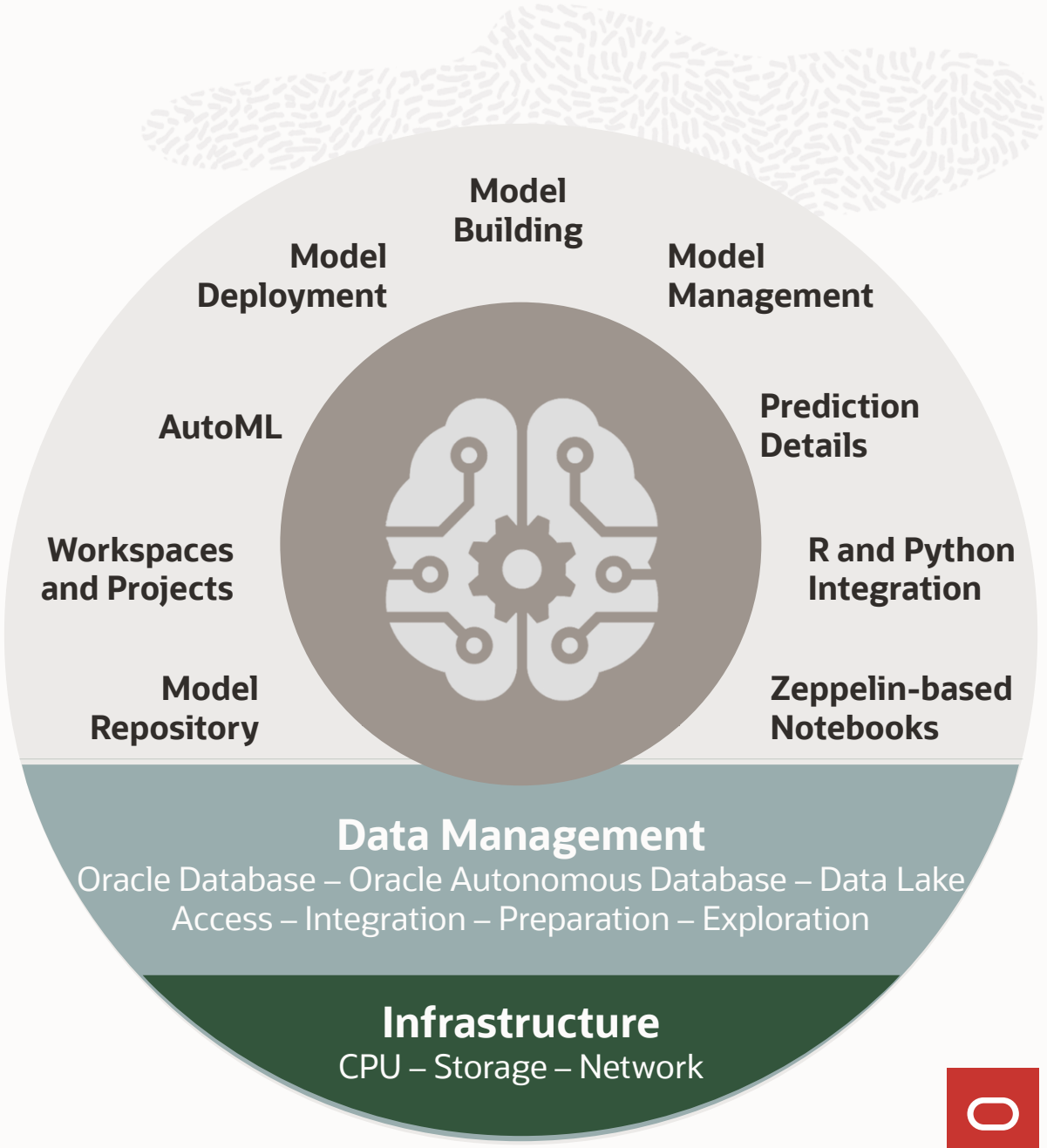
Support data scientist productivity and empower non-experts with AutoML

