

NLP – PYTHON – ALTERYX

WENDY WONG

SENIOR ANALYTICS CONSULTANT AND DIGITAL
ACCELERATOR AT PWC AUSTRALIA





AGENDA

Learning Objectives

- Define Python and compare it to other programming languages
- Discuss NLP and how it is used in different industries
- Define Alteryx, explain how it is used and why it is so popular today in the marketplace

Disclaimer: The following slides do not reflect the opinions of my employer, but are my own personal opinions



ABOUT ME



ABOUT ME – FROM FINANCIAL PLANNING TO DATA SCIENCE

Perpetual 



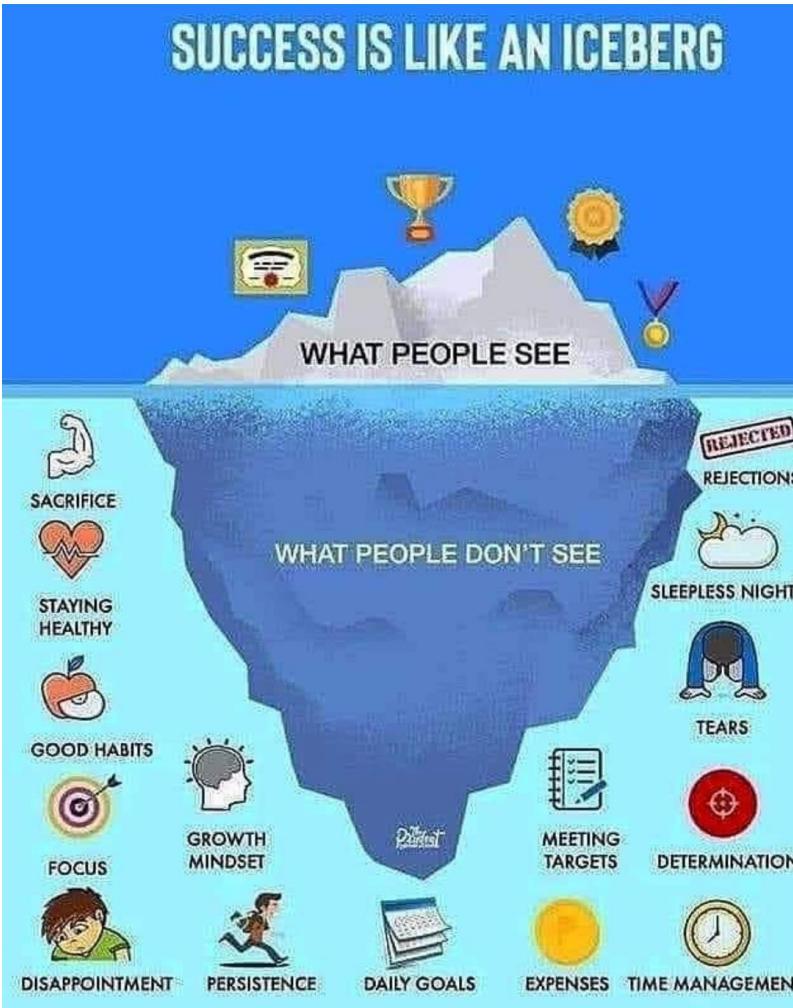


Vishy Narayanan • 1st

Partner | Chief Digital & Information Officer | PwC Australia

1w

A timely reminder of what lies beneath the surface when sometimes all we see (and want to see) is what's on top...pls spare a thought for those that are doing it tough and lend your support 🙏



Tip of the Iceberg

LinkedIn post

- You are making progress even when you do not feel you are.
- Grit
- GFC
- Reskill /Retrain
- Be uncomfortable



ABOUT ME – NEW LIFE IN DATA SCIENCE

iapa



The screenshot shows the homepage of the WiDSconference.org website. The header features the WiDS logo and navigation links for CONFERENCE, VIDEOS, DATATHON, PODCAST, NEWS, and CONTACT. The main content area features a large image of the Stanford University campus, specifically the Main Quad, with the text "Global Women in Data Science Conference" overlaid. Below this, a subtext states: "The next Women in Data Science Conference will be held March 2, 2020 at Stanford University and an estimated 150 regional events worldwide!"

About Women in Data Science

The Women in Data Science (WiDS) initiative aims to inspire and educate data scientists worldwide, regardless of gender, and to support women in the field.

WiDS started as a conference at Stanford in November 2015. Now, WiDS includes a global conference, with approximately 150+ regional events worldwide; a datathon, encouraging participants to hone their skills using a social impact challenge; and a podcast, featuring leaders in the field talking about their work, their journeys, and lessons learned.





