

ORACLE

Art of Possible with AI & Data Science

Leveraging Procurement Vendor Segmentation, Anomaly Detection and Forecasting

August 2023

EMEA Data Science Specialists



Demo Inspiration

This demonstration will showcase the capabilities of Oracle Data Science/Machine Learning Platform in aiding procurement teams with data-driven decision-making.

Objective is:

- **Classifying vendors** or suppliers into different categories based on various criteria to streamline procurement processes
- Identifying unusual or **abnormal patterns** in spending data
- **Forecasting** total daily/weekly/monthly procurement spending for each vendor.



Demo Flow

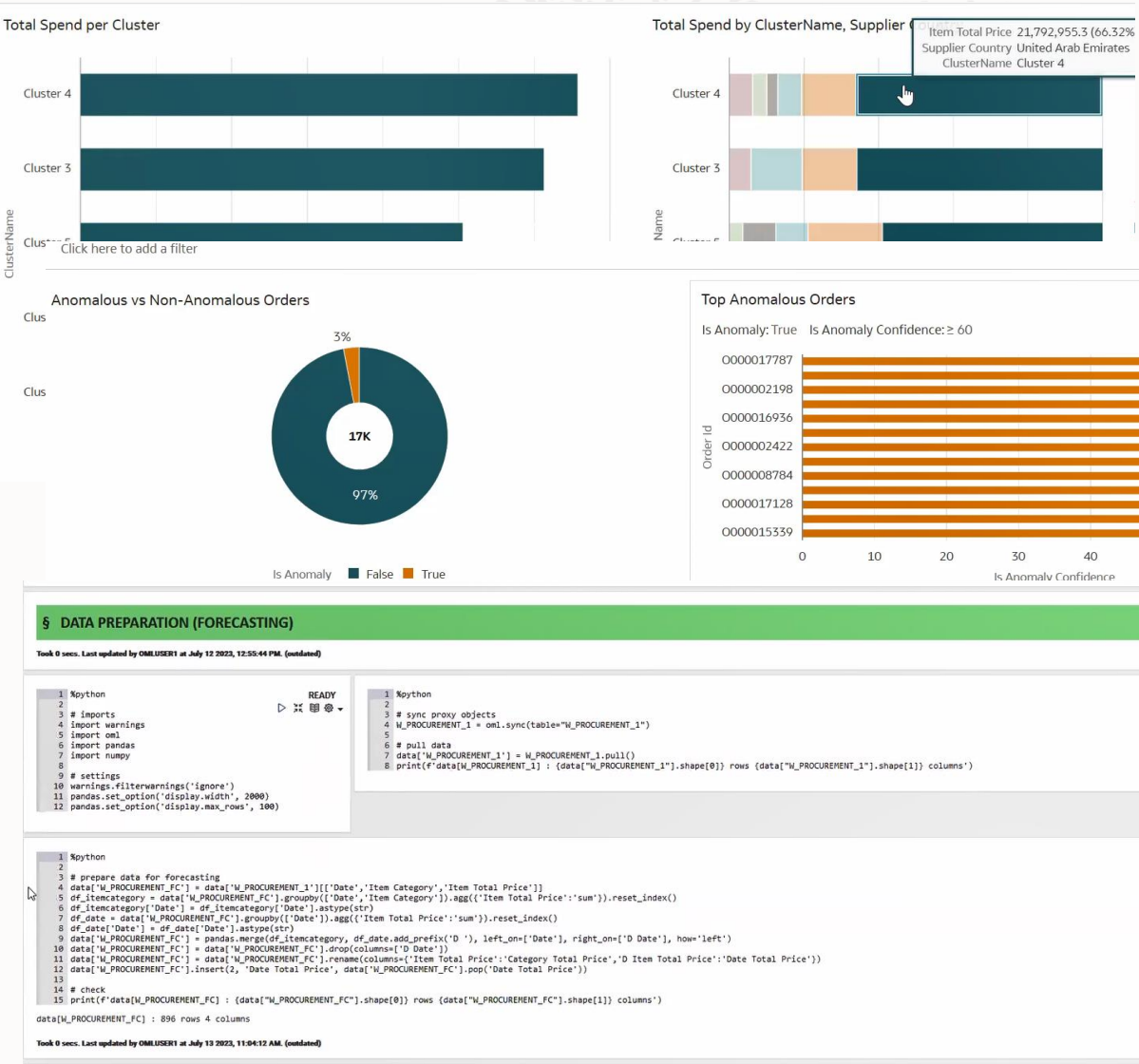
1. Summary

2. How did we achieve this?

- Data Discovery & Preparation
- Modeling
- Actionable Insights for Business

3. Behind the Scenes

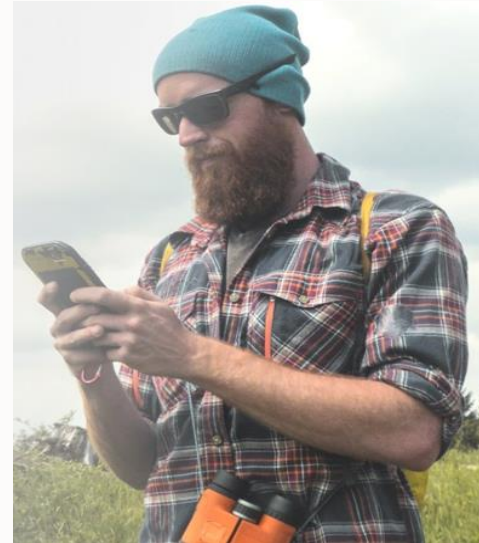
- Oracle Data Science/ML Platform



Target Personas



Procurement manager wants to understand future procurement spendings, any anomalies in spendings and different vendor categories to manage relationships with suppliers effectively



Data scientists wants an end-to-end platform to address data science lifecycle and help procurement decision making process with ML

Summary

At the end of this demo, the questions that procurement manager will be able to answer are:

Vendor Segmentation

How can we categorize and segment our vendors based on their characteristics, performance, or strategic importance?

Which vendors are most suitable for specific procurement needs or categories?