Gain algorithm-specific data preparation, integrated text mining, and partitioned models

Benefit from over 30+ high performance in-database machine learning algorithms

Deploy and update machine learning models in production via SQL and REST APIs

Deploy R and Python user-defined functions using managed processes



Data Science Service

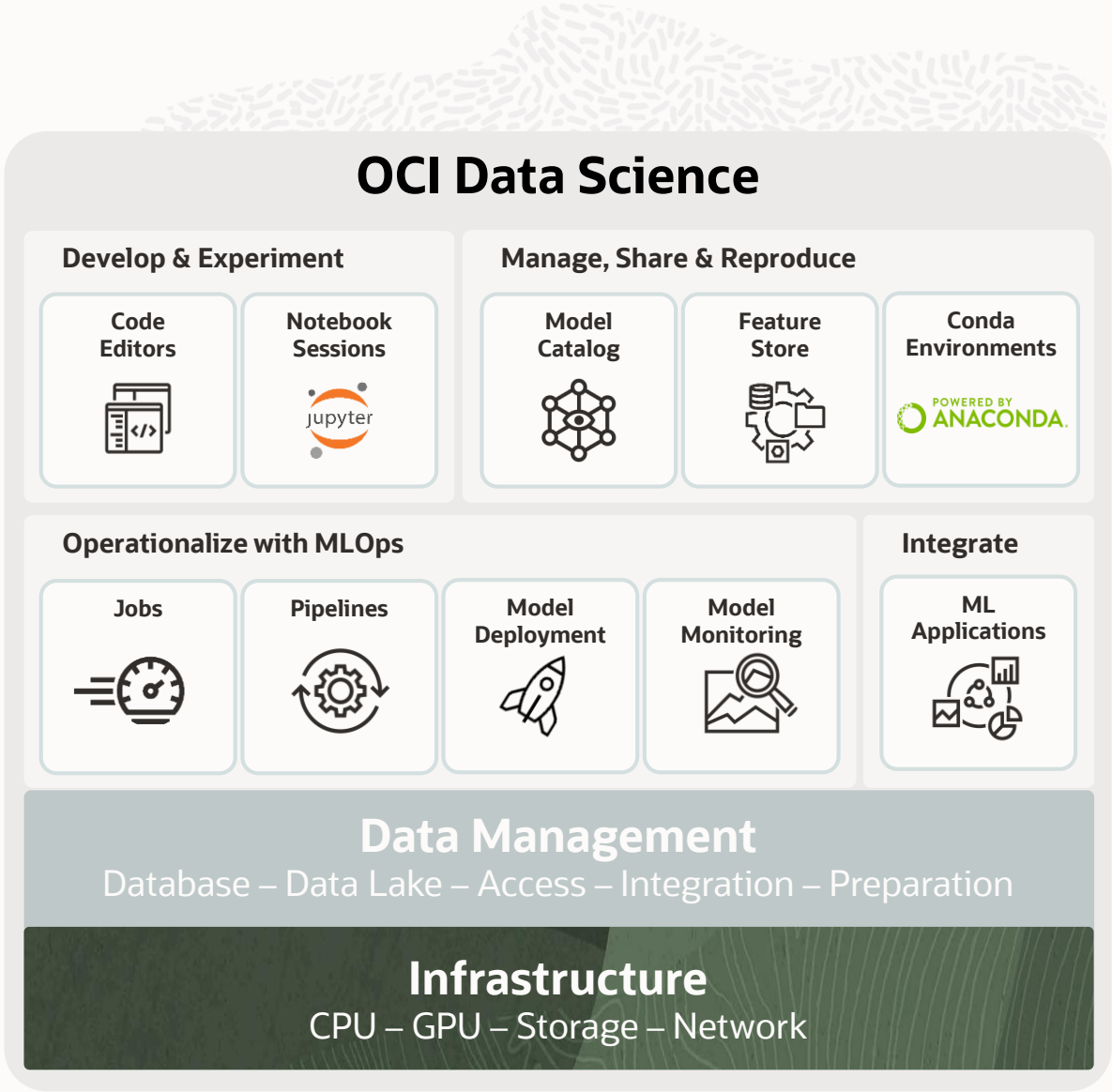
Accelerate and automate the entire end-to-end data science lifecycle

Use favorite open-source Python tools and frameworks

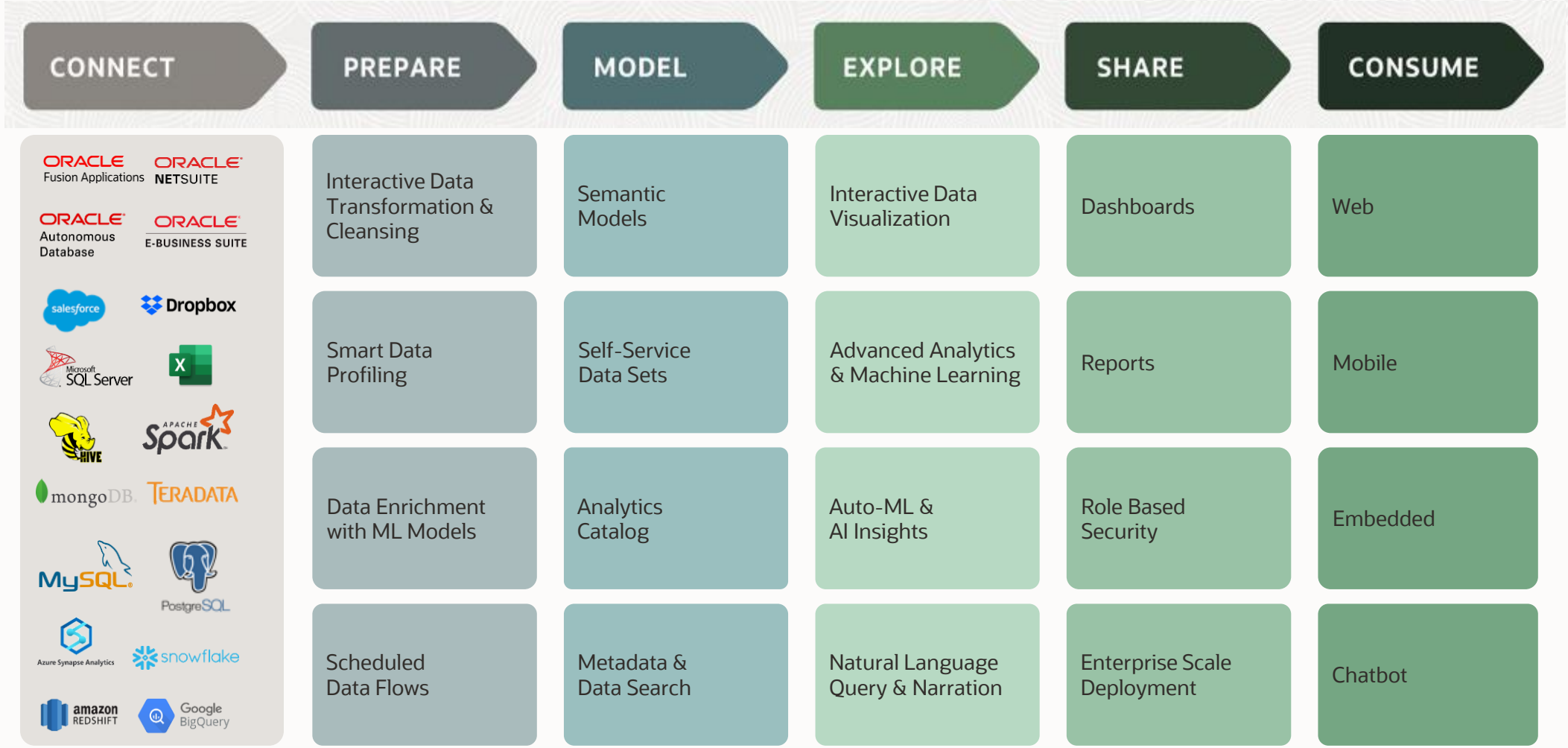
Gain enterprise-grade MLOps with flexible interfaces and unlimited scale

Collaborate with teammates on shareable and reproducible data science assets

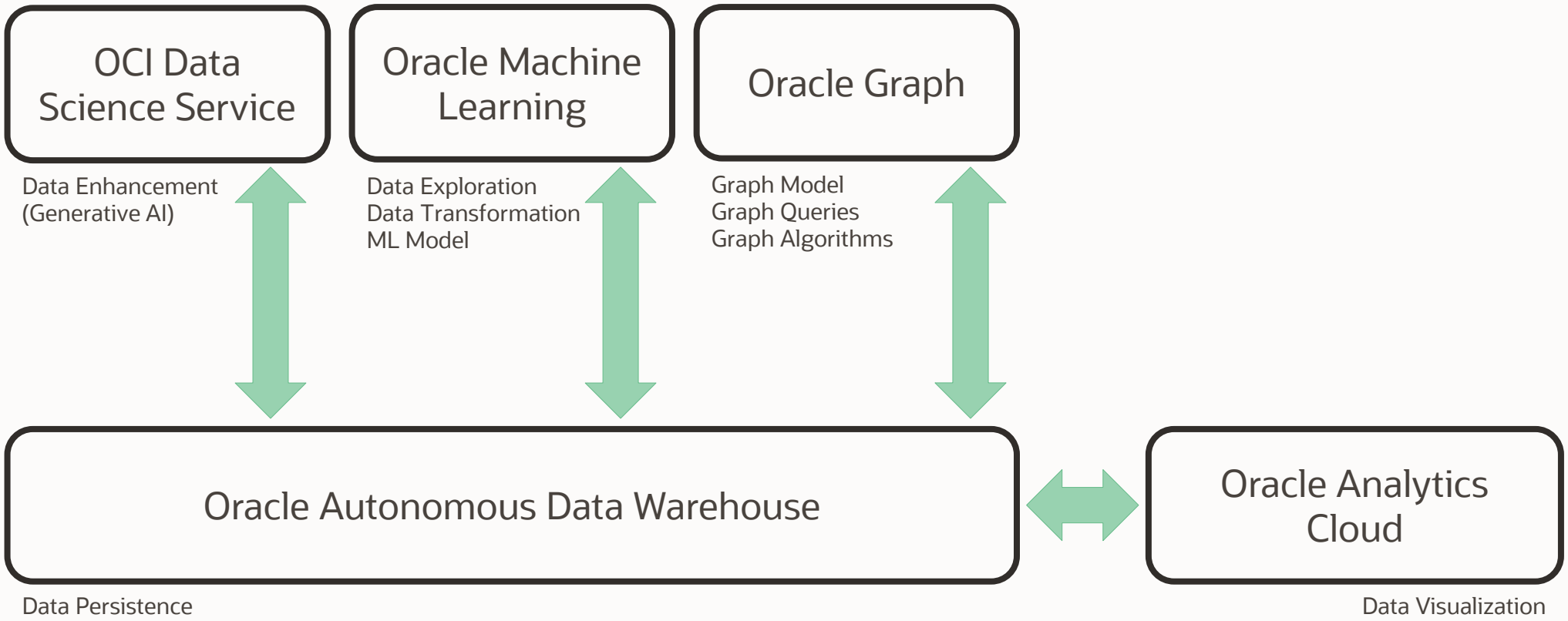
Run large-scale workloads with access to bare metal GPUs and distributed data processing and model training



Oracle Analytics Cloud



High Level Architecture



Other Potential Contributions


ORACLE