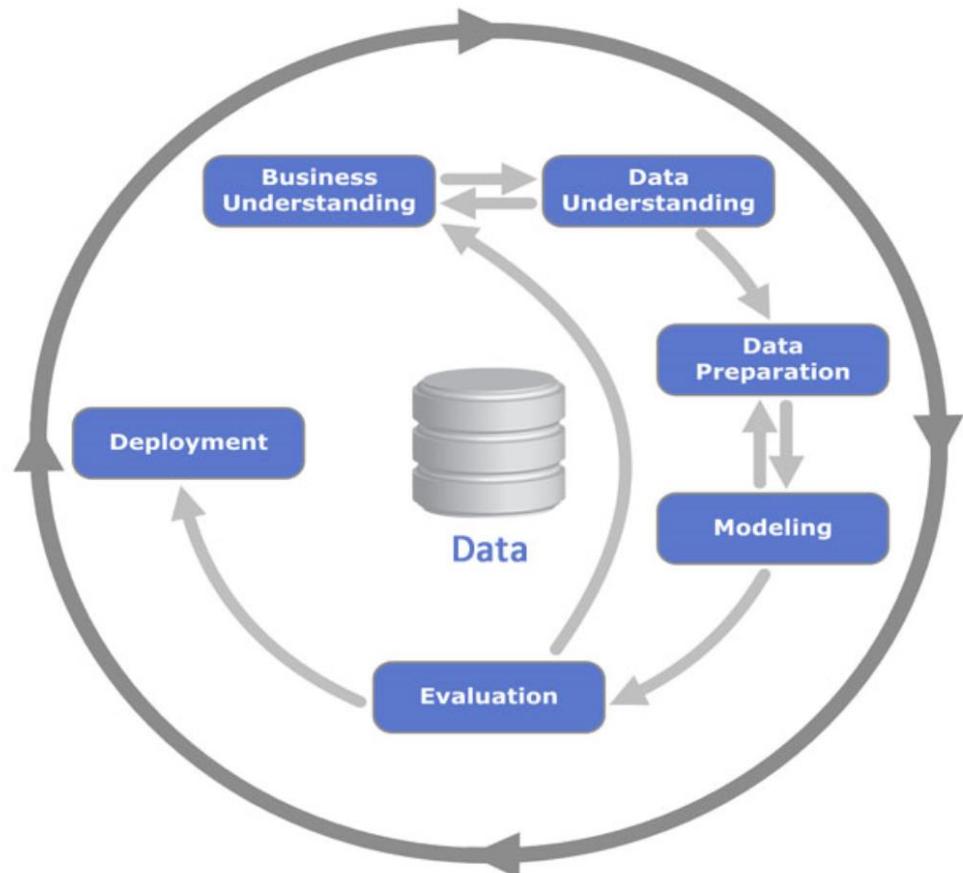
DATA SCIENCE



CRISP-DM Process Diagram

WHAT IS CRISP DM?

Cross Industry Standard Process for Data Mining



- Data Science
- Data Analytics
- Consulting



PYTHON



PYTHON

- Python is an open source language

What is Python?

- Python is an open-source
- Python is free
- Python was created by Guido Van Rossum in 1991
- Object-oriented language can execute code that includes data attributes

Where is Python used today?	Machine Learning Algorithm
<ul style="list-style-type: none">• Recommendation Engines e.g. Amazon, Netflix, Online shopping	Collaborative filtering
<ul style="list-style-type: none">• Virtual Assistants e.g. Google Home, Alexa	Deep Learning (Recurrent Neural Networks) and Natural Language Processing
<ul style="list-style-type: none">• Commercial machine learning applications e.g. customer segmentation by News Corp's readers in the cloud using AWS	K-means clustering
<ul style="list-style-type: none">• Rapid prototyping e.g. Run an experiment which group of customers will be shown a new website by Atlassian	A/B Testing
<ul style="list-style-type: none">• Analyzing relationships e.g. Analysing the Sydney trains network to improve Opal card services	Network Analysis, Graph databases
<ul style="list-style-type: none">• Prediction problem e.g. Predict which class(segment) of customers Westpac will apply for a credit card	Multi-class Logistic regression
<ul style="list-style-type: none">• Data Analysis e.g. Cleaning large datasets at PwC and reproducing the code. Reduce data entry error	Libraries Numpy (data manipulation with dataframes) and Scikit-Learn for mathematical calculations
<ul style="list-style-type: none">• Text Analytics e.g. Allianz Insurance wants to analyze survey feedback	Natural Language Processing



PRE-WORK DOWNLOAD PYTHON 3.7

Download Python 3.7 version

<https://www.anaconda.com/distribution/>

Windows | macOS | Linux

Anaconda 2019.07 for Windows Installer

Python 3.7 version

Download

64-Bit Graphical Installer (486 MB)
32-Bit Graphical Installer (418 MB)

Python 2.7 version

Download

64-Bit Graphical Installer (427 MB)
32-Bit Graphical Installer (361 MB)