Understanding Anomalies in Procurement Spendings

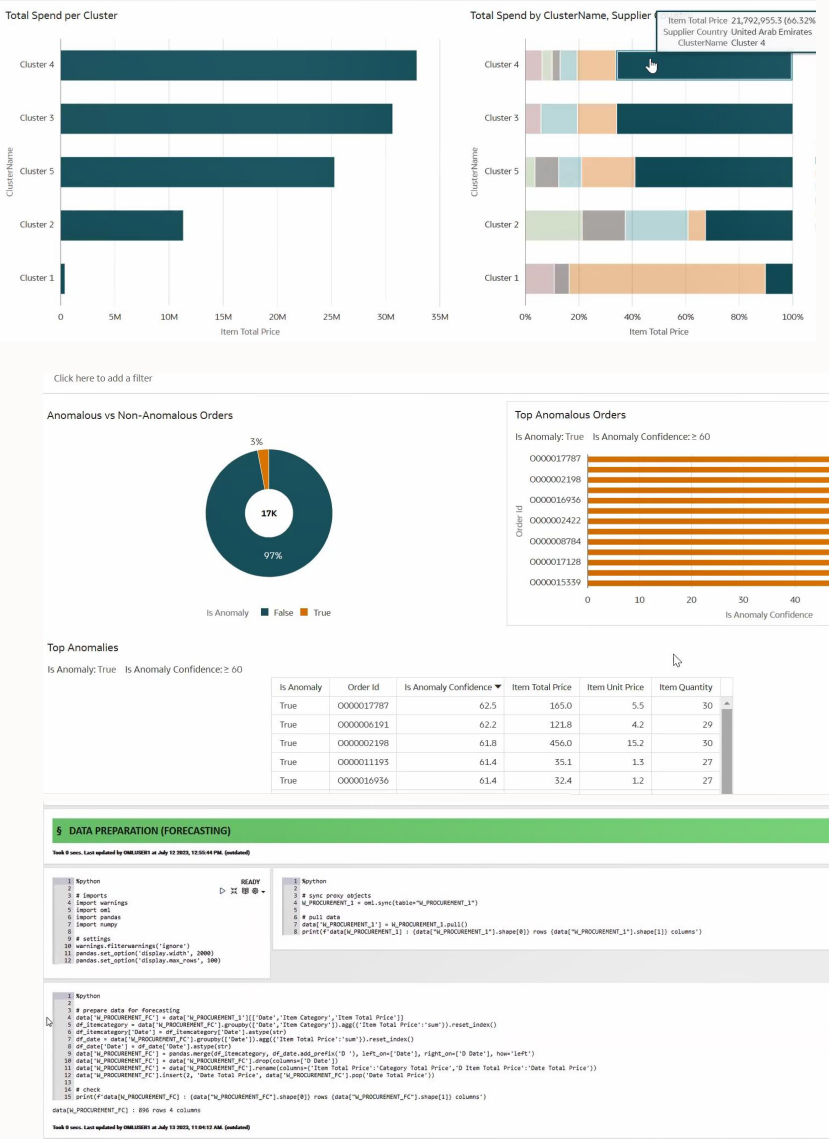
Are there any unusual or suspicious spending patterns that might indicate abnormal spending?

Which procurement transactions exhibit spending patterns that deviate significantly from the norm?

Procurement Spending Forecasting

What will be the expected spending levels for specific procurement categories in the future?

Can we identify potential cost-saving opportunities by analyzing historical spending patterns?





How have we achieved this?

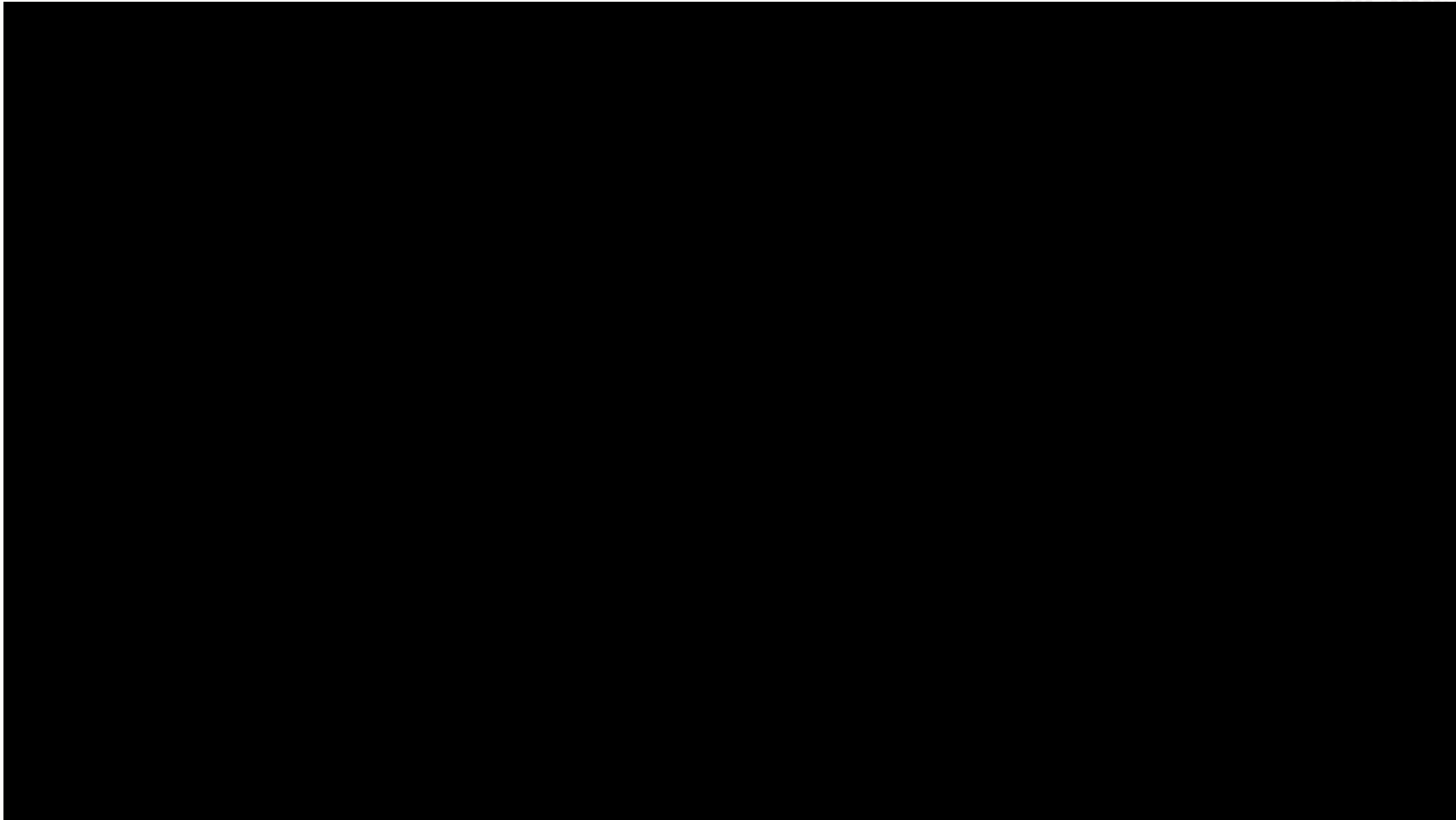
Initial Data Discovery & Preprocessing



Procurement manager imports data to OAC. Discovers data that can provide an initial insight about procurement spendings and improves data with her business expertise



Initial Data Discovery & Preprocessing



Data Preprocessing Cont'd



Procurement manager wants to have a high-quality time series data to predict future spendings as a citizen data scientist



Data scientist assesses data quality issues, fix them and shares final time series procurement data with procurement manager

Data Preprocessing Cont'd

The screenshot displays the Oracle Machine Learning (OML) workspace interface. At the top, the header shows 'ORACLE Machine Learning' and 'OMLUSER1 Project'. The main workspace area is titled 'Procurement-Demo-WIP' and shows a 'DATA QUALITY' section. The interface indicates that the data was last updated by OMLUSER1 at July 11 2023, 3:17:04 PM. Below this, two code snippets are shown, each with a 'FINISHED' status indicator.

Code Snippet 1:

```
1 %python
2
3 # check NULL values
4 for col in data['W_PROCEUREMENT_1'].columns:
5     nulls = data['W_PROCEUREMENT_1'][col].isna().sum()
6     ratio = round(100*nulls/data['W_PROCEUREMENT_1'].shape[0])
7     print(f'data[W_PROCEUREMENT_1]["{col}"] : {ratio} % nulls')
8
9     data[W_PROCEUREMENT_1]["Date"] : 0 (0%) nulls
10    data[W_PROCEUREMENT_1]["Day Of Week"] : 0 (0%) nulls
11    data[W_PROCEUREMENT_1]["Is Weekend"] : 0 (0%) nulls
12    data[W_PROCEUREMENT_1]["Is Event"] : 0 (0%) nulls
13    data[W_PROCEUREMENT_1]["Supplier Id"] : 0 (0%) nulls
14    data[W_PROCEUREMENT_1]["Supplier Country"] : 870 (5%) nulls
15    data[W_PROCEUREMENT_1]["Supplier Years"] : 1244 (7%) nulls
16    data[W_PROCEUREMENT_1]["Supplier Price Index"] : 2314 (13%) nulls
17    data[W_PROCEUREMENT_1]["Supplier Discount"] : 2359 (13%) nulls
18    data[W_PROCEUREMENT_1]["Item Category"] : 0 (0%) nulls
19    data[W_PROCEUREMENT_1]["Item Quantity"] : 929 (5%) nulls
20    data[W_PROCEUREMENT_1]["Item Unit Price"] : 1248 (7%) nulls
21    data[W_PROCEUREMENT_1]["Item Total Price"] : 2385 (13%) nulls
22    data[W_PROCEUREMENT_1]["Order Category"] : 0 (0%) nulls
23    data[W_PROCEUREMENT_1]["Order Urgency"] : 0 (0%) nulls
24    data[W_PROCEUREMENT_1]["Order Delivery Status"] : 0 (0%) nulls
25    data[W_PROCEUREMENT_1]["Order Id"] : 0 (0%) nulls
```

Code Snippet 2:

```
1 %python
2
3 # check aggregation consistency
4 def checkConsistency(row):
5     return True if row['Min']==row['Max'] else False
6 for col in ['Day Of Week', 'Is Weekend', 'Is Event']:
7     df = data['W_PROCEUREMENT_1']-data['W_PROCEUREMENT_1'][col].isna()[['Date', col]]
8     df = df.groupby(['Date']).agg({'col': ['min', 'max']})
9     df.columns = ['Min', 'Max']
10    df['Check'] = df.apply(checkConsistency, axis=1)
11    inconsistencies = df[df['Check']==False].shape[0]
12    ratio = round(100*inconsistencies/df.shape[0])
13    print(f'data[W_PROCEUREMENT_1]["{col}"] : {inconsistencies} ({ratio}% inconsistencies)')
14 for col in ['Supplier Country', 'Supplier Years', 'Supplier Price Index', 'Supplier Discount']:
15     df = data['W_PROCEUREMENT_1']-data['W_PROCEUREMENT_1'][col].isna()[['Supplier Id', col]]
16     df = df.groupby(['Supplier Id']).agg({'col': ['min', 'max']})
17     df.columns = ['Min', 'Max']
18     df['Check'] = df.apply(checkConsistency, axis=1)
19     inconsistencies = df[df['Check']==False].shape[0]
20     ratio = round(100*inconsistencies/df.shape[0])
```





Anomaly Detection in Procurement Spendings

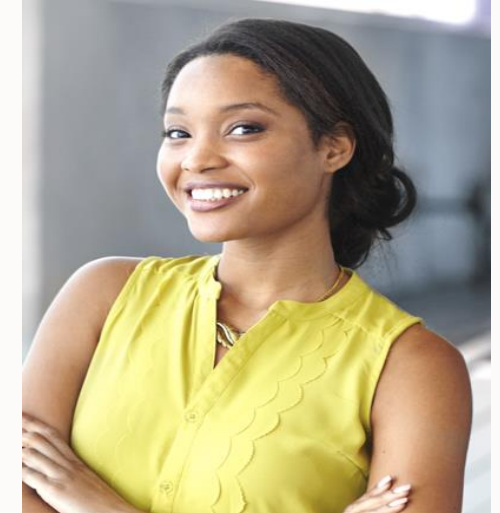
Anomaly Detection in Procurement Spendings



Procurement manager wants to understand anomalies in procurement spendings



Data scientist creates an Anomaly Detection Model and shares spending anomalies data with procurement manager



Procurement manager analyses anomalies to take immediate action.

Anomaly Detection in Procurement Spendings - Modeling

The screenshot displays the Oracle Machine Learning (OML) workspace interface. At the top, the header shows "ORACLE Machine Learning" on the left and "OMLUSER1 Project" and "OMLUSER1" on the right. Below the header, the workspace name "Procurement-Demo-WIP" is visible. The main area contains three notebooks, each with a green header bar indicating its status as "FINISHED".

§ DATA PREPARATION (ANOMALY DETECTION)

1 Python

```
1 # imports
2 # sync proxy objects
3 # sync proxy objects
4 W_PROCEUREMENT_1 = oml.sync(table="W_PROCEUREMENT_1")
5
6 # pull data
7 data[W_PROCEUREMENT_1] = W_PROCEUREMENT_1.pull()
8 print(f"data[W_PROCEUREMENT_1] : {data[W_PROCEUREMENT_1].shape[0]} rows {data[W_PROCEUREMENT_1].shape[1]} columns")
9
10 data[W_PROCEUREMENT_1] : 17300 rows 17 columns
```

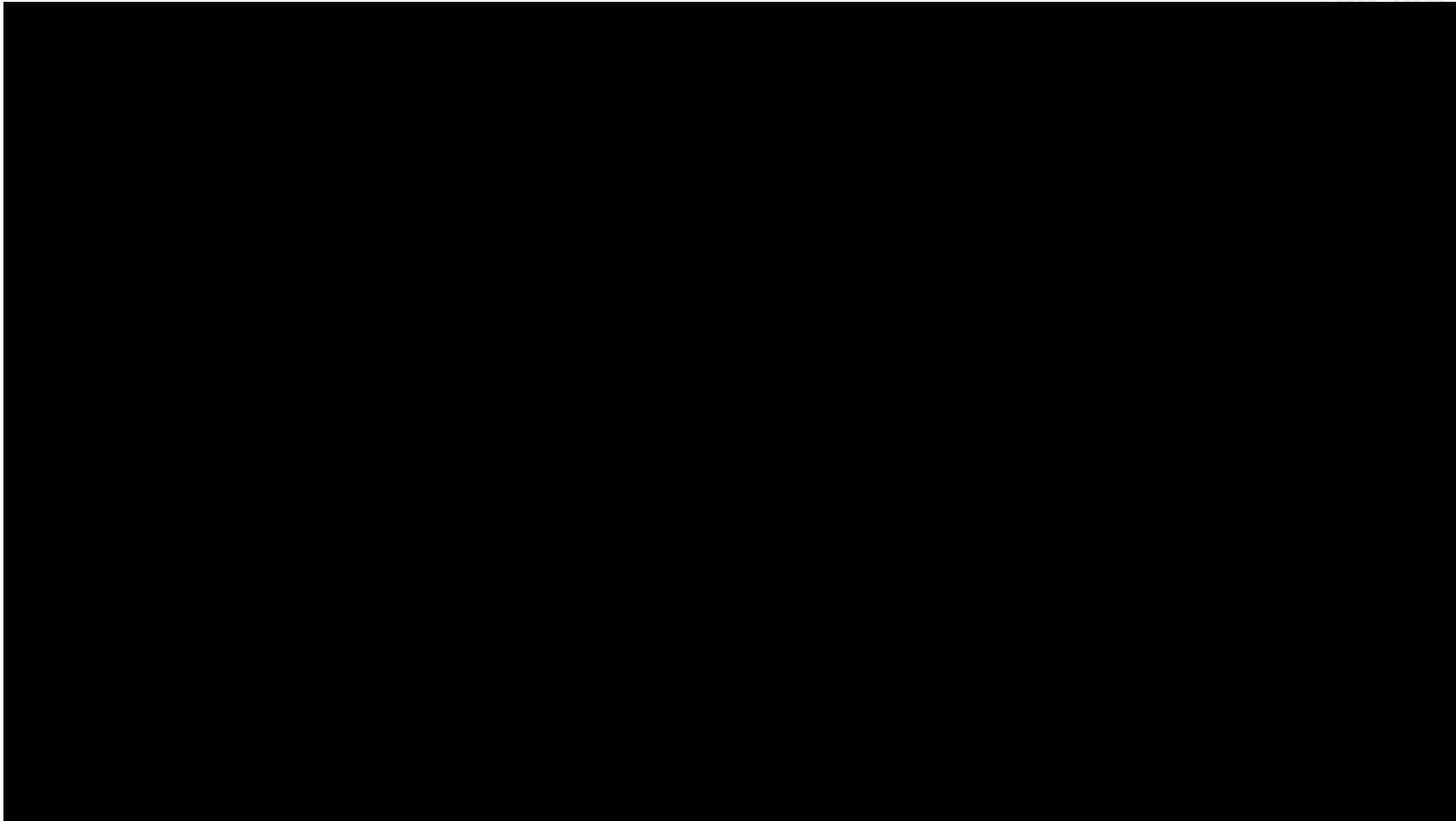
2 Python

```
1 # imports
2 # sync proxy objects
3 # sync proxy objects
4 W_PROCEUREMENT_1 = oml.sync(table="W_PROCEUREMENT_1")
5
6 # pull data
7 data[W_PROCEUREMENT_1] = W_PROCEUREMENT_1.pull()
8 print(f"data[W_PROCEUREMENT_1] : {data[W_PROCEUREMENT_1].shape[0]} rows {data[W_PROCEUREMENT_1].shape[1]} columns")
9
10 data[W_PROCEUREMENT_1] : 17300 rows 17 columns
```

3 Python

```
1 # imports
2 # create anomaly detection model
3 W_PROCEUREMENT_AD = W_PROCEUREMENT_1[['Order Id', 'Item Quantity', 'Item Unit Price', 'Item Total Price']]
4
5 try:
6     oml.drop(model="W_PROCEUREMENT_AD")
7 except:
8     pass
9
10 odm_settings = {'SWMS_OUTLIER_RATE': '0.03',
11                 'SWMS_REGULARIZER': 'SWMS_REGULARIZER_L1',
12                 'SWMS_CONV_TOLERANCE': '0.001'}
13 W_PROCEUREMENT_AD = oml.svm("anomaly_detection", **odm_settings)
14 W_PROCEUREMENT_AD.fit(W_PROCEUREMENT_AD, None, model_name="W_PROCEUREMENT_AD", case_id="Order Id")
15
16 # process results
17 def createIsAnomaly(val):
18     return 'True' if val==0 else 'False'
19 def createIsAnomalyConfidence(val):
20     return round(100*val,1)
21
22 W_PROCEUREMENT_AD = W_PROCEUREMENT_AD.predict(W_PROCEUREMENT_AD, supplemental_cols=W_PROCEUREMENT_AD, proba=True)
23 data[W_PROCEUREMENT_AD] = W_PROCEUREMENT_AD.pull()
24 data[W_PROCEUREMENT_AD]['Is Anomaly'] = data[W_PROCEUREMENT_AD]['PREDICTION'].apply(createIsAnomaly)
25 data[W_PROCEUREMENT_AD]['Is Anomaly Confidence'] = data[W_PROCEUREMENT_AD]['PROBABILITY'].apply(createIsAnomalyConfidence)
26 data[W_PROCEUREMENT_AD] = data[W_PROCEUREMENT_AD].drop(columns=['PREDICTION', 'PROBABILITY'])
27
28 # check
29 print(f"data[W_PROCEUREMENT_AD] : {data[W_PROCEUREMENT_AD].shape[0]} rows {data[W_PROCEUREMENT_AD].shape[1]} columns")
30
31 data[W_PROCEUREMENT_AD] : 17300 rows 6 columns
```

Anomaly Detection in Procurement Spendings – Business Insights





Supplier Segmentation

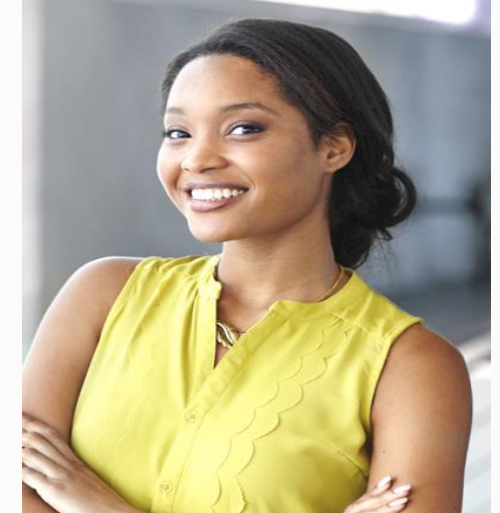
Supplier Segmentation



Procurement manager wants to segment suppliers based on their characteristics. She talks to data scientist to create an aggregated data at supplier level.



Data scientist aggregates data at supplier level and shares this final data with procurement manager



Procurement manager builds supplier segments and discovers different suppliers and optimize procurement strategies

Supplier Segmentation - Data Preparation

ORACLE Machine Learning

OMLUSER1 Project
OMLUSER1 Workspace

OMLUSER1

5 DATA PREPARATION (CLUSTERING)

Took 0 secs. Last updated by OMLUSER1 at July 12 2023, 12:55:31 PM. (outdated)

```
1 %python
2
3 # imports
4 import warnings
5 import oml
6 import pandas
7 import numpy
8
9 # settings
10 warnings.filterwarnings('ignore')
11 pandas.set_option('display.width', 2000)
12 pandas.set_option('display.max_rows', 100)
```

Finished

Took 0 secs. Last updated by OMLUSER1 at July 13 2023, 11:21:10 AM. (outdated)

```
1 %python
2
3 # sync proxy objects
4 W_PROCEUREMENT_1 = oml.sync(table="W_PROCEUREMENT_1")
5
6 # pull data
7 data[W_PROCEUREMENT_1] = W_PROCEUREMENT_1.pull()
8 print(f"data[W_PROCEUREMENT_1] : {data[W_PROCEUREMENT_1].shape[0]} rows {data[W_PROCEUREMENT_1].shape[1]} columns")
```

Finished

Took 1 sec. Last updated by OMLUSER1 at July 13 2023, 12:22:00 PM. (outdated)

data[W_PROCEUREMENT_1] : 17300 rows 17 columns

```
1 %python
2
3 # prepare data for clustering
4 data[W_PROCEUREMENT_CL] = data[W_PROCEUREMENT_1][['Supplier Id','Supplier Country','Supplier Years','Supplier Price Index','Supplier Discount',
5 'Date','Is Weekend','Is Event','Item Category','Item Total Price','Order Category','Order Urgency','Order Delivery Status']]
6 def agg_isval(val):
7     def custom_count(ser):
8         return ser.value_counts().get(val, 0)
9     return custom_count
10 def agg_rndavg():
11     def custom_average(ser):
12         return round(ser.mean())
13     return custom_average
14 data[W_PROCEUREMENT_CL] = data[W_PROCEUREMENT_CL].groupby(['Supplier Id','Supplier Country','Supplier Years','Supplier Price Index','Supplier Discount'],agg=[agg_isval('Beauty'),'Orders ItemCat Electronics':('Item Category',agg_isval('Electronics')),
15 'Orders Is Weekend':('Is Weekend',agg_isval('Yes')), 'Orders Not Weekend':('Is Weekend',agg_isval('No')),
16 'Orders Is Event':('Is Event',agg_isval('Yes')), 'Orders Not Event':('Is Event',agg_isval('No')),
17 'Orders ItemCat Fashion':('Item Category',agg_isval('Fashion')), 'Orders ItemCat Beauty':('Item Category',agg_isval('Beauty')), 'Orders ItemCat Electronics':('Item Category',agg_isval('Electronics')),
18 'Orders ItemCat Marketing':('Item Category',agg_isval('Marketing')), 'Orders ItemCat FinServices':('Item Category',agg_isval('Financial Services')), 'Orders ItemCat InfTechnology':('Item Category',agg_isval('Information Technology')),
19 'Orders ItemCat Facility':('Item Category',agg_isval('Facility')), 'Orders ItemCat Entertainment':('Item Category',agg_isval('Entertainment')), 'Orders ItemCat Supermarket':('Item Category',agg_isval('Supermarket')),
20 'Orders Avg Total Price':('Item Total Price',agg_rndavg()), 'Orders Category Product':('Order Category',agg_isval('Product')), 'Orders Category Service':('Order Category',agg_isval('Service')),
21 'Orders Urgency Low':('Order Urgency',agg_isval('Low')), 'Orders Urgency Medium':('Order Urgency',agg_isval('Medium')), 'Orders Urgency High':('Order Urgency',agg_isval('High')),
22 'Orders Delivery OnTime':('Order Delivery Status',agg_isval('On-Time')), 'Orders Delivery Late':('Order Delivery Status',agg_isval('Late')), 'Orders Delivery Rejected':('Order Delivery Status',agg_isval('Rejected'))]).reset_index()
23
24 # show
25 print(f"data[W_PROCEUREMENT_CL] : {data[W_PROCEUREMENT_CL].shape[0]} rows {data[W_PROCEUREMENT_CL].shape[1]} columns")
```

Finished

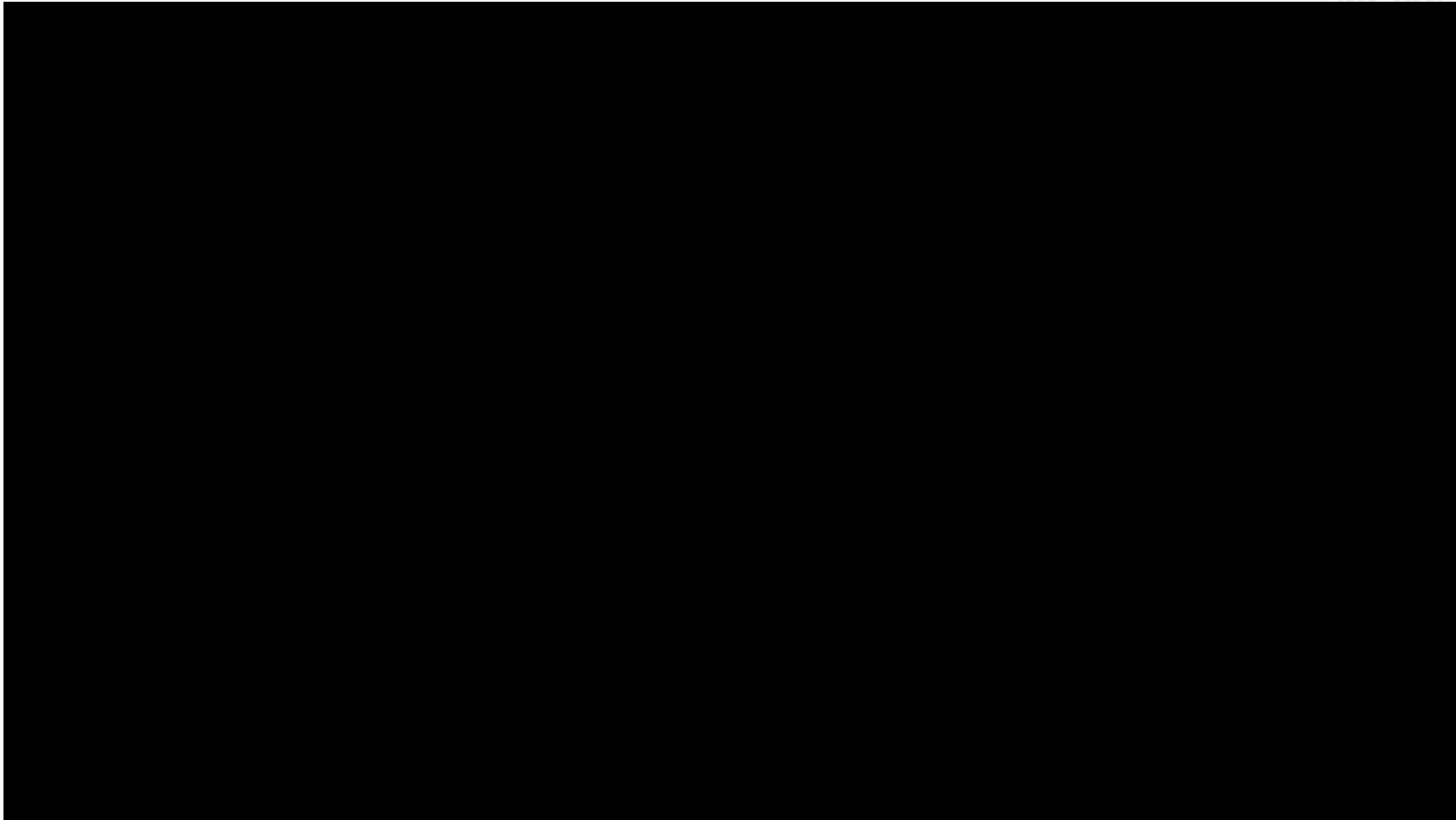
Took 0 secs. Last updated by OMLUSER1 at July 13 2023, 12:37:26 PM. (outdated)

data[W_PROCEUREMENT_CL] : 98 rows 28 columns

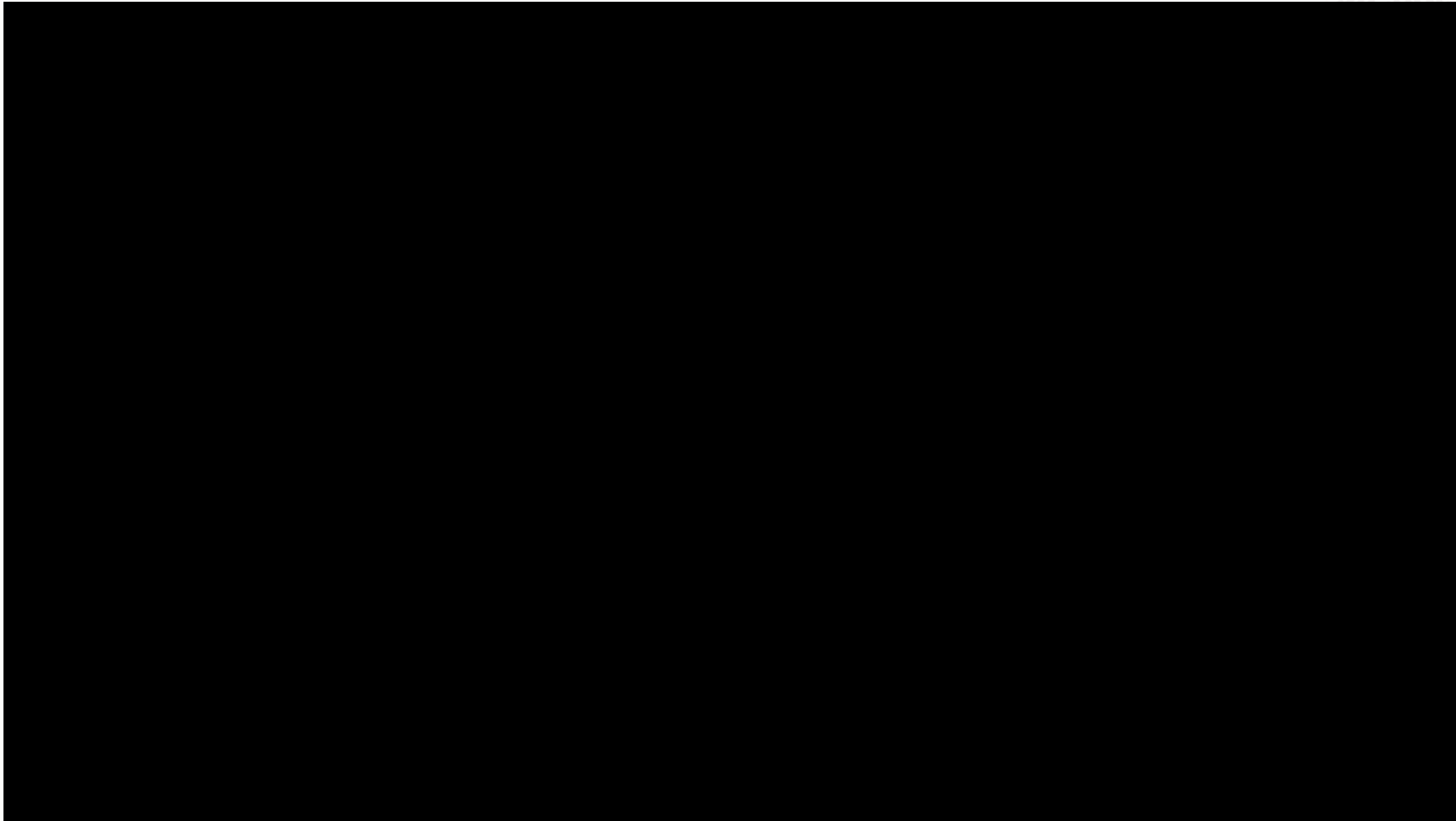
```
1 %python
2
3 # push data
4 try:
5     oml.drop(table="W_PROCEUREMENT_CL")
6 except:
7     pass
```

Finished

Supplier Segmentation - Modeling



Supplier Segmentation - Business Insights





Forecasting Procurement Spendings

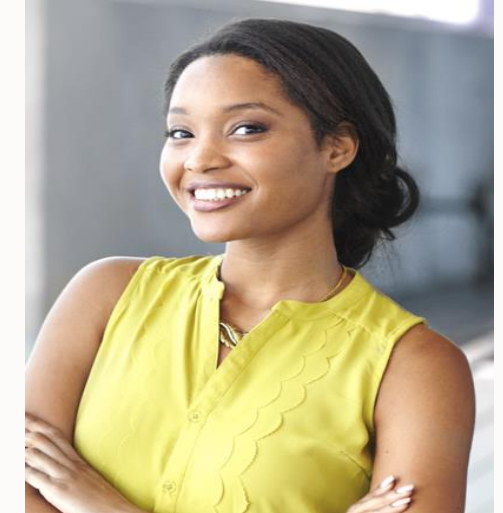
Forecasting Procurement Spendings



Procurement manager wants to predict procurement spendings for each category and day to take proactive actions about spendings. She asks data scientist to prepare a final data to be used in her citizen data scientist environment



Data scientist prepares final data for category-based procurement spending forecasting and shares with procurement manager.



Procurement manager builds spending forecasting models and discover expected spending patterns for the future

Forecasting Procurement Spendings - Data Preparation

ORACLE Machine Learning

OMLUSER1 Project
OMLUSER1 Workspace

OMLUSER1

08	109	299	58	350	92	59	59	68	39
88	105	283	40	348	83	62	47	57	39

Took 0 secs. Last updated by OMLUSER1 at July 13 2023, 12:37:42 PM. (outdated)

§ DATA PREPARATION (FORECASTING)

FINISHED

Took 0 secs. Last updated by OMLUSER1 at July 12 2023, 12:55:44 PM. (outdated)

```
1 %python
2
3 # imports
4 import warnings
5 import oml
6 import pandas
7 import numpy
8
9 # settings
10 warnings.filterwarnings('ignore')
11 pandas.set_option('display.width', 2000)
12 pandas.set_option('display.max_rows', 100)
```

READY

```
1 %python
2
3 # sync proxy objects
4 W_PROCEUREMENT_1 = oml.sync(table="W_PROCEUREMENT_1")
5
6 # pull data
7 data[W_PROCEUREMENT_1] = W_PROCEUREMENT_1.pull()
8 print(f"data[W_PROCEUREMENT_1] : {data[W_PROCEUREMENT_1].shape[0]} rows {data[W_PROCEUREMENT_1].shape[1]} columns")
```

READY

```
1 %python
2
3 # prepare data for forecasting
4 data[W_PROCEUREMENT_FC] = data[W_PROCEUREMENT_1][['Date', 'Item Category', 'Item Total Price']]
5 df_itemcategory = data[W_PROCEUREMENT_FC].groupby(['Date', 'Item Category']).agg({'Item Total Price': 'sum'}).reset_index()
6 df_itemcategory['Date'] = df_itemcategory['Date'].astype(str)
7 df_date = data[W_PROCEUREMENT_FC].groupby(['Date']).agg({'Item Total Price': 'sum'}).reset_index()
8 df_date['Date'] = df_date['Date'].astype(str)
9 data[W_PROCEUREMENT_FC] = pandas.merge(df_itemcategory, df_date.add_prefix('D '), left_on=['Date'], right_on=['D Date'], how='left')
10 data[W_PROCEUREMENT_FC] = data[W_PROCEUREMENT_FC].drop(columns=['D Date'])
11 data[W_PROCEUREMENT_FC] = data[W_PROCEUREMENT_FC].rename(columns={'Item Total Price': 'Category Total Price', 'D Item Total Price': 'Date Total Price'})
12 data[W_PROCEUREMENT_FC].insert(2, 'Date Total Price', data[W_PROCEUREMENT_FC].pop('Date Total Price'))
13
14 # check
15 print(f"data[W_PROCEUREMENT_FC] : {data[W_PROCEUREMENT_FC].shape[0]} rows {data[W_PROCEUREMENT_FC].shape[1]} columns")
```

FINISHED

data[W_PROCEUREMENT_FC] : 896 rows 4 columns

Took 0 secs. Last updated by OMLUSER1 at July 13 2023, 11:04:12 AM. (outdated)

```
1 %python
2
3 # push data
4 try:
5     oml.drop(table=W_PROCEUREMENT_FC)
6 except:
7     pass
8 W_PROCEUREMENT_FC = oml.create(data[W_PROCEUREMENT_FC], table=W_PROCEUREMENT_FC)
```

FINISHED

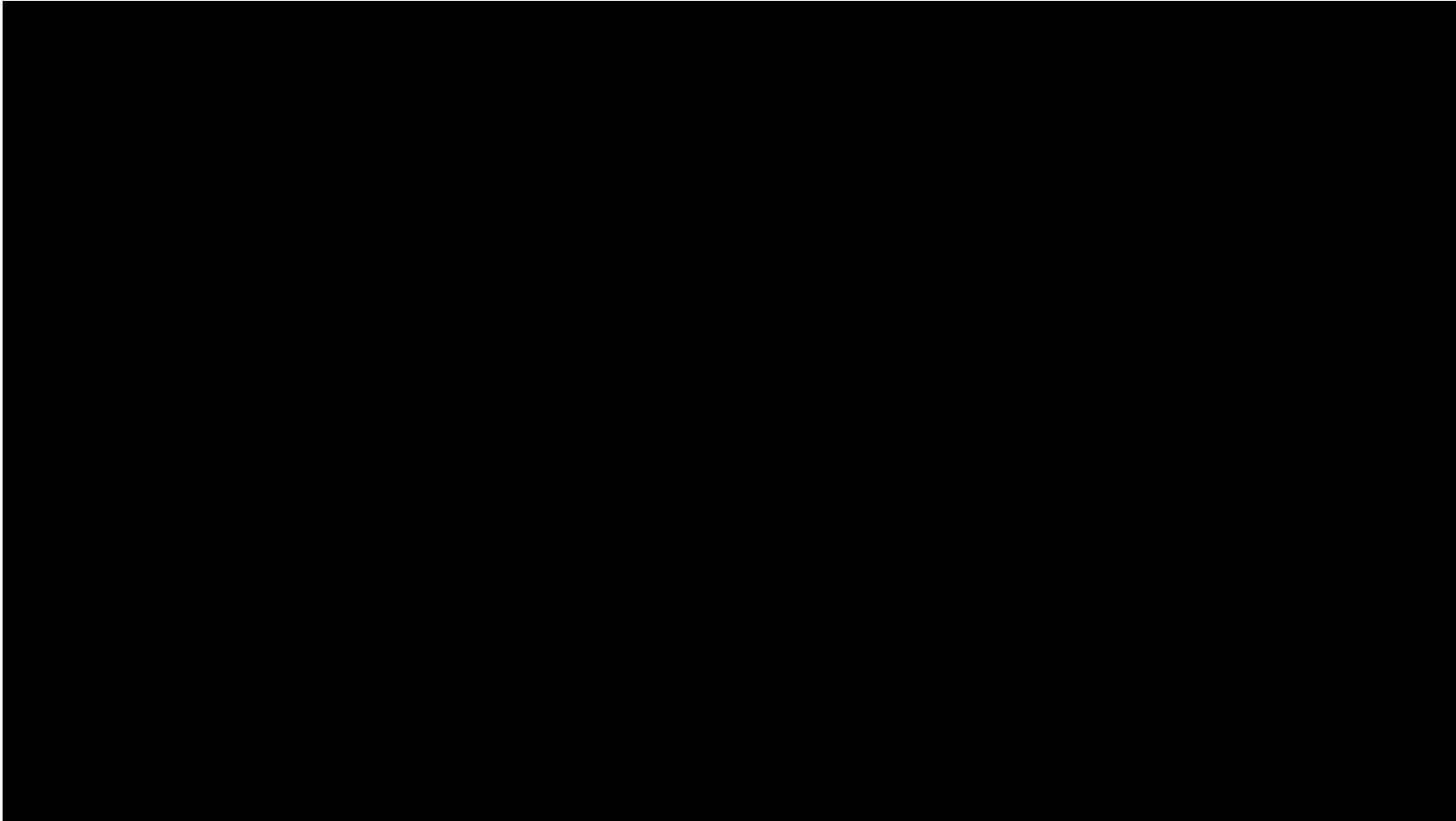
Took 1 sec. Last updated by OMLUSER1 at July 13 2023, 11:04:16 AM. (outdated)

```
1 %sql
```

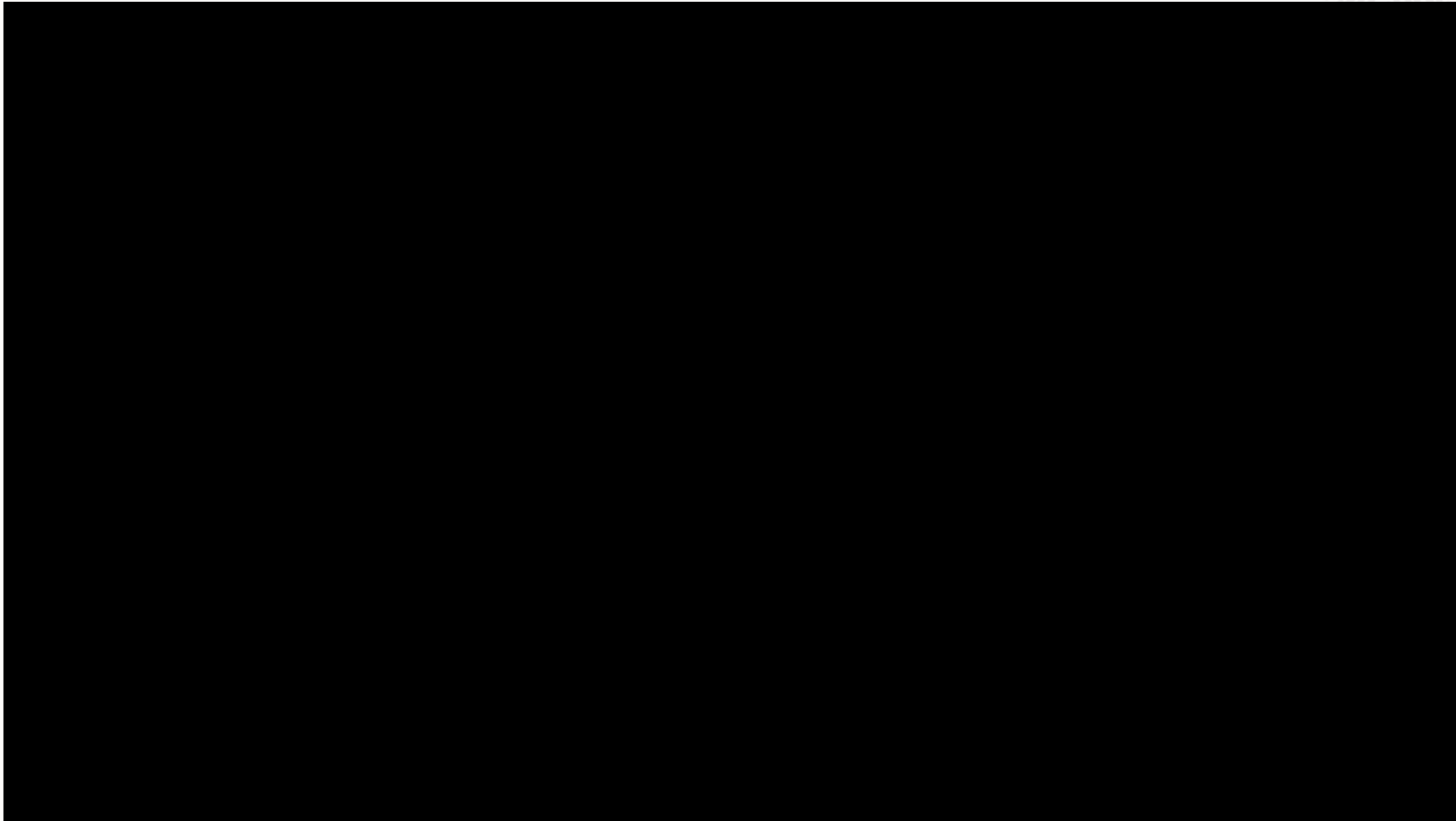
FINISHED



Forecasting Procurement Spendings - Modeling



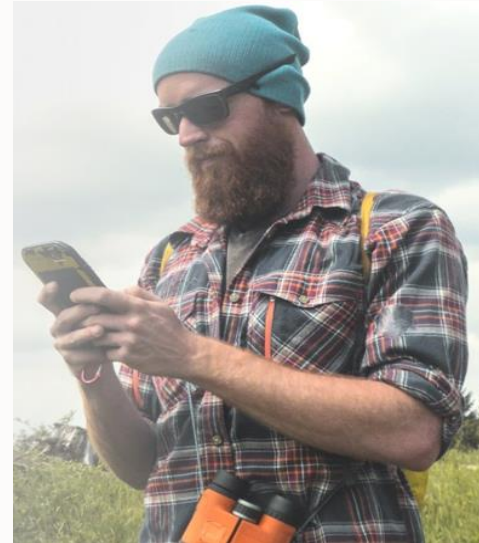
Forecasting Procurement Spendings - Business Insights



Summary

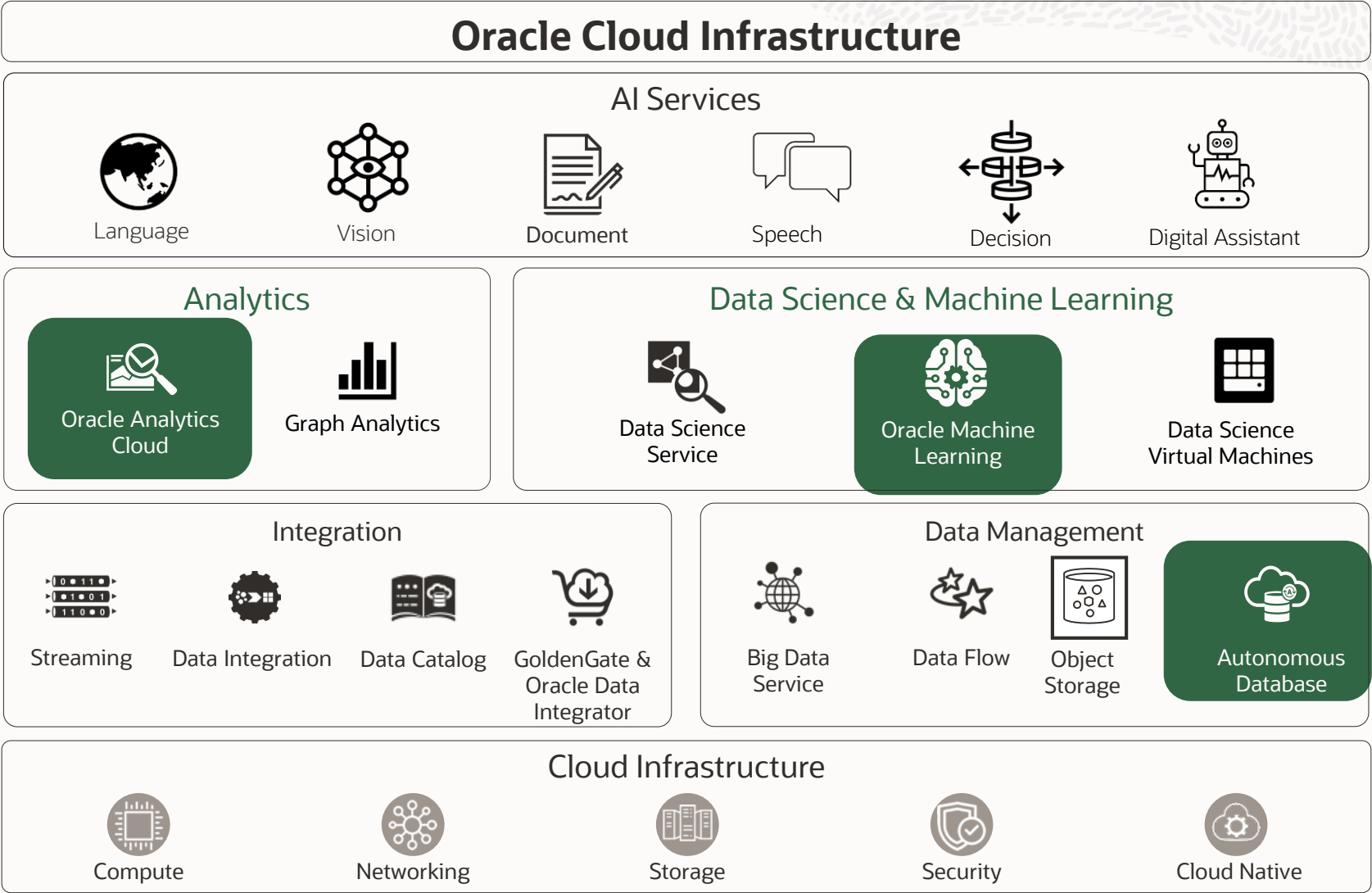


Procurement manager is happy to understand supplier behavior, spending anomalies and forecasting future procurement spendings



Data scientists is happy to have an end-to-end platform for advanced data science requirements

Behind the Scenes: Oracle Data Science Platform



ORACLE