Data Exploration Made Easy With Oracle-ADS!

Phil Godfrey – Principal Data Analytics & AI Consultant, Vertice

ORACLE

Helpful Links –

KAGGLE – TITANIC DATASET:
https://www.kaggle.com/competitions/titanic/data

ORACLE ADS –DOCUMENTATION:
https://accelerated-data-science.readthedocs.io/en/latest/index.html

ORACLE LIVE LAB – GETTING STARTED WITH OCI DATA SCIENCE:
https://apexapps.oracle.com/pls/apex/r/dbpm/livelabs/view-workshop?wid=673&clear=RR,180&session=17058798830297

ORACLE CLOUD FREE TIER:
https://www.oracle.com/cloud/free/

Future & Past TechCasts:

Click to see Live TechCast page

Jan 25th

Fusion Analytics EPM Cloud Connector Lessons Learned

Presented by **John Whitaker & Carlos Mendez**

Feb 8th

Getting industry data ready for sharing and AI

Presented by **Jason Duncan-Wilson**

TechCast Archive

2024 | 2023 | 2022 | 2021 | 2020 | 2019

Date	Title	Presenter(s)	Replay	Download(s)
Jan 12	A&D Summit Preview	Dan Vlamis, Abi Giles-Haigh, Cathye Pendley, Wayne Van Sluys, Jean Ihm, and Roger Cressey	Video	Slides

Submit a topic to share at **https://andouc.org/techcasts/**

Analytics and Data ORACLE USER COMMUNITY

www.andouc.org

Oracle Spatial & Graph SIG

Analytics and Data
ORACLE USER COMMUNITY

We Have Merch!

Show your “Tech Side” in everything you do!

Visit the AnDOUC Store at ANDOUC.ORG

Analytics and Data
ORACLE USER COMMUNITY

Let's Connect

Website
http://andouc.org/

Chat with the Experts
https://bit.ly/Join-ANDOUC-Slack

Watch Previous TechCasts
https://bit.ly/3qmGgHN

@AnalyticAndData

https://www.facebook.com/AnDOracleUserCommunity

https://www.linkedin.com/company/analytics-and-data-oracle-user-community

Spatial + Graph SIG
bit.ly/Spatial-Graph-LinkedIn

Analytics and Data
ORACLE USER COMMUNITY

Save the Date!

Analytics and Data Summit 2024

April 9-11, 2024
Oracle Conference Center
Redwood Shores, California

www.andouc.org/andsummit2024

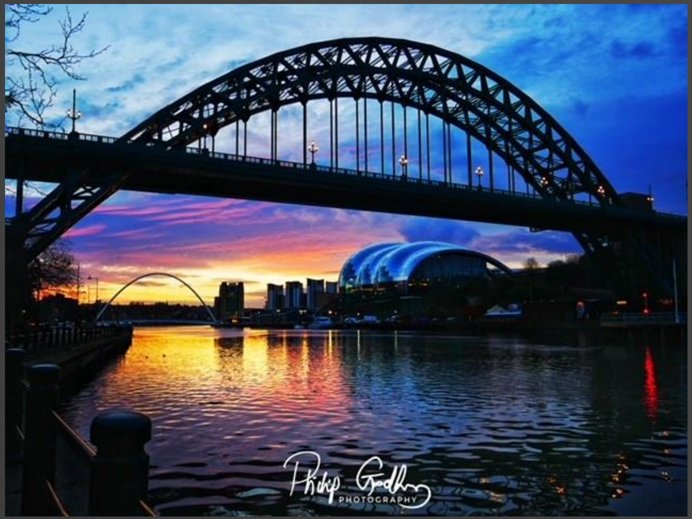
Data Exploration made easy with Oracle-ADS!

Phil Godfrey – Principal Data Analytics & AI Consultant

Introductions

Bio

- Principal Data Analytics & AI Consultant at Vertice.
- Over 10 years' experience working with data across various business sectors including finance, HR and healthcare.
- Data Science & Machine Learning experience across roles in Vertice and NHS Business Services Authority.
- Oracle ACE Associate.
- Outside of work I’m a keen photographer and set-up a photography business in January 2021.
- https://www.etsy.com/shop/pgodfreyphotography



Who We Are

Customer.
Data.
Solutions.

Oracle Cloud Managed Service Provider (CMSP)	**Oracle Cloud World Firsts in 2016, 2017 and 2022 on OCI**	Successive, Multiple **Oracle Global Excellence Awards**	Successfully Delivering Oracle Expert Services Since 2010
2023 Oracle Europe West Cloud/Tech Partner Award **Innovation**	Oracle UK & Ireland **Autonomous Database Partner of the Year 2020**	Successive, Multiple **Oracle UK & Ireland Partner of the Year Awards**	**Founded in 2010** **All** **Oracle Practitioners**

Engage the Customer. Enhance the Data. Enable the Solutions.

What We Do

Data Analytics & AI

DATA TRANSFORMATION

- Data Analytics
- New Data Platform
- Oracle Financial Services Analytical Applications
- Data Lakehouse
- Data Warehouse
- Data Mesh

DATA DIGITAL SERVICES

- AI, ML, Data Science
- Cloud Native
- Apps Modernisation Enablement
- DevOps and ML Ops

Vertice

ORACLE

Database & Infrastructure

DATA PLATFORM

- Cloud Strategy
- Multi-Cloud (OCI, Azure & AWS)
- Hybrid Deployment
- Cloud@Customer (ExaC@C)
- SaaS Integration

DATA MODERNISATION

- Technology Debt
- Client Modernisation
- Database Consolidation
- Cloud Back-up / DR
- *Cyber Security (Partner)*

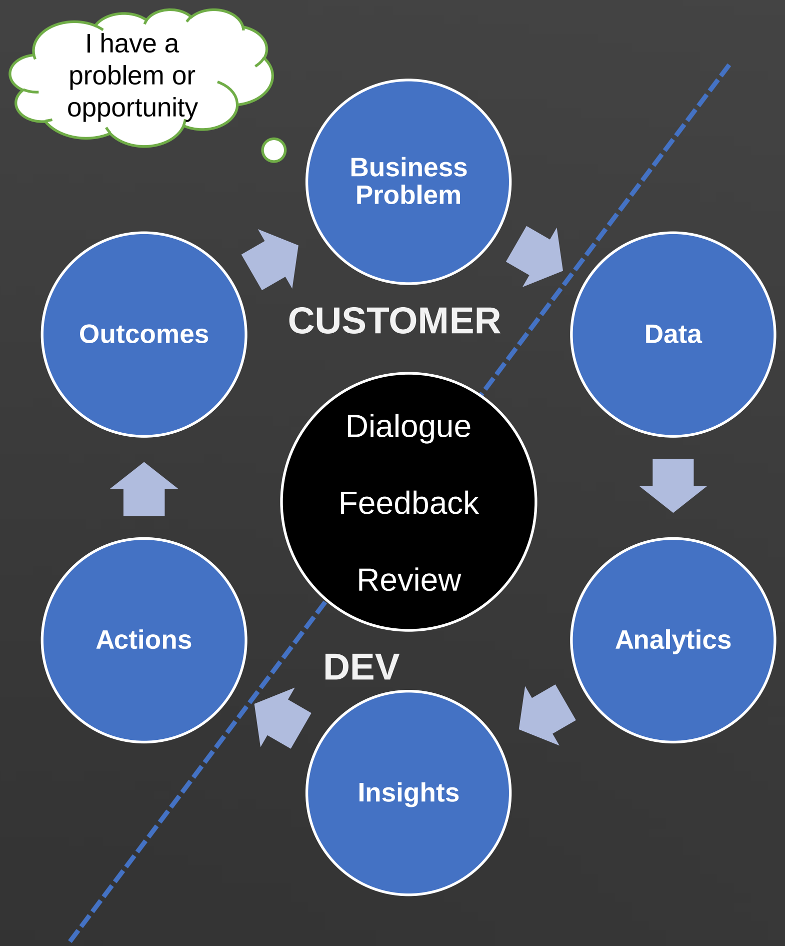
Aim

Work through Exploratory Data Analysis of Titanic data, showcase how Oracle-ADS (Accelerated Data Science SDK) can help!

How?

- Utilise Titanic data that is freely available.
- Perform some typical EDA, to help understand the dataset in detail.
- Show you how you can use Oracle-ADS to help ☺

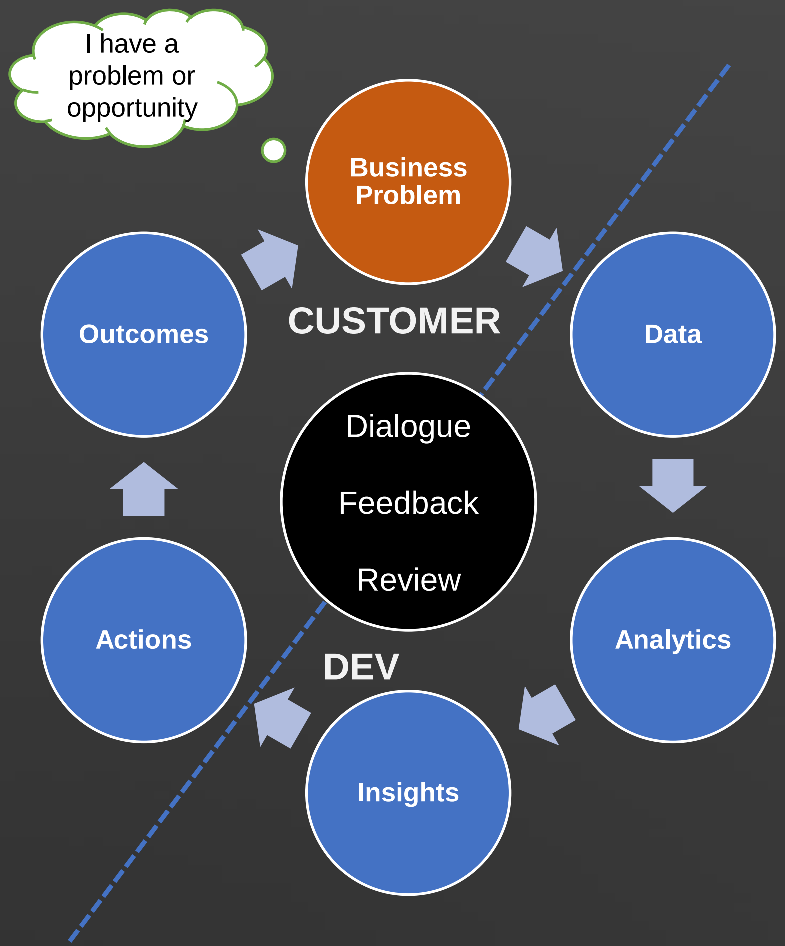
Data Science - CRISP-DM for Delivery



- The operating model must be flexible to respond to the needs of the customers

- Cross-industry standard process for data mining (CRISP-DM)
- Data mining is a process of extracting and discovering patterns in large data sets involving methods at the intersection of machine learning, statistics, and database systems