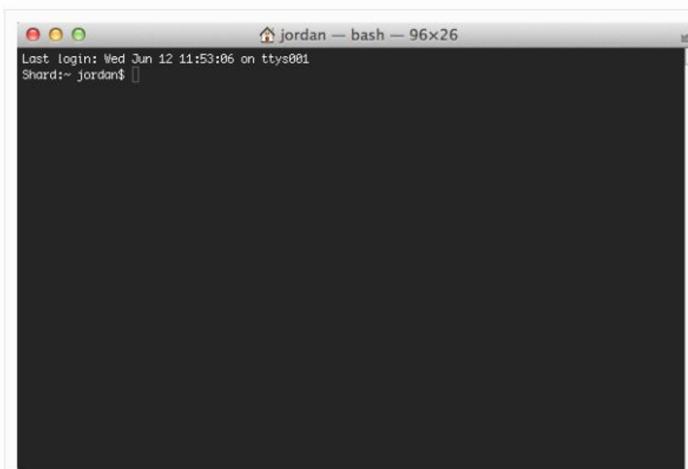


Launch Python 3.7 from Mac

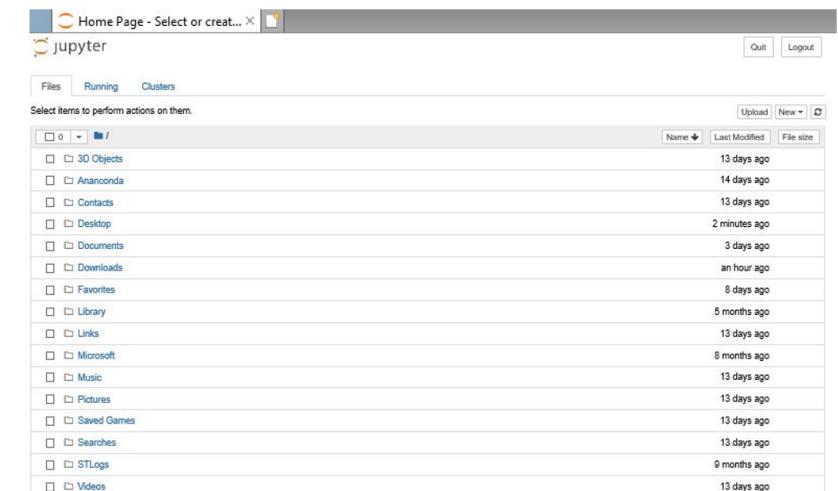
1. Create a **new folder** e.g. Python Training from your Desktop.
2. In **Terminal**, type **Jupyter Notebook**.

Launch Jupyter
Notebook

Entering Terminal



A screenshot of a terminal window titled "jordan — bash — 96x26". The window shows the command line interface with the prompt "jordan\$". Above the terminal, the status bar indicates "Last login: Wed Jun 12 11:53:06 on ttys001".



A screenshot of the Jupyter Notebook interface. The title bar says "Home Page - Select or creat... X". The main area shows a list of files in a tree view under the root directory "/". The list includes various folders and files like "3D Objects", "Ananconda", "Contacts", "Desktop", "Documents", "Downloads", "Favorites", "Library", "Links", "Microsoft", "Music", "Pictures", "Saved Games", "Searches", "STLogs", and "Videos". Each item has a timestamp next to it indicating when it was last modified.

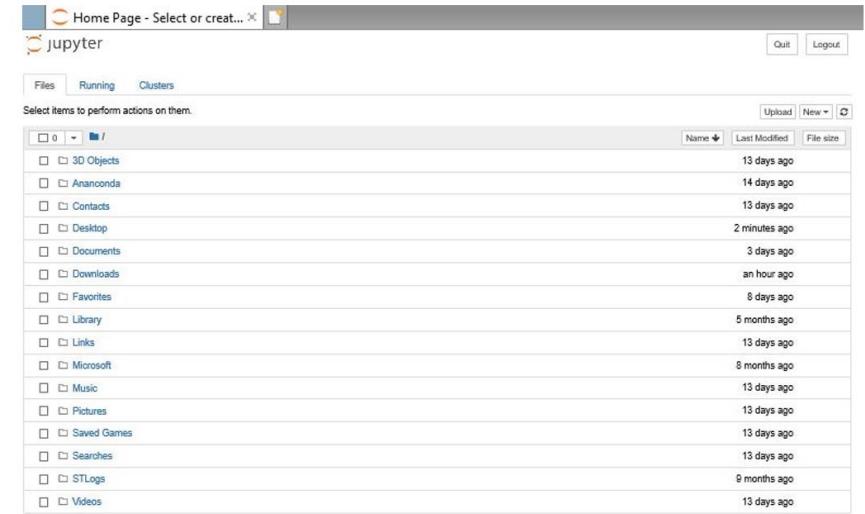
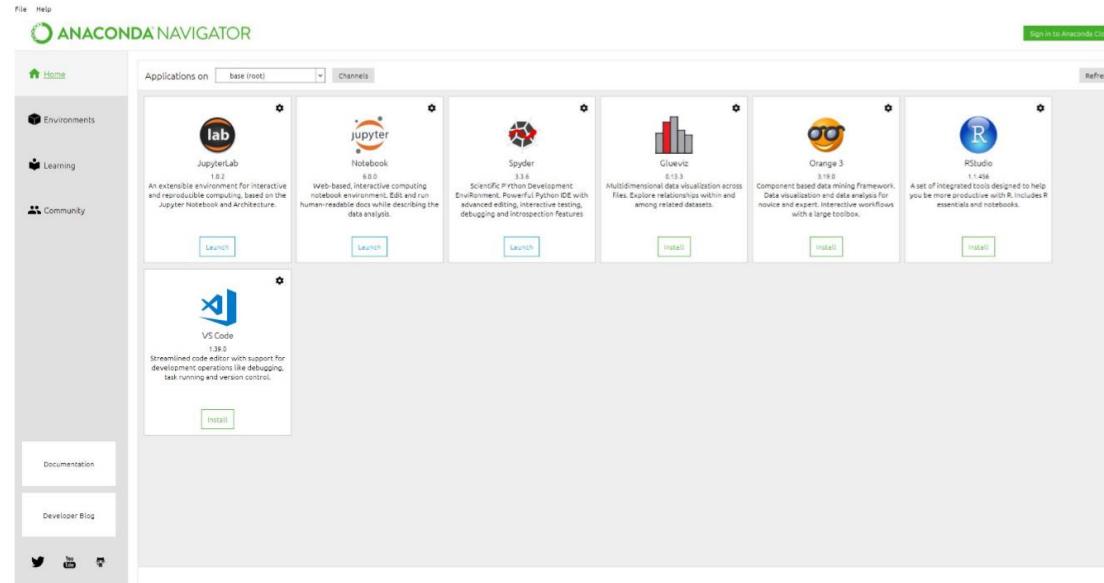
Last Modified	Name
13 days ago	3D Objects
14 days ago	Ananconda
13 days ago	Contacts
2 minutes ago	Desktop
3 days ago	Documents
an hour ago	Downloads
8 days ago	Favorites
5 months ago	Library
13 days ago	Links
8 months ago	Microsoft
13 days ago	Music
13 days ago	Pictures
13 days ago	Saved Games
13 days ago	Searches
9 months ago	STLogs
13 days ago	Videos



Launch Python 3.7 from Windows

1. Create a **new folder** e.g. Python Training from your Desktop.
2. From your **Start Menu**, type **Anaconda Navigator (Anaconda)**.
3. Click **Ok** on the pop up message.
4. Click **Launch from Jupyter Notebook**.

Launch Jupyter
Notebook



MAC Users:

In Terminal, type:
jupyter notebook to
launch your notebook

Top Analytics, Data Science, Machine Learning Software

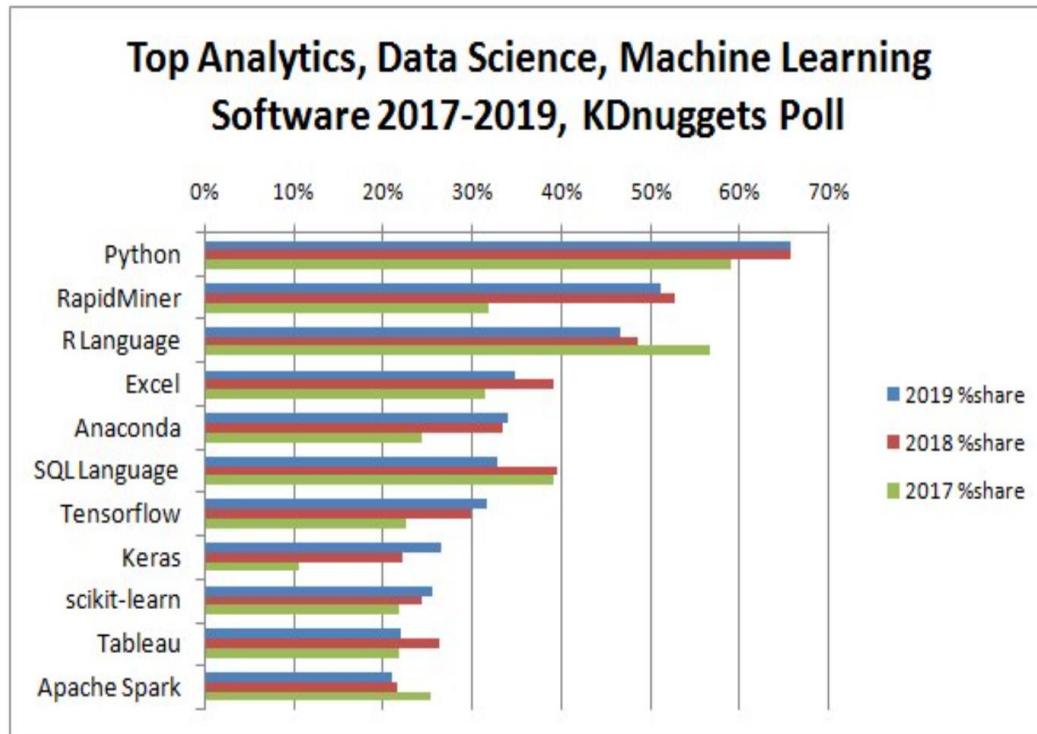


Fig 1: KDnuggets Analytics/Data Science 2019 Software Poll: top tools in 2019, and their share in the 2017, 2018 polls

PYTHON

- Reproducibility
- Turn your code into a slide deck for your boss
- Open Source
- Notebooks – code + output
- Data Visualisation
- Produce a report as a data analyst
- Deep Learning
- Machine Learning



LAB – PYTHON DEMO



CREATE A DATABASE ENGINE IN PYTHON

Slide Type **Slide** ▾

Create database engine in Python

In [1]: **▶** Slide Type **Slide** ▾

```
1 # Install the package psycopg2
```

In [2]: **▶** Slide Type **Slide** ▾

```
1 pip install psycopg2
```

Requirement already satisfied: psycopg2 in c:\users\wendy\anaconda3\anaconda3_1\lib\site-packages (2.8.4)

Note: you may need to restart the kernel to use updated packages.

In [3]: **▶** Slide Type **Slide** ▾

```
1 import psycopg2
```

PYTHON REFRESHER

11/25/2019

17

3. Python Basics

Python as a Calculator

Division

```
In [1]: print(5/8)
```

```
0.625
```

Multiplication

```
In [2]: 10*4
```

```
Out[2]: 40
```

```
¶
```

```
In [3]: print(100*1.1**7)
```

```
194.87171000000012
```

Assign a variable

```
In [4]: weight = 55
```

```
In [5]: height = 1.79
```

```
In [6]: ### Calculate BMI = weight/Height ^2
```

```
In [7]: bmi = weight/height **2
```

```
In [8]: bmi
```

```
Out[8]: 17.165506694547613
```

Check the data type

What is the data type of bmi?

```
In [9]: type(bmi)
```

```
Out[9]: float
```

CONNECTING TO A DATABASE - PYTHON

Python - Connecting to the database

```
### example: AWS PostgreSQL RDS database

* users - contains publicly-viewable data about each user in the MDSI Slack instance.
* channels - contains data about each public channel in the MDSI Slack instance.
* messages - contains all messages posted in public channels in the MDSI Slack instance
```

In [4]:

```
1 # Import packages
2 from sqlalchemy import create_engine
3 import pandas as pd
```

In [5]:

```
1 # Create an engine that connects to the AWS PostgreSQL RDS database to the psycopg2 driver: engine
2 engine = create_engine("postgresql+psycopg2://dsp2019:oZkK6vgRbvDK@/mdsisslack.clnutj7nhgyn.us-east-2.rds.
```

In [6]:

```
1 # Print table names
2 print(engine.table_names())
```

```
['messages', 'user_analysis', 'users', 'channels']
```

SELECT AND JOIN YOUR DATA

Write a SELECT statement to query the 'users' table

In [8]:

```
1 # Perform query: rs  
2 rs = con.execute('select * from users')
```

In [9]:

```
1 # Save results of the query to a pandas dataframe: df  
2 df = pd.DataFrame(rs.fetchall())
```

In [10]:

```
1 # Close connection  
2 con.close()
```

In [11]:

```
1 # Set the Dataframe column names  
2 df.columns = rs.keys()
```

In [12]:

```
1 # Print the dataframe to the shell  
2 print(df)
```

Joining 'messages' and 'channels' tables with INNER Join

In [37]:

```
1 # Import packages  
2 from sqlalchemy import create_engine  
3 import pandas as pd
```

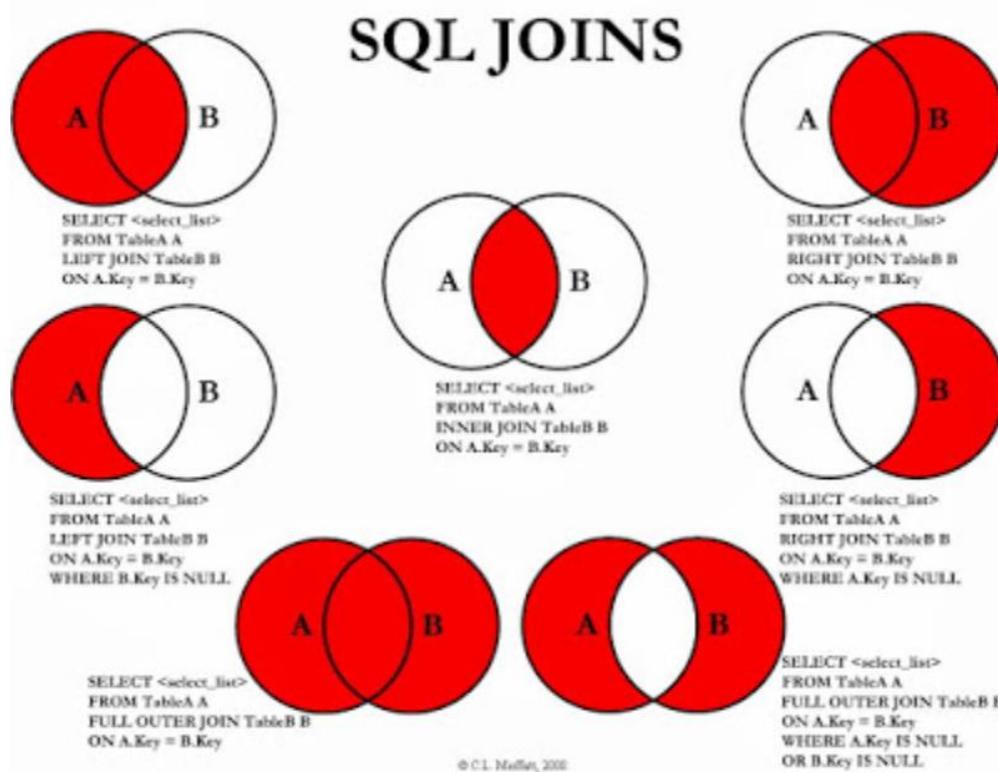
In [38]:

```
1 # Create an engine that connects to the AWS PostgreSQL RDS database to the psycopg2 driver: engine  
2 engine = create_engine("postgresql+psycopg2://dsp2019:oZkK6vgRbvDK@mdsslack.clnutj7nhgyn.us-east-2.rds.  
3 < />
```

In [39]:

```
1 # Open engine connection  
2 con = engine.connect()
```

CHECK WHICH JOIN IS APPROPRIATE FOR YOU



- Do you want to preserve user names on the left?
- How will you treat N/A's?
- An inner join may reduce your data points
- What problem are you trying to answer?

Write to CSV and read into R

In [*]:

```
1 # write dataframe to csv
2
3 df4 = df3.to_csv('df3.csv')
```

Slide Type **Slide** ▾

Install Feather to pass data between Python and R

In [*]:

```
1 pip install feather-format
```

Slide Type **Slide** ▾

**WRITE PYTHON
FILE TO R**

Call Python from R Markdown

- <https://rstudio.github.io/reticulate/>

```
13
14 ````{python}
15 import pandas
16 flights = pandas.read_csv("flights.csv")
17 flights = flights[flights['dest'] == "ORD"]
18 flights = flights[['carrier', 'dep_delay', 'arr_delay']]
19 flights = flights.dropna()
20 ``
21
22 ````{r, fig.width=7, fig.height=3}
23 library(ggplot2)
24 ggplot(py$flights, aes(carrier, arr_delay)) + geom_point() + geom_jitter()
25 ````
```

In [101]:

Slide Type **Slide** ▾

```
1 # Start with generating a word cloud on the variable 'user_name':  
2 text = df3.user_name[0]  
3  
4 # Create and generate a word cloud image:  
5 wordcloud = WordCloud().generate(text)  
6  
7 # Display the generated image:  
8 plt.imshow(wordcloud, interpolation='bilinear')  
9 plt.axis("off")  
10 plt.show()
```



PYTHON WORD CLOUDS – TEXT DATA



LAB - PYTHON

DEMO 2



DATA UNDERSTANDING

City of New York Complaints

Csv file

```
[2]:  
## Load the data into a Pandas Dataframe from csv file  
import pandas as pd  
df = pd.read_csv('Complaint_Problems.csv')  
### Inspect the first 5 values of the dataset  
df.head()
```

Dataframe

ProblemID	ComplaintID	UnitTypeID	UnitType	SpaceTypeID	SpaceType	TypeID	Type	MajorCategoryID	MajorCategory	MinorCategory
0	17307278	8412850	91 APARTMENT	543	ENTIRE APARTMENT	1	EMERGENCY	56	DOOR/WINDOW	
1	17317058	8417365	91 APARTMENT	543	ENTIRE APARTMENT	3	NON EMERGENCY	63	UNSANITARY CONDITION	
2	17016467	8249017	91 APARTMENT	545	ENTRANCE/FOYER	1	EMERGENCY	56	DOOR/WINDOW	
3	14548958	6967900	91 APARTMENT	541	BATHROOM	1	EMERGENCY	9	PLUMBING	
4	14548959	6967900	91 APARTMENT	541	BATHROOM	3	NON EMERGENCY	9	PLUMBING	

Variable

Data Dictionary

There were 18 variables within the data set with 1045040 rows

- * ProblemID
- * ComplaintID
- * UnitTypeID
- * UnitType
- * SpaceTypeID
- * SpaceType
- * TypeID
- * Type
- * MajorCategoryID
- * MajorCategory
- * MinorCategoryID
- * MinorCategory
- * CodeID
- * Code
- * StatusID
- * Status
- * StatusDate
- * StatusDescription

Tasks:

1. Insights from the dataset

- * Exploratory Data Analysis

2. Build a machine learning model that performs MultiClass classification to predict the outcome of complaint type

- * Type: 1=EMERGENCY 2=HARZARDOUS 3=IMMEDIATE EMERGENCY 4=NON EMERGENCY

* StatusDescription

* UnitType

* SpaceType

* MajorCategory

* MinorCategory

* Code

* Status

* StatusDate

Variables

11/25/2019

24



DATA PREPARATION

City of New York Complaints - Transform categorical variables

In [3]:

```
## Exploratory Data Analysis
#### Create a new dataframe for exploratory data analysis
df2 = df[['MajorCategory', 'Code', 'Status', 'StatusDate', 'Type', 'MinorCategory']]
df2.head()
```

Out[3]:

	MajorCategory	Code	Status	StatusDate	Type	MinorCategory
0	DOOR/WINDOW	LOOSE OR DEFECTIVE	CLOSE	3/31/17	EMERGENCY	WINDOW FRAME
1	UNSANITARY CONDITION	MICE	CLOSE	3/16/17	NON EMERGENCY	PESTS
2	DOOR/WINDOW	LOCK BROKEN OR MISSING	CLOSE	3/3/17	EMERGENCY	DOOR
3	PLUMBING	BROKEN OR MISSING	CLOSE	7/29/14	EMERGENCY	BATHTUB/SHOWER
4	PLUMBING	FAUCET BROKEN/MISSING/LEAKING	CLOSE	8/4/14	NON EMERGENCY	BATHTUB/SHOWER

In [6]:

```
y_enc = l_encoder.transform(y)
np.unique(y_enc)
```

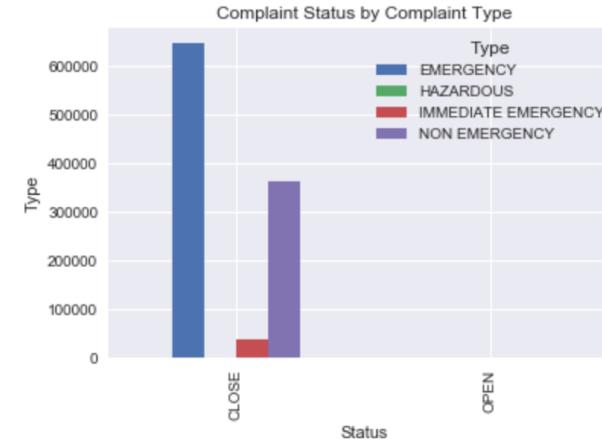
Out[6]:

```
array([0, 1, 2, 3])
```

In [44]:

```
1 # Barplot of Status grouped by Type
2 pd.crosstab(df2.Status, df2.Type).plot(kind='bar')
3 plt.title('Complaint Status by Complaint Type')
4 plt.xlabel('Status')
5 plt.ylabel('Type')
6
```

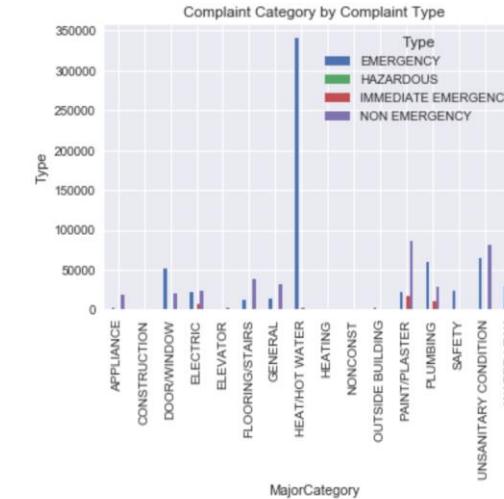
Out[44]: Text(0,0.5,'Type')



In [38]:

```
1 # Barplot of MajorCategory grouped by Type
2 pd.crosstab(df2.MajorCategory, df2.Type).plot(kind='bar')
3 plt.title('Complaint Category by Complaint Type')
4 plt.xlabel('MajorCategory')
5 plt.ylabel('Type')
```

Out[38]: Text(0,0.5,'Type')



Imbalanced classes – resampling methods cross validation

EXPLANATORY INSIGHTS



PREDICTIVE MODELLING

City of New York Complaints - Model Evaluation

Slide Type: Slide

Model Selection

We are now ready to experiment with different machine learning models, evaluate their accuracy and find the source of any potential issues.

We will benchmark the following four models:

1. Logistic Regression
2. (Multinomial) Naive Bayes
3. Linear Support Vector Machine
4. Random Forest

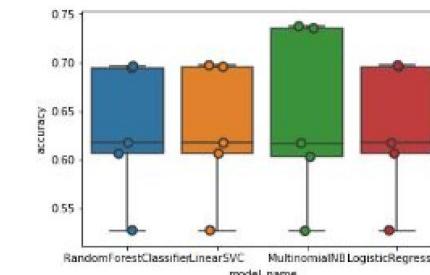
In [19]:

```
from sklearn.linear_model import LogisticRegression
from sklearn.ensemble import RandomForestClassifier
from sklearn.svm import LinearSVC
from sklearn.model_selection import cross_val_score
models = [
    RandomForestClassifier(n_estimators=200, max_depth=3, random_state=0),
    LinearSVC(),
    MultinomialNB(),
    LogisticRegression(random_state=0),
]
CV = 5
cv_df = pd.DataFrame(index=range(CV * len(models)))
for model in models:
    model_name = model.__class__.__name__
    accuracies = cross_val_score(model, features, labels, scoring='accuracy', cv=CV)
    for fold_idx, accuracy in enumerate(accuracies):
        entries.append((model_name, fold_idx, accuracy))
cv_df = pd.DataFrame(entries, columns=['model_name', 'fold_idx', 'accuracy'])
import seaborn as sns
sns.boxplot(x='model_name', y='accuracy', data=cv_df)
sns.stripplot(x='model_name', y='accuracy', data=cv_df,
              size=8, jitter=True, edgecolor="gray", linewidth=2)
plt.show()
```

- Logistic Regression
- Random Forest
- Multi-nomial Naive Bayes
- Linear Support Vector Machine

Slide Type: Slide

Multi-Class Model Selection



Visualise the Model
Linear Support vector

- Multinomial Naive Bayes and
- Linear Support Vector Machine

Performed better than the other classifiers



EVALUATE THE MODEL

City of New York Complaints - Model Evaluation

```
In [22]:  
Slide Type | Slide  
  
model = MultinomialNB()  
X_train, X_test, y_train, y_test, indices_train, indices_test = train_test_split(features, labels, df.index, test_size=0.2)  
model.fit(X_train, y_train)  
y_pred = model.predict(X_test)  
from sklearn.metrics import confusion_matrix  
conf_mat = confusion_matrix(y_test, y_pred)  
fig, ax = plt.subplots(figsize=(10,10))  
sns.heatmap(conf_mat, annot=True, fmt="d",  
            xticklabels=TypeID_df.Type.values, yticklabels=TypeID_df.Type.values)  
plt.ylabel('Actual')  
plt.xlabel('Predicted')  
plt.show()
```

```
Slide Type | Slide

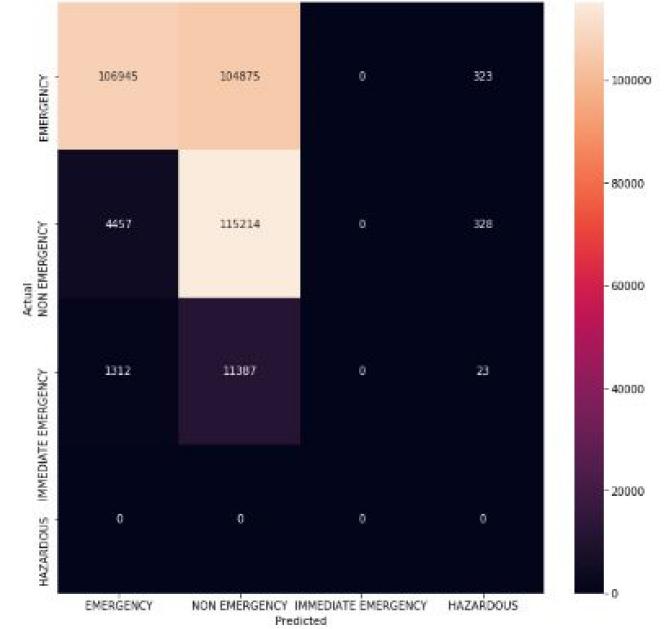
[[{"category": "EMERGENCY", "subcategory": "Top unigrams", "text": ". complaint", "order": 1}, {"category": "EMERGENCY", "subcategory": "Top unigrams", "text": ". violations", "order": 2}, {"category": "EMERGENCY", "subcategory": "Top bigrams", "text": ". . complaint closed", "order": 3}, {"category": "EMERGENCY", "subcategory": "Top bigrams", "text": ". . department housing", "order": 4}, {"category": "HAZARDOUS", "subcategory": "Top unigrams", "text": ". .", "order": 1}, {"category": "HAZARDOUS", "subcategory": "Top unigrams", "text": ". .", "order": 2}, {"category": "HAZARDOUS", "subcategory": "Top bigrams", "text": ". . issued information", "order": 3}, {"category": "HAZARDOUS", "subcategory": "Top bigrams", "text": ". . www nyc", "order": 4}, {"category": "IMMEDIATE EMERGENCY", "subcategory": "Top unigrams", "text": ". .", "order": 1}, {"category": "IMMEDIATE EMERGENCY", "subcategory": "Top unigrams", "text": ". .", "order": 2}, {"category": "IMMEDIATE EMERGENCY", "subcategory": "Top bigrams", "text": ". . violations issued", "order": 3}, {"category": "IMMEDIATE EMERGENCY", "subcategory": "Top bigrams", "text": ". . conditions violations", "order": 4}, {"category": "NON EMERGENCY", "subcategory": "Top unigrams", "text": ". .", "order": 1}, {"category": "NON EMERGENCY", "subcategory": "Top unigrams", "text": ". .", "order": 2}, {"category": "NON EMERGENCY", "subcategory": "Top bigrams", "text": ". . conditions violations", "order": 3}, {"category": "NON EMERGENCY", "subcategory": "Top bigrams", "text": ". . development inspected", "order": 4}], [{"text": "Fitting the model"}]]
```

Fitting the mode

		Actual Values	
		Positive (1)	Negative (0)
Predicted Values	Positive (1)	TP	FP
	Negative (0)	FN	TN

confusion matrix

Confusion Matrix - produces **model evaluation metrics**





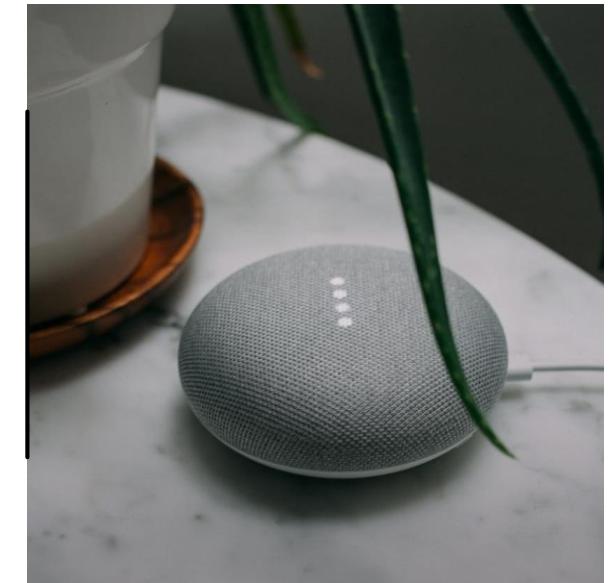
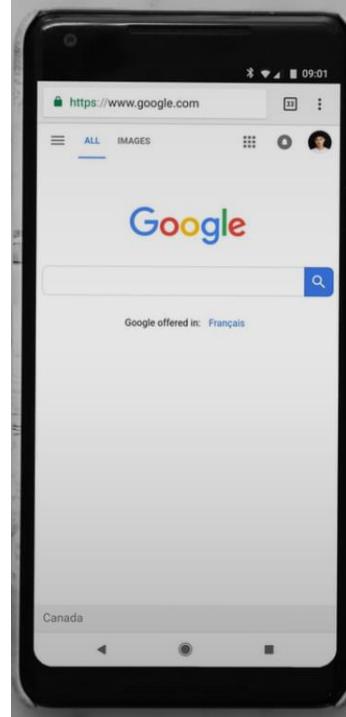
MODEL EVALUATION METRICS

The **confusion matrix**, which is a breakdown of predictions into a table showing correct predictions and the types of incorrect predictions made. Ideally, you will only see numbers in the diagonal, which means that all your predictions were correct!

- Precision is a measure of a classifier's exactness. The higher the precision, the more accurate the classifier.
- Recall is a measure of a classifier's completeness. The higher the recall, the more cases the classifier covers.
 - The F1 Score or F-score is a weighted average of precision and recall.
 - Area under the ROC curve.



NLP



NLP – UNSTRUCTURED DATA NATURAL LANGUAGE PROCESSING

Data Dictionary

There were 18 variables within the data set with 1045040 rows

- ProblemID
- ComplaintID
- UnitTypeID
- UnitType
- SpaceTypeID
- SpaceType
- TypeID
- Type
- MajorCategoryID
- MajorCategory
- MinorCategoryID
- MinorCategory
- CodeID
- Code
- StatusID
- Status
- StatusDate
- StatusDescription

The screenshot shows the Data.gov interface for a dataset titled 'Complaint Problems' from the 'City of New York'. The page includes the Data.gov logo, a search bar, and navigation links for DATA, TOPICS, IMPACT, APPLICATIONS, DEVELOPERS, and CONTACT. The main content area features the dataset's title, a large image of the Great Seal of the State of New York, and a brief description stating it is a Non-Federal dataset with different Terms of Use than Data.gov. It also notes the dataset was updated on November 6, 2019. Below this, sections for 'Access & Use Information' and 'Downloads & Resources' are displayed. The 'Access & Use Information' section includes links for Public, Non-Federal, and License information. The 'Downloads & Resources' section lists files available for download: Comma Separated Values File (CSV), RDF File, and JSON File. A tooltip on the CSV download button indicates it will direct to an external website with different content and privacy policies.

Data Source

- City of New York complaint problems dataset is publicly available from the website data.gov via the link:

<https://catalog.data.gov/dataset/complaint-problems-7052e>

- Data published by: data.cityofnewyork.us
- Data last updated: 6 September 2019

OBTAIN THE DATA