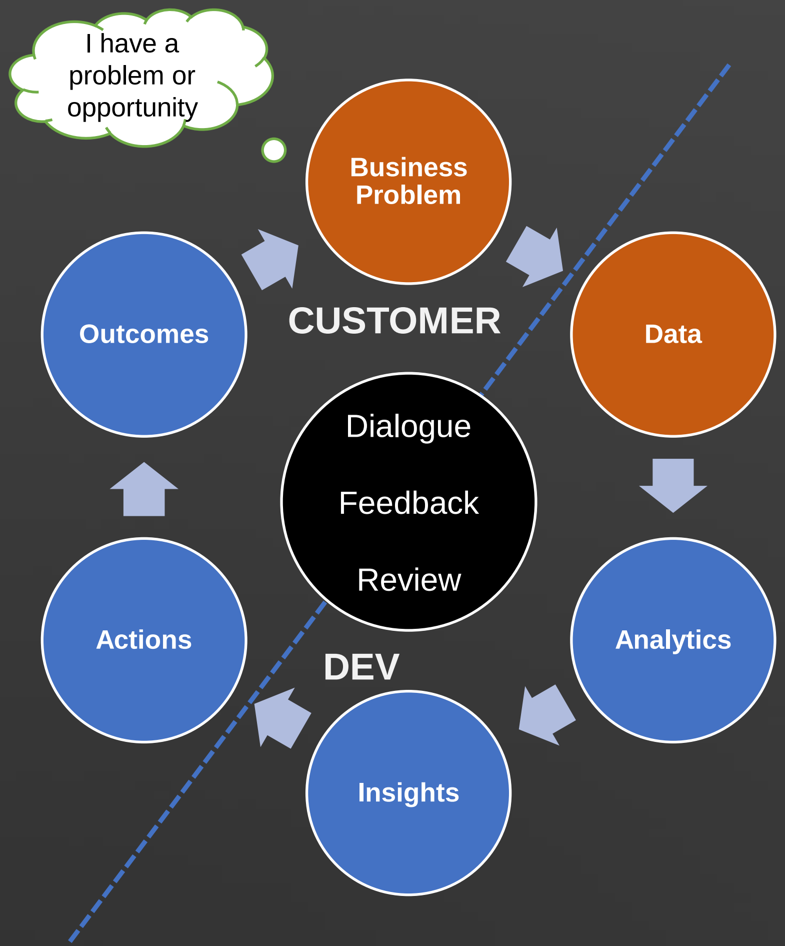
Data Science - CRISP-DM for Delivery



- The operating model must be flexible to respond to the needs of the customers

- Cross-industry standard process for data mining (CRISP-DM)
- Data mining is a process of extracting and discovering patterns in large data sets involving methods at the intersection of machine learning, statistics, and database systems

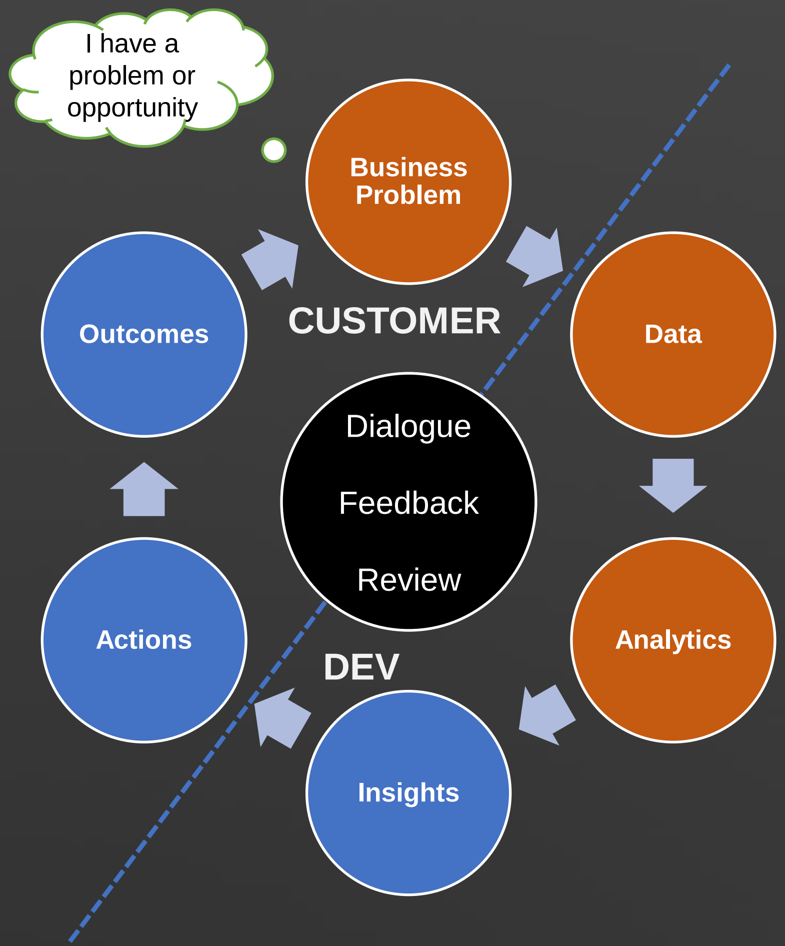
Data Science - CRISP-DM for Delivery



- The operating model must be flexible to respond to the needs of the customers

- Cross-industry standard process for data mining (CRISP-DM)
- Data mining is a process of extracting and discovering patterns in large data sets involving methods at the intersection of machine learning, statistics, and database systems

Data Science - CRISP-DM for Delivery



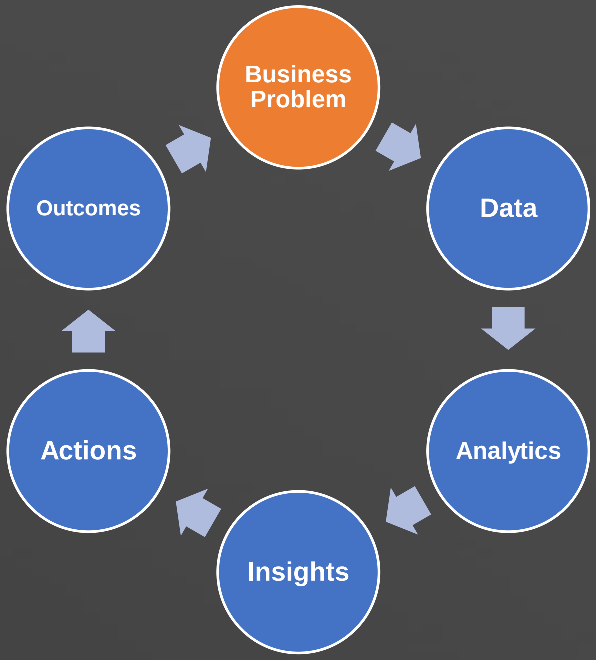
- The operating model must be flexible to respond to the needs of the customers

- Cross-industry standard process for data mining (CRISP-DM)
- Data mining is a process of extracting and discovering patterns in large data sets involving methods at the intersection of machine learning, statistics, and database systems

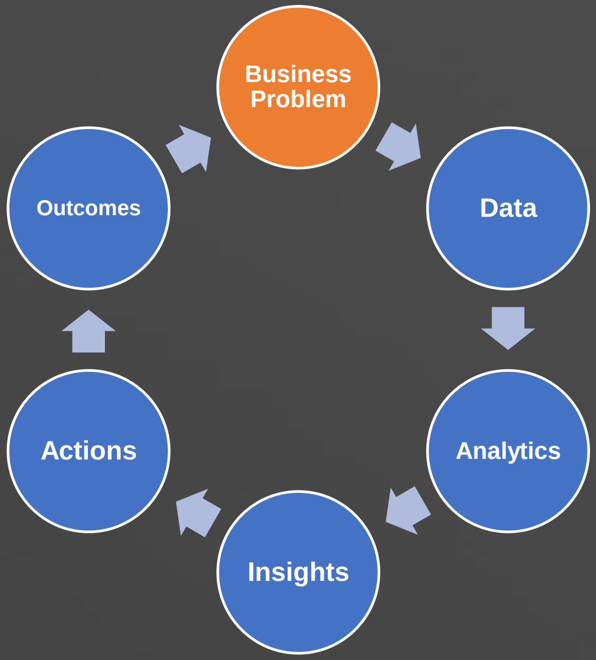
Business Problem

- Presented with a new dataset.

- As a Data Scientist / Analyst I need to explore and understand it:



Business Problem



- Presented with a new dataset.

- As a Data Scientist / Analyst I need to explore and understand it:

  What does the dataset look like?

  How many columns / rows does it have?

  What are the data types?

  Are there any missing values I need to worry about?

  What should I do with those?

  Etc…

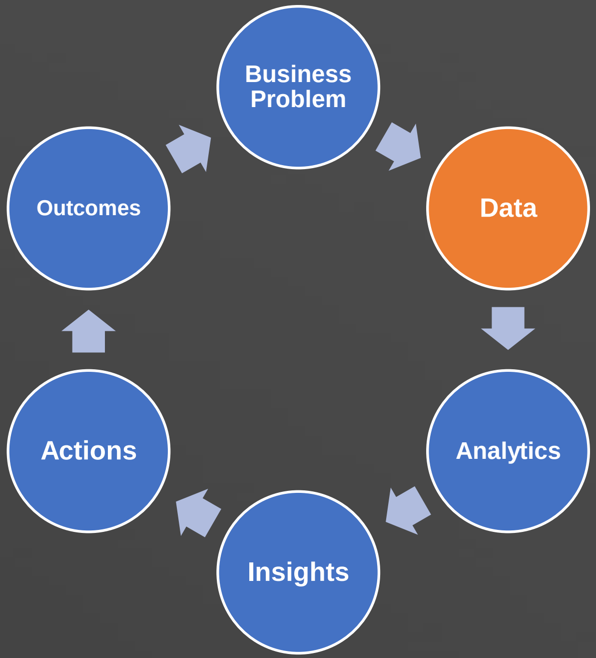
Data

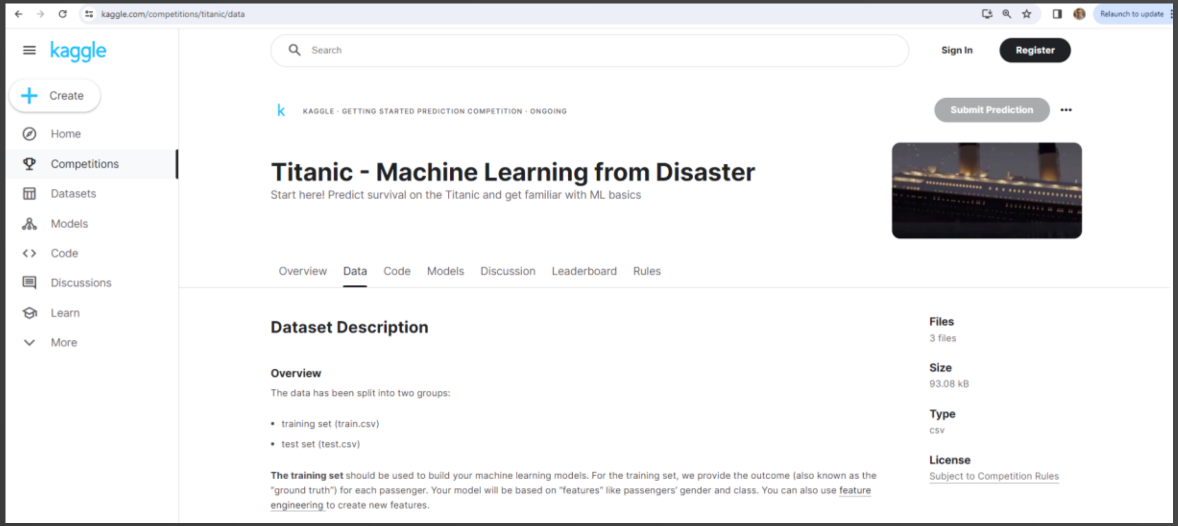
- Utilizing Titanic dataset that is available from [Kaggle](https://www.kaggle.com).
  - Well-known dataset.
  - We can focus on Oracle-ADS, rather than needing to understand a complex dataset.

- 1 file required:
  - Train.csv

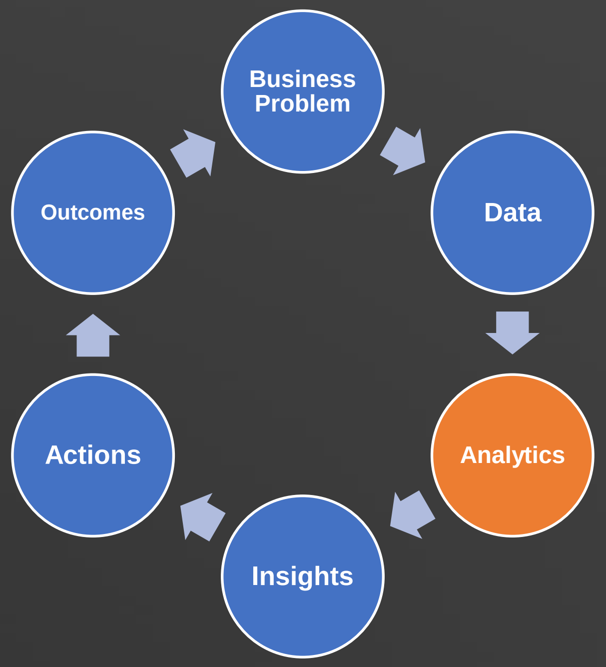
  Note (Test.csv is also available)

- Data can be downloaded as csv's, or you can utilise the Kaggle API.





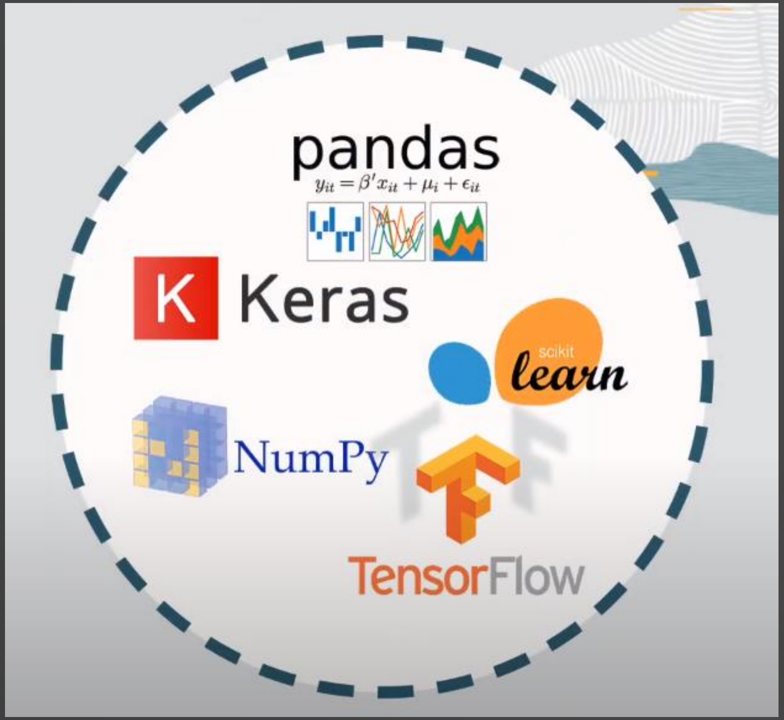
Exploratory Data Analysis in Oracle Data Science



Overview of the Oracle Data Science Service

- JupyterLab-based environment allows data scientists to experiment and develop models.

- Within a Jupyter lab you can:
  - Write code in Python
  - Access a variety of open-source libraries
  - **Oracle Accelerated Data Science** Python Library (ADS)



Oracle Accelerated Data Science (Oracle-ads)

- Oracle Accelerated Data Science SDK is a user-friendly Python toolkit that supports the data scientist through their entire end-to-end data science workflow.

- It speeds up common data science activities by providing tools that automate and simplify common data science tasks:

  - Model Deployment
  - Jobs
  - ML Pipelines
  - Data Flow
  - Object Storage
  - Vault
  - Autonomous Database.

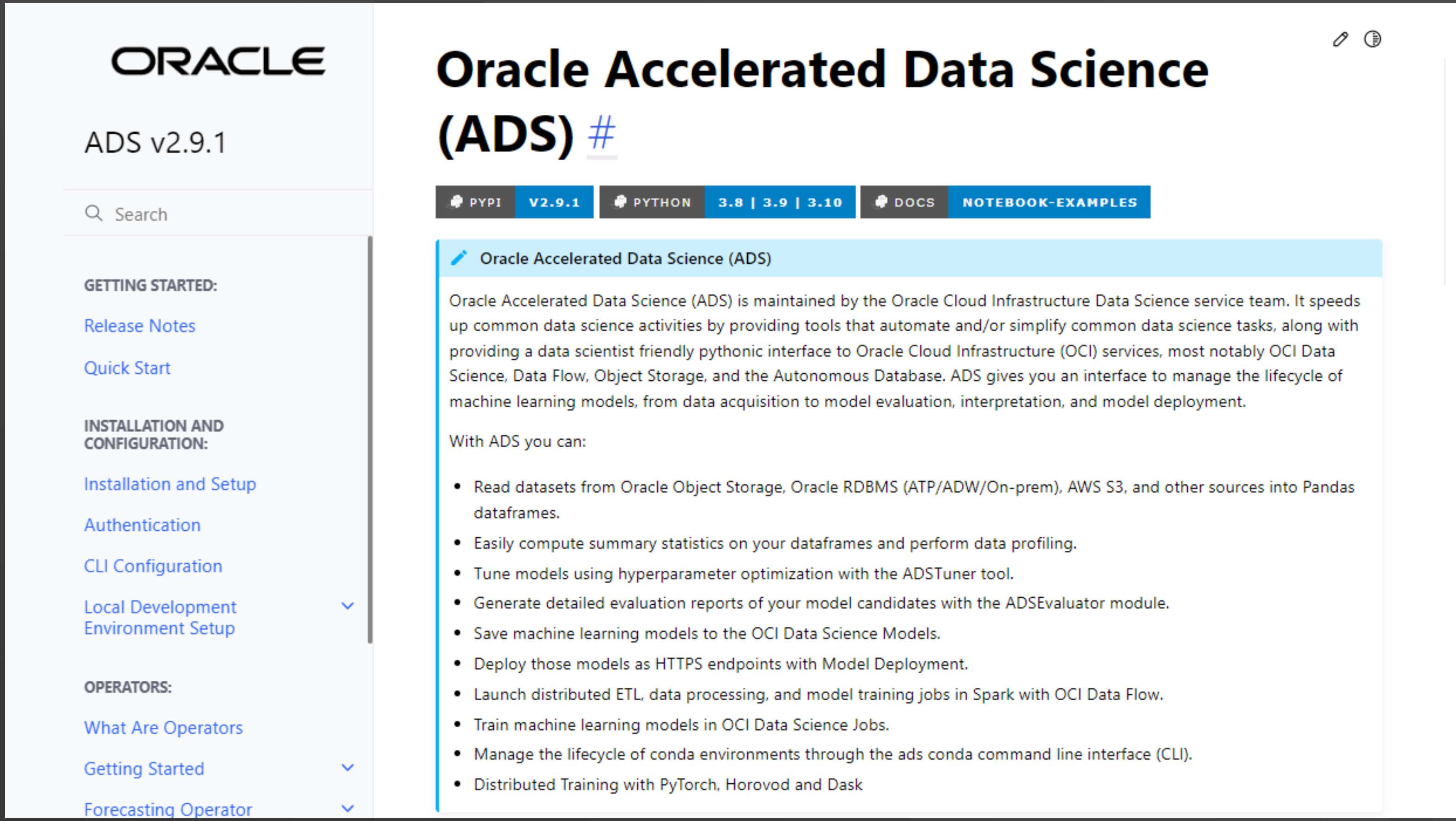
Code to install:

```
python3 -m pip install oracle-ads
```

- ADS gives you an interface to manage the life cycle of machine learning models, from data acquisition to model evaluation, interpretation, and model deployment.

Oracle Accelerated Data Science (Oracle-ads)

- If you're looking for any further information, you can access the documentation [here]



ORACLE

ADS v2.9.1

Search

GETTING STARTED:

Release Notes

Quick Start

INSTALLATION AND CONFIGURATION:

Installation and Setup

Authentication

CLI Configuration

Local Development Environment Setup

OPERATORS:

What Are Operators

Getting Started

Forecasting Operator

Oracle Accelerated Data Science (ADS)

PYPI V2.9.1 | PYTHON 3.8 | 3.9 | 3.10 | DOCS NOTEBOOK-EXAMPLES

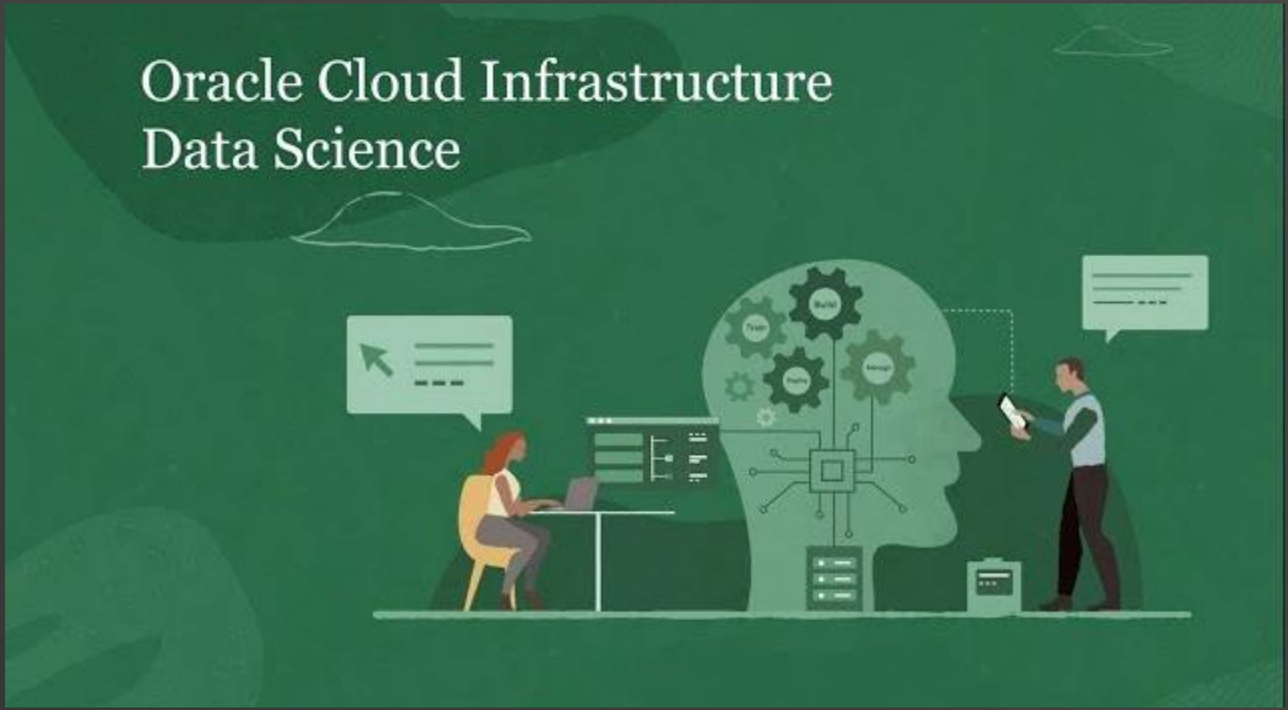
Oracle Accelerated Data Science (ADS)

Oracle Accelerated Data Science (ADS) is maintained by the Oracle Cloud Infrastructure Data Science service team. It speeds up common data science activities by providing tools that automate and/or simplify common data science tasks, along with providing a data scientist friendly pythonic interface to Oracle Cloud Infrastructure (OCI) services, most notably OCI Data Science, Data Flow, Object Storage, and the Autonomous Database. ADS gives you an interface to manage the lifecycle of machine learning models, from data acquisition to model evaluation, interpretation, and model deployment.

With ADS you can:

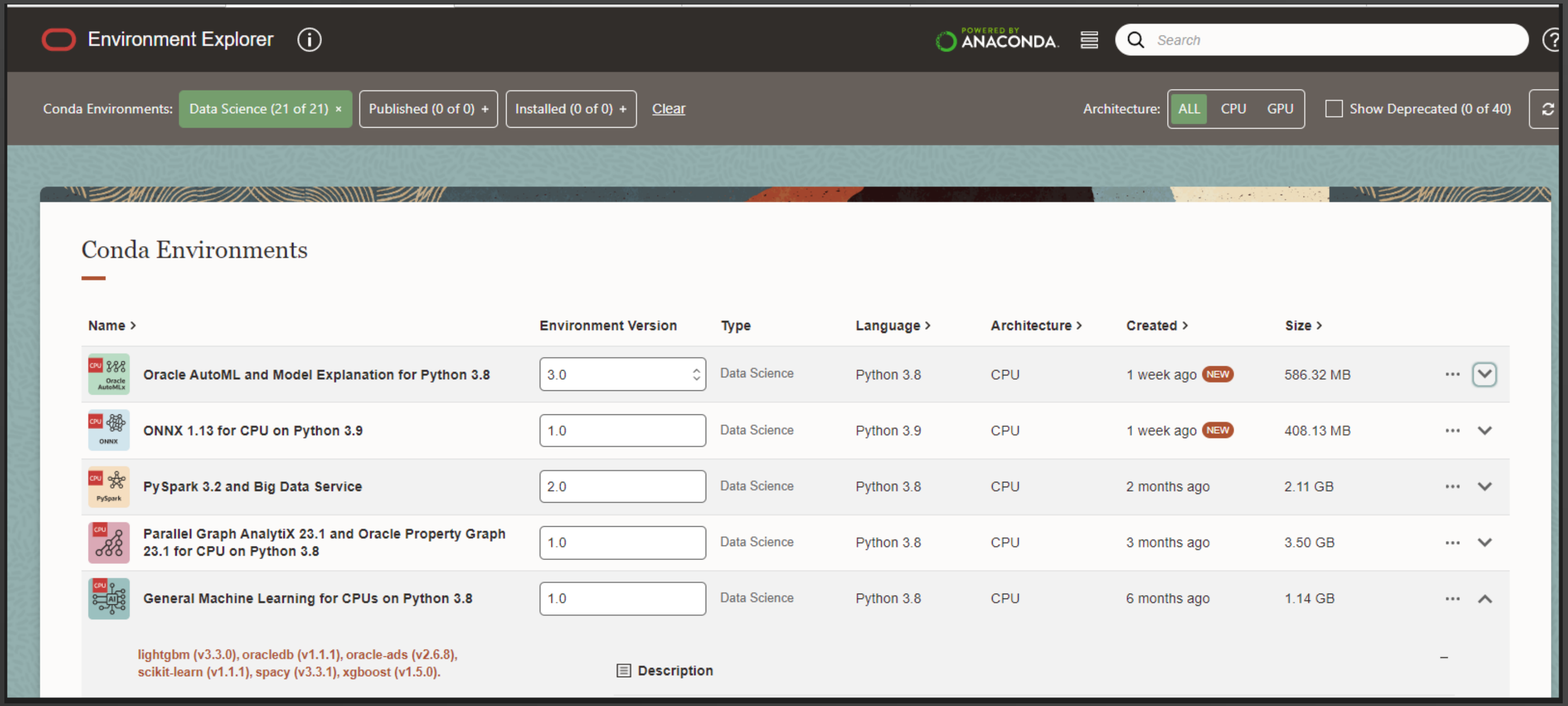
- Read datasets from Oracle Object Storage, Oracle RDBMS (ATP/ADW/On-prem), AWS S3, and other sources into Pandas dataframes.
- Easily compute summary statistics on your dataframes and perform data profiling.
- Tune models using hyperparameter optimization with the ADSTuner tool.
- Generate detailed evaluation reports of your model candidates with the ADSEvaluator module.
- Save machine learning models to the OCI Data Science Models.
- Deploy those models as HTTPS endpoints with Model Deployment.
- Launch distributed ETL, data processing, and model training jobs in Spark with OCI Data Flow.
- Train machine learning models in OCI Data Science Jobs.
- Manage the lifecycle of conda environments through the ads conda command line interface (CLI).
- Distributed Training with PyTorch, Horovod and Dask

Oracle Data Science Platform



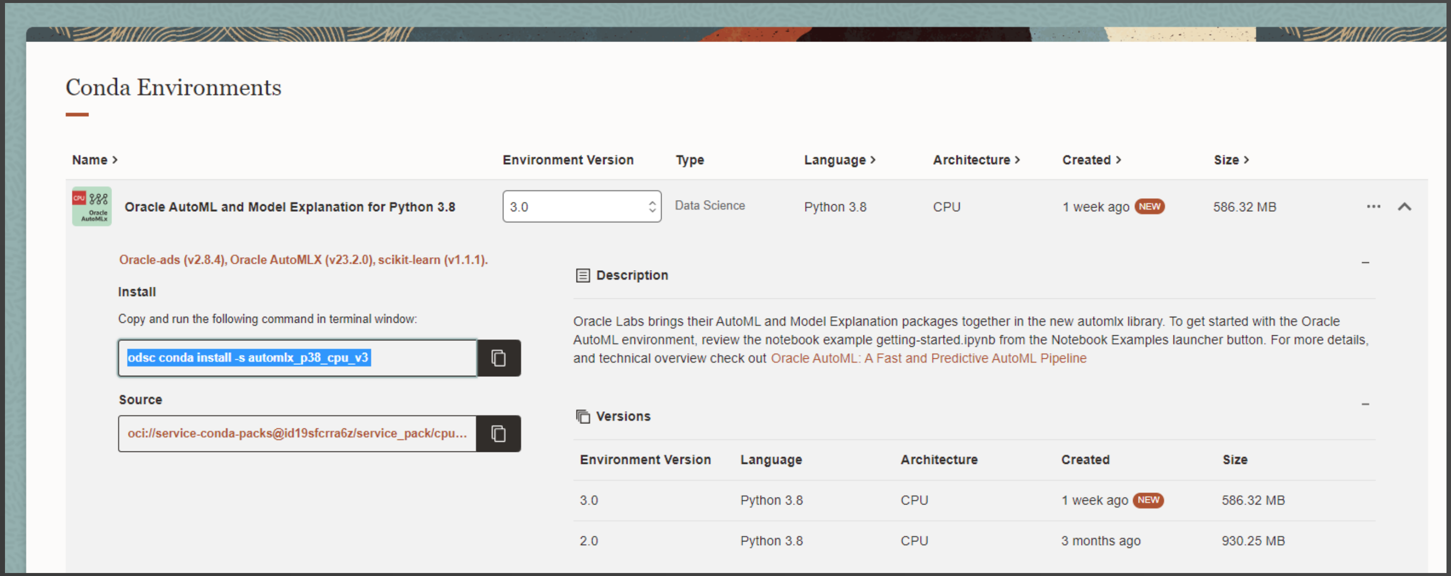
Install conda environment

- Navigate to "Environment Explorer" for a list of published conda environments.
- Published and updated by Oracle on a regular basis.



Install conda environment

- Navigate to “Terminal" and paste in the command (right).
- This will install the conda environment for us to use.




```
(base) bash-4.2 odsc conda install -s generalml_p38_cpu_v1
Environment slug: generalml_p38_cpu_v1
INFO:ODSC:Downloading conda pack generalml_p38_cpu_v1...
INFO:ODSC:Writing to /home/datascience/.generalml_p38_cpu_v1.tar.gz
Downloading pack generalml_p38_cpu_v1: 100%|                                        | 1.14G/1.14G [00:16<00:00, 74.4MB/s]
INFO:ODSC:download complete
INFO:ODSC:Extracting conda environment `/home/datascience/.generalml_p38_cpu_v1.tar.gz`
INFO:ODSC:Running conda-unpack script.
INFO:ODSC:Downloading Notebooks for the pack: General Machine Learning for CPUs on Python 3.8
INFO:ODSC:Conda environment has been successfully installed.
Removing: /home/datascience/.generalml_p38_cpu_v1.tar.gz
The environment setup is complete. To activate, run `conda activate /home/datascience/conda/generalml_p38_cpu_v1` in your terminal. It may take a few seconds for the kernel to appear in the JupyterLab launcher. To change the descript
ion of the environment, update /home/datascience/conda/generalml_p38_cpu_v1/_manifest.yaml.
(base) bash-4.2 conda activate /home/datascience/conda/generalml_p38_cpu_v1
```

Exploratory Data Analysis

- We need to import any relevant packages we want to use.
- Import our Titanic training data from a csv using pandas.

Import Packages

```
import pandas as pd
import ads
```

Load Data (csv file from Kaggle using `train.csv`

```
# Import data
titanic_df = pd.read_csv('../Data/train.csv')
```

Exploratory Data Analysis

[12]:
```
# Overview (head)
titanic_df.head()
```

[12]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN	S
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85	C
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	S
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123	S
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	NaN	S

[13]:
```
# Overview (tail)
titanic_df.tail()
```

[13]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
886	887	0	2	Montvila, Rev. Juozas	male	27.0	0	0	211536	13.00	NaN	S
887	888	1	1	Graham, Miss. Margaret Edith	female	19.0	0	0	112053	30.00	B42	S
888	889	0	3	Johnston, Miss. Catherine Helen "Carrie"	female	NaN	1	2	W./C. 6607	23.45	NaN	S
889	890	1	1	Behr, Mr. Karl Howell	male	26.0	0	0	111369	30.00	C148	C
890	891	0	3	Dooley, Mr. Patrick	male	32.0	0	0	370376	7.75	NaN	Q

Exploratory Data Analysis

Check descriptive statistics of Titanic data

[11]:
```
# Descriptive statistics
titanic_df.describe()
```

[11]:

	PassengerId	Survived	Pclass	Age	SibSp	Parch	Fare
count	891.000000	891.000000	891.000000	714.000000	891.000000	891.000000	891.000000
mean	446.000000	0.383838	2.308642	29.699118	0.523008	0.381594	32.204208
std	257.353842	0.486592	0.836071	14.526497	1.102743	0.806057	49.693429
min	1.000000	0.000000	1.000000	0.420000	0.000000	0.000000	0.000000
25%	223.500000	0.000000	2.000000	20.125000	0.000000	0.000000	7.910400
50%	446.000000	0.000000	3.000000	28.000000	0.000000	0.000000	14.454200
75%	668.500000	1.000000	3.000000	38.000000	1.000000	0.000000	31.000000
max	891.000000	1.000000	3.000000	80.000000	8.000000	6.000000	512.329200

This gives us a few early insights to the data already, which is why its so important to start here.

- Total samples are 891 or 40% of the actual number of passengers on board the Titanic (2,224).
- Survived is a categorical feature with 0 or 1 values.
- Around 38% samples survived, which is representative of the actual survival rate at 32%.

Exploratory Data Analysis

Check for missing values

We know which fields we have in the dataset, and the size of the dataset, but it's important to consider any missing data. We can do this using a function we've created **draw_missing_data_table**

[14]:
```python
# function
def draw_missing_data_table(df):
    total=titanic_df.isnull().sum().sort_values(ascending=False)
    percent=(titanic_df.isnull().sum()/df.isnull().count()).sort_values(ascending=False)*100
    missing_data=pd.concat([total,percent],axis=1,keys=['Total','Percent'])
    return missing_data
```

[15]:
```python
# Analyse missing data
draw_missing_data_table(titanic_df)
```

[15]:

	Total	Percent
Cabin	687	77.104377
Age	177	19.865320
Embarked	2	0.224467
PassengerId	0	0.000000
Survived	0	0.000000
Pclass	0	0.000000
Name	0	0.000000
Sex	0	0.000000
SibSp	0	0.000000
Parch	0	0.000000
Ticket	0	0.000000
Fare	0	0.000000

- We can see that **Age** has 177 missing values in our Train data.
- For other values we may want to impute missing values, but for missing ages, we'll leave them as blank.
- **Cabin** has over 3/4 of values missing, so we could **drop** this variable from our Train data.

Exploratory Data Analysis

- As well as the .describe() function in previous slide, **Pandas** also includes groupby operators.

Function	Description
count	Number of non-null observations
sum	Sum of values
mean	Mean of values
mad	Mean absolute deviation
median	Arithmetic median of values
min	Minimum
max	Maximum
mode	Mode

```
[21]: # Sum by sex
      titanic_df.groupby('Sex').sum()
```

[21]:

Sex	PassengerId	Survived	Pclass
female	135343	233	678
male	262043	109	1379

```
[23]: # Count by sex
      titanic_df.groupby('Sex').count()
```

[23]:

Sex	PassengerId	Survived	Pclass	Name	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
female	314	314	314	314	261	314	314	314	314	97	312
male	577	577	577	577	453	577	577	577	577	107	577

Exploratory Data Analysis

- As well as the .describe() function in previous slide, **Pandas** also includes groupby operators.

- We can pass in multiple variables into the group by, e.g., **Survived / PassengerId**.

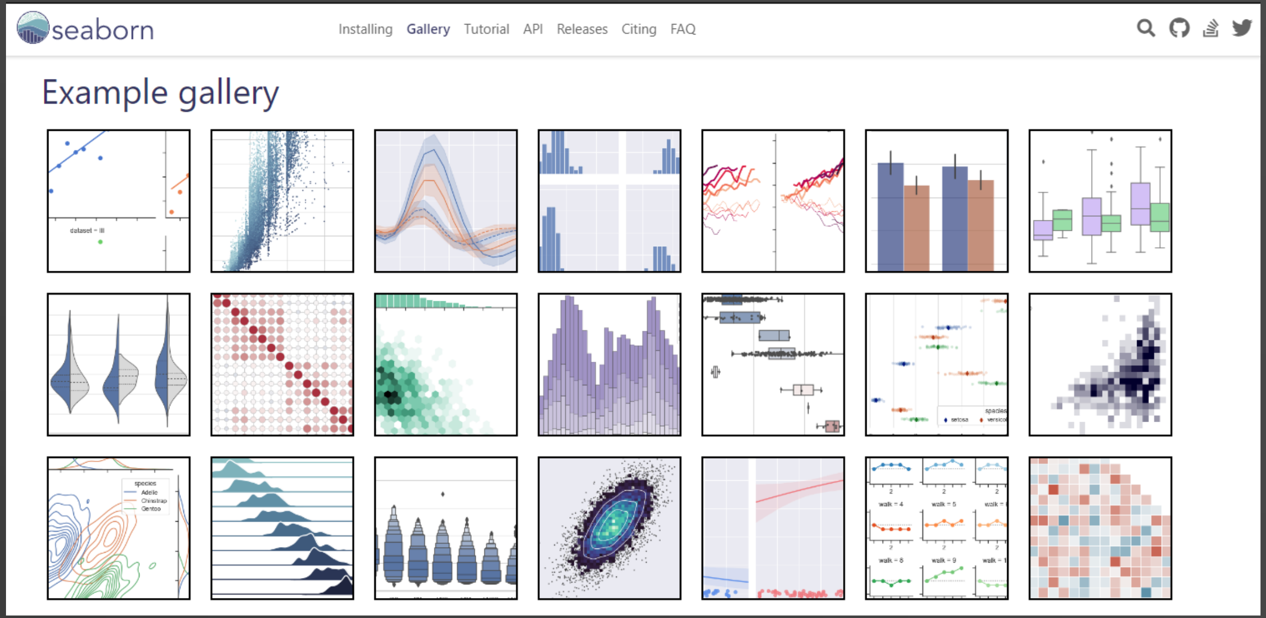
```
[24]: # count Passenger ID group by Survived
      titanic_df.groupby('Survived')['PassengerId'].count()

[24]: Survived
      0    549
      1    342
      Name: PassengerId, dtype: int64
```

Function	Description
count	Number of non-null observations
sum	Sum of values
mean	Mean of values
mad	Mean absolute deviation
median	Arithmetic median of values
min	Minimum
max	Maximum
mode	Mode

Exploratory Data Analysis – Visualisations

- The next step of our data exploration would be to **visualize the information**.
- It can often be that visualizations can return additional insights.
- We'll use 2 libraries for this:
  - **pyplot (from matplotlib)**
  - **seaborn**

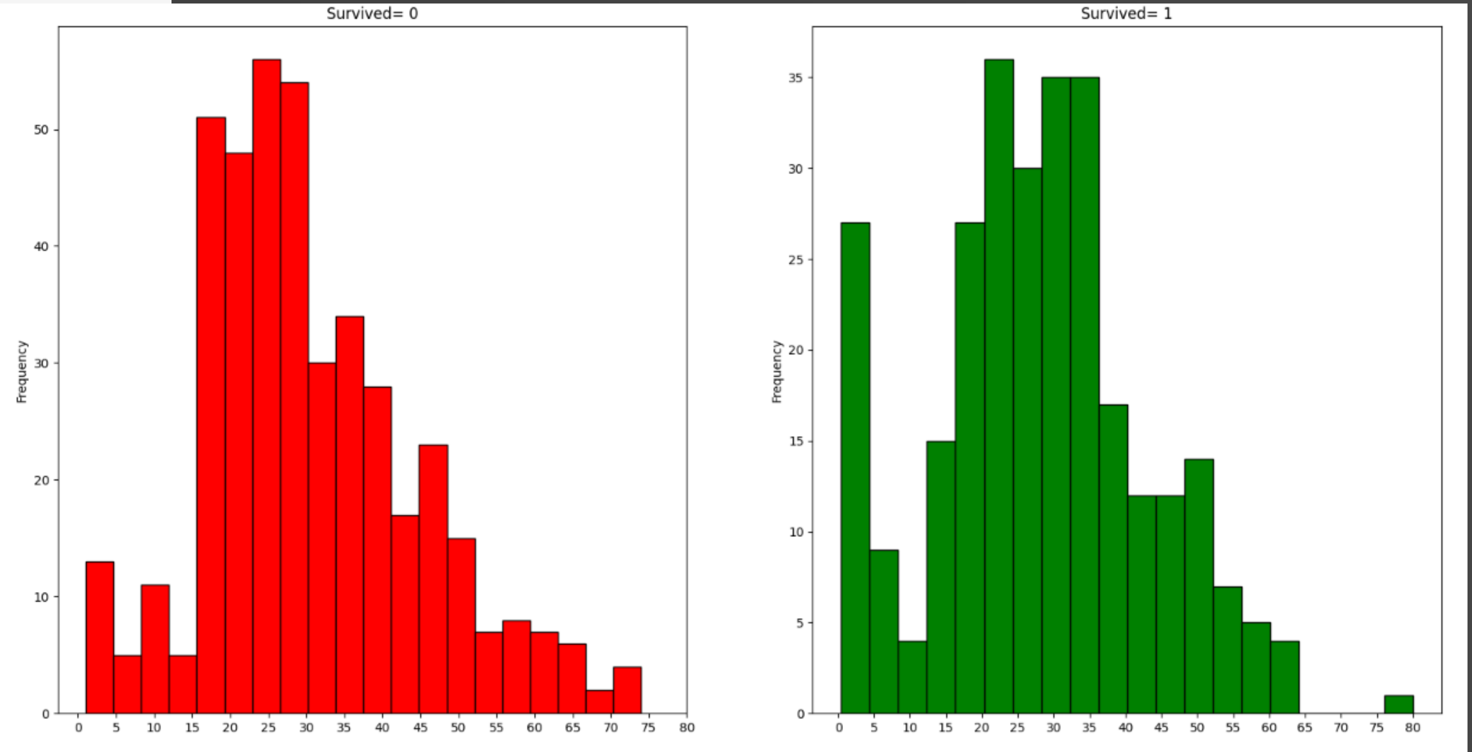
Both popular packages for visualizations in Python




```
import matplotlib.pyplot as plt
import seaborn as sns
```

Exploratory Data Analysis – Visualisations

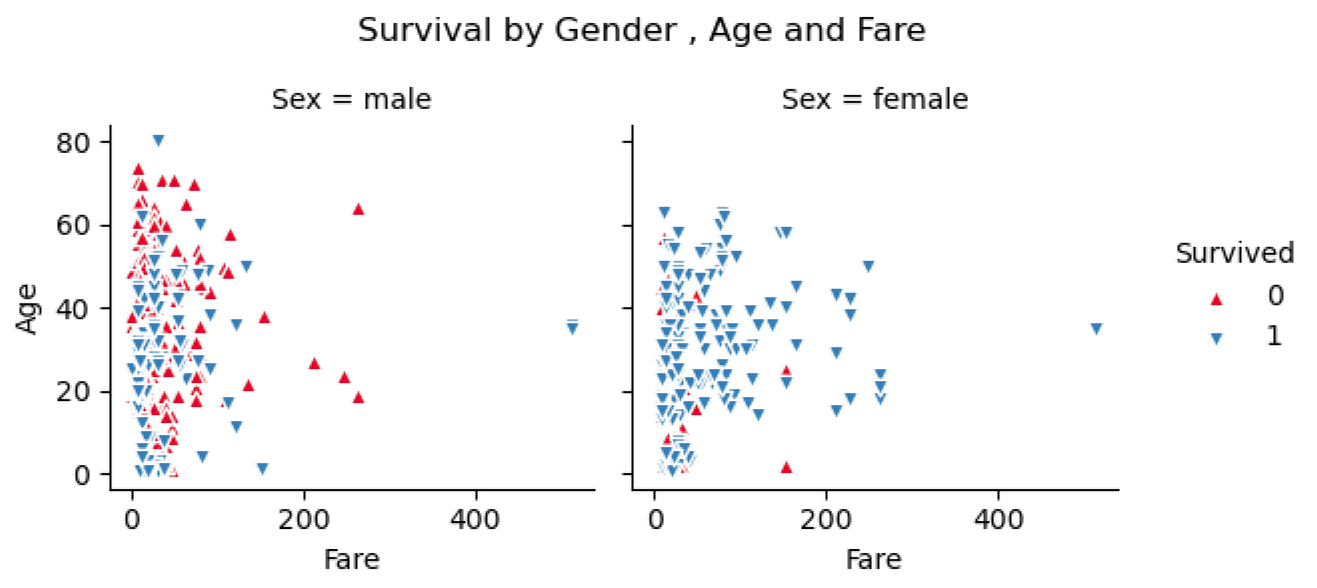
```
# Plot Frequency of those who Survived by Age
f,ax=plt.subplots(1,2,figsize=(20,10))

titanic_df[titanic_df['Survived']==0].Age.plot.hist(ax=ax[0],bins=20,edgecolor='black',color='red')
ax[0].set_title('Survived= 0')
x1=list(range(0,85,5))
ax[0].set_xticks(x1)

titanic_df[titanic_df['Survived']==1].Age.plot.hist(ax=ax[1],color='green',bins=20,edgecolor='black')
ax[1].set_title('Survived= 1')
x2=list(range(0,85,5))
ax[1].set_xticks(x2)
plt.show()
```



Exploratory Data Analysis – Visualisations

- Facet Grid to show multiple plots in a single cell – often very useful when comparing attributes.
- There are varying types of plots we can use, such as **scatterplots**.
- These are very customizable, We can also set Palletes, Margin Titles, Legends and Subtitles

```
g = sns.FacetGrid(
    titanic_df,
    hue="Survived",
    col="Sex",
    margin_titles=True,
    palette="Set1",
    hue_kws=dict(marker=["^", "v"]))
g.map(plt.scatter, "Fare", "Age",edgecolor="w").add_legend()
plt.subplots_adjust(top=0.8)
g.fig.suptitle('Survival by Gender , Age and Fare');
```



Exploratory Data Analysis – Visualisations

- There's lots of code written, just to perform some basic EDA.

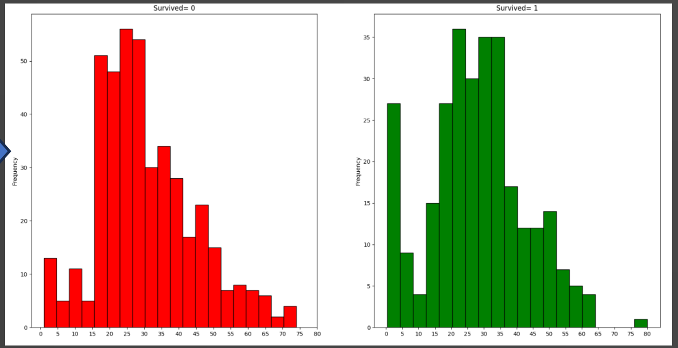
- There are many pieces of code I could write and throw away as part of this process.

```python
# function
def draw_missing_data_table(df):
    total=titanic_df.isnull().sum().sort_values(ascending=False)
    percent=(titanic_df.isnull().sum()/df.isnull().count()).sort_values(ascending=False)*100
    missing_data=pd.concat([total,percent],axis=1,keys=['Total','Percent'])
    return missing_data
```

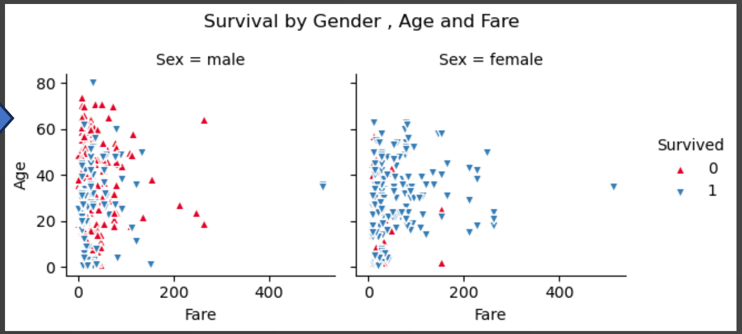
[15]:

	Total	Percent
Cabin	687	77.104377
Age	177	19.865320
Embarked	2	0.224467
PassengerId	0	0.000000

```python
# Plot Frequency of those who Survived by Age
f,ax=plt.subplots(1,2,figsize=(20,10))

titanic_df[titanic_df['Survived']==0].Age.plot.hist(ax=ax[0],bins=20,edgecolor='black',color='red')
ax[0].set_title('Survived= 0')
x1=list(range(0,85,5))
ax[0].set_xticks(x1)

titanic_df[titanic_df['Survived']==1].Age.plot.hist(ax=ax[1],color='green',bins=20,edgecolor='black')
ax[1].set_title('Survived= 1')
x2=list(range(0,85,5))
ax[1].set_xticks(x2)
plt.show()
```



```python
g = sns.FacetGrid(
    titanic_df,
    hue="Survived",
    col="Sex",
    margin_titles=True,
    palette="Set1",
    hue_kws=dict(marker=["^", "v"]))
g.map(plt.scatter, "Fare", "Age",edgecolor="w").add_legend()
plt.subplots_adjust(top=0.8)
g.fig.suptitle('Survival by Gender , Age and Fare');
```



Exploratory Data Analysis – Oracle-ADS

- By nature, exploratory data analysis can be very time consuming.

- There are some pre-packaged functions within Oracle-ADS that can help.

Oracle-ADS – Show_in_notebook

- Oracle ADS show_in_notebook method creates a preview of all the basic information about the data set.

```
[5]: # Import libraries
     import ads
     from ads.dataset.factory import DatasetFactory

[6]: # Convert the data set to an ADSDataset requried for "show_in_notebook" function
     titanic_ds = DatasetFactory.open(titanic_df, target="Survived").set_positive_class(1)
```

Oracle-ADS – **Show_in_notebook**

- Oracle ADS show_in_notebook method creates a preview of all the basic information about the data set.

```
[5]: # Import libraries
     import ads
     from ads.dataset.factory import DatasetFactory

[6]: # Convert the data set to an ADSDataset requried fo
     titanic_ds = DatasetFactory.open(titanic_df, ta
```

Loading a dataset with DatasetFactory *can be slower* than simply reading the same dataset with Pandas.

Added data visualizations and data profiling benefits of the ADSDataset object.

File Edit View Run Kernel Git Tabs Settings Help

Oracle-ADS exploration.ipynb

Code

Python [conda env:generalml_p

Oracle-ADS

Oracle ADS `show_in_notebook` method creates a preview of all the basic information about the data set.

It gives a great overview the data, number of rows and columns, data types/feature types of each column, visualisations of each column, correlations, and warnings about columns. These warnings are things like sparsley populated, or highly skewe columns for example.

```
[ ]: # Import libraries
import ads
from ads.dataset.factory import DatasetFactory
```

```
[ ]: # Convert the data set to an ADSDataset requried for "show_in_notebook" function
titanic_ds = DatasetFactory.open(titanic_df, target="Survived").set_positive_class(1)
```

```
[ ]: titanic_ds.show_in_notebook()
```

0 2 Python [conda env:generalml_p38_cpu_v1] | Idle Saving completed Mode: Command Ln 2, Col 38 Oracle-ADS exp

Oracle-ADS – Show_in_notebook

Excellent summary & overview of the data

Show_in_notebook

Correlations

Data Types

Feature Types

Vizualisations

Warnings

Number of rows / columns

Oracle-ADS – Suggest_recommendations

- Oracle-ADS isn't just limited to `show_in_notebook` feature.
- Oracle ADS has built-in functions to help with data cleaning, using the `suggest_recommendations` function.
- Runs in one line of code.

Oracle-ADS exploration.ipynb

Code

Python [conda env

7 WARNING(S) found

`Age` has 177.0 (19.9%) missing values. Consider remove the column or replace null values.	missing
`Cabin` has 687.0 (77.1%) missing values. Consider remove the column or replace null values.	missing
`Name` has a high cardinality: every value is distinct	high-cardinality
`Ticket` has a high cardinality: 681 distinct values	high-cardinality
`Cabin` has a high cardinality: 148 distinct values	high-cardinality
`SibSp` has 608 (68.24%) zeros)	zeros
`Parch` has 678 (76.09%) zeros)	zeros

[ ]:
```
titanic_ds.suggest_recommendations()
```

2 Python [conda env:generalml_p38_cpu_v1] | Idle Saving completed Mode: Edit Ln 1, Col 37

Oracle-ADS

Show_in_notebook

▾ **Warnings (7)**

7 WARNING(S) found

`Age` has 177.0 (19.9%) missing values. Consider remove the column or replace null values.

`Cabin` has 687.0 (77.1%) missing values. Consider remove the column or replace null values.

Suggest_recommendations

```
[6]: titanic_ds.suggest_recommendations()
```

[6]:

Message	Variables	Suggested	Action	Code
Contains mostly unique values(100.00%)	PassengerId	Drop	Drop	.drop_columns(["PassengerId"])
			Do nothing	
Contains missing values(19.87%)	Age	Fill missing values with mean	Drop	.drop_columns(["Age"])
			Fill missing values with mean	.fillna({"Age": 29.6991})
			Fill missing values with median	.fillna({"Age": 28.0})
			Fill missing values with frequent	.fillna({"Age": 24.0})
			Fill missing values with constant	.fillna({"Age": "constant"})
			Do nothing	
Contains missing values(77.10%)	Cabin	Drop	Drop	.drop_columns(["Cabin"])
			Fill missing values with frequent	.fillna({"Cabin": "C23 C25 C27"})
			Fill missing values with constant	.fillna({"Cabin": "constant"})
			Do nothing	
Contains missing values(2)	Embarked	Fill missing values with frequent	Drop	.drop_columns(["Embarked"])
			Fill missing values with frequent	.fillna({"Embarked": "S"})
			Fill missing values with constant	.fillna({"Embarked": "constant"})
			Do nothing	

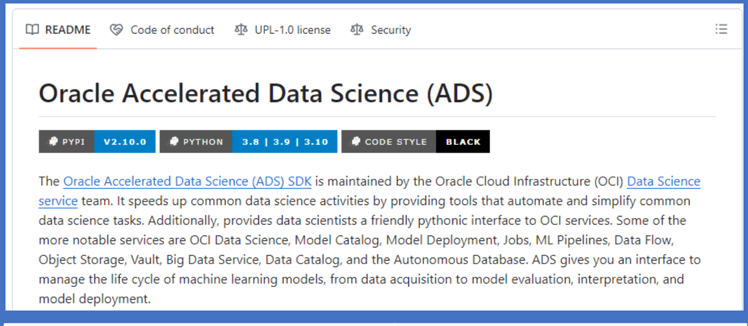
Oracle-ADS

Manual

- We can now take these recommendations and apply them in “Data Preparation / Data Cleaning” stages.
- This would be manual process - there are times this is required, and useful.

Programmatically

- Oracle-ADS can do this for us, using another function **auto_transform**.
- This will apply all recommended changes from **suggest_recommendations** to return a transformed dataset.


README Code of conduct UPL-1.0 license Security

Oracle Accelerated Data Science (ADS)

PYPI v2.10.0 PYTHON 3.8 | 3.9 | 3.10 CODE STYLE BLACK

The Oracle Accelerated Data Science (ADS) SDK is maintained by the Oracle Cloud Infrastructure (OCI) Data Science service team. It speeds up common data science activities by providing tools that automate and simplify common data science tasks. Additionally, provides data scientists a friendly pythonic interface to OCI services. Some of the more notable services are OCI Data Science, Model Catalog, Model Deployment, Jobs, ML Pipelines, Data Flow, Object Storage, Vault, Big Data Service, Data Catalog, and the Autonomous Database. ADS gives you an interface to manage the life cycle of machine learning models, from data acquisition to model evaluation, interpretation, and model deployment.

ORACLE ADS v2.10.0

File Edit View Run Kernel Git Tabs Settings Help

Oracle-ADS exploration.ipynb

Code

Python [conda env

[6]:
```
titanic_ds.suggest_recommendations()
```

[6]:

Message	Variables	Suggested	Action	Code
Contains mostly unique values(100.00%)	PassengerId	Drop	Drop	.drop_columns(["PassengerId"])
			Do nothing	
Contains missing values(19.87%)	Age	Fill missing values with mean	Drop	.drop_columns(["Age"])
			Fill missing values with mean	.fillna({"Age": 29.6991})
			Fill missing values with median	.fillna({"Age": 28.0})
			Fill missing values with frequent	.fillna({"Age": 24.0})
			Fill missing values with constant	.fillna({"Age": "constant"})
			Do nothing	
Contains missing values(77.10%)	Cabin	Drop	Drop	.drop_columns(["Cabin"])
			Fill missing values with frequent	.fillna({"Cabin": "C23 C25 C27"})
			Fill missing values with constant	.fillna({"Cabin": "constant"})
			Do nothing	
Contains missing values(2)	Embarked	Fill missing values with frequent	Drop	.drop_columns(["Embarked"])
			Fill missing values with frequent	.fillna({"Embarked": "S"})
			Fill missing values with constant	.fillna({"Embarked": "constant"})
			Do nothing	

[ ]:
```
transformed_titanic_ds = titanic_ds.auto_transform()
```

[ ]:
```
transformed_titanic_ds.visualize_transforms()
```

0 2 Python [conda env:generalml_p38_cpu_v1] | Idle Saving completed Mode: Command Ln 1, Col 53 Or

Oracle-ADS – auto_transform

- We can see that **Passenger ID** and **Cabin** have been dropped.
- There are no missing values present in **Age** or **Embarked**.
- Age – missing populated with mean value.
- Embarked – missing populated with most frequent.
- Runs in one line of code.

```
[16]: transformed_titanic_ds.head()
```

[16]:

	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Embarked
0	False	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	S
1	True	1	Cumings, Mrs. John Bradley (Florence Briggs Thayer)	female	38.0	1	0	PC 17599	71.2833	C
2	True	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	S
3	True	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	S
4	False	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	S

```
[17]: def draw_missing_data_table(transformed_titanic_ds):
    total=transformed_titanic_ds.isnull().sum().sort_values(ascending=False)
    percent=(transformed_titanic_ds.isnull().sum()/transformed_titanic_ds.isnull().count()).sort_values(ascending=False)*100
    missing_data=pd.concat([total,percent],axis=1,keys=['Total','Percent'])
    return missing_data
```

```
[18]: # Analyse missing data
draw_missing_data_table(transformed_titanic_ds)
```

[18]:

	Total	Percent
Survived	0	0.0
Pclass	0	0.0
Name	0	0.0
Sex	0	0.0
Age	0	0.0
SibSp	0	0.0
Parch	0	0.0
Ticket	0	0.0
Fare	0	0.0
Embarked	0	0.0

Summary

Oracle Accelerated Data Science (Oracle-ads)

The package also contains a number of methods in the ADS SDK to automatically **visualize** a dataset and understand it in greater detail.

Show_in_notebook()

provides a comprehensive preview of a data set's basic information

```
[5]: titanic_ds.show_in_notebook()
```

Summary

Type: BinaryClassificationDataset

891 Rows, 12 Columns

Column Types:

- **categorical:** 6 features
- **ordinal:** 4 features
- **continuous:** 2 features

Note: Visualizations use a sampled subset of the dataset, this is to improve plotting performance. The sample size is calculated to be statistically significant within the confidence level: 95 and confidence interval: 1.0. The sampled data has 891 rows

- The confidence *level* refers to the long-term success rate of the method, that is, how often this type of interval will capture the parameter of interest.
- A specific confidence *interval* gives a range of plausible values for the parameter of interest

Features (12)

Correlations

Warnings (7)

suggest_recommendations()

displays issues and recommends changes to resolve data issues

```
titanic_ds.suggest_recommendations()
```

Message	Variables	Suggested	Action	Code
Contains mostly unique values(100.00%)	PassengerId	Drop	Drop	.drop_columns(["PassengerId"])
			Do nothing	
Contains missing values(19.87%)	Age	Fill missing values with mean	Drop	.drop_columns(["Age"])
			Fill missing values with mean	.fillna({"Age": 29.6991})
			Fill missing values with median	.fillna({"Age": 28.0})
			Fill missing values with frequent	.fillna({"Age": 24.0})
			Fill missing values with constant	.fillna({"Age": "constant"})
			Do nothing	
Contains missing values(77.10%)	Cabin	Drop	Drop	.drop_columns(["Cabin"])
			Fill missing values with frequent	.fillna({"Cabin": "C23 C25 C27"})
			Fill missing values with constant	.fillna({"Cabin": "constant"})
			Do nothing	
Contains missing values(2)	Embarked	Fill missing values with frequent	Drop	.drop_columns(["Embarked"])
			Fill missing values with frequent	.fillna({"Embarked": "S"})
			Fill missing values with constant	.fillna({"Embarked": "constant"})
			Do nothing	

Oracle Accelerated Data Science (Oracle-ads)

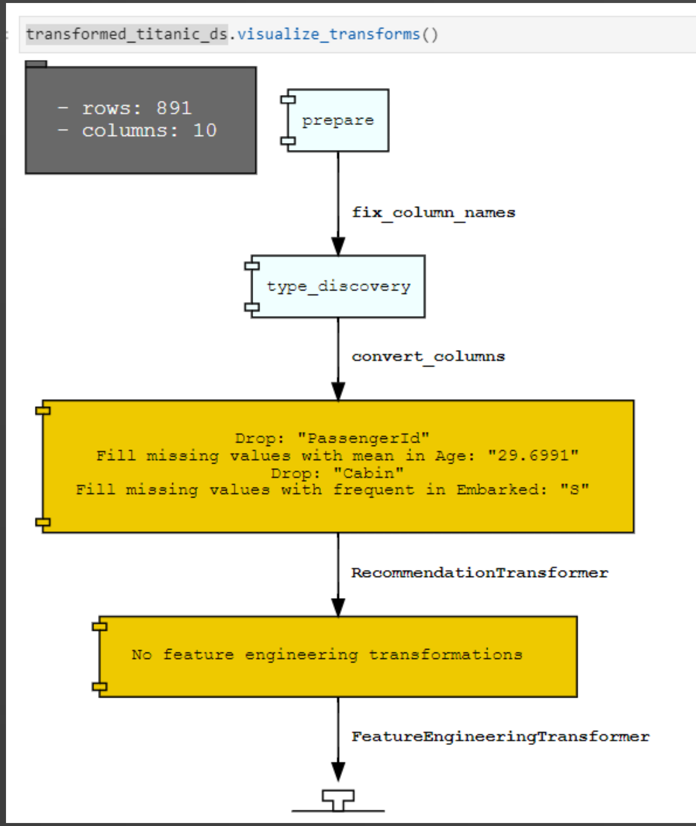
It contains several methods in the ADS SDK to automatically **transform** a dataset

auto_transform()

returns a transformed data set, with all recommendations and optimizations applied automatically

visualize_transforms()

visualizes the transformation that has been performed on a data set



Accelerated Data Science – with shortcuts!

- Oracle Data Science includes several keyboard shortcuts that can greatly enhance your productivity and save you time.

- Here are a few of my favourites that you might try…

Accelerated Data Science – with shortcuts!

Accelerated Data Science – with shortcuts!

Ctrl+Enter: Run the current cell.

Shift+Enter: Run the current cell and move to the next cell.

Alt+Enter: Run the current cell and insert a new cell below.

Ctrl+/: Comment or uncomment the selected code.

Esc+M: Convert the current cell to a Markdown cell.

Esc+Y: Convert the current cell to a code cell.

Any Questions?

Phil Godfrey

Principal Data Analytics & AI Consultant, Vertice

Philip.Godfrey@verticecloud.com

@PhilipGodfrey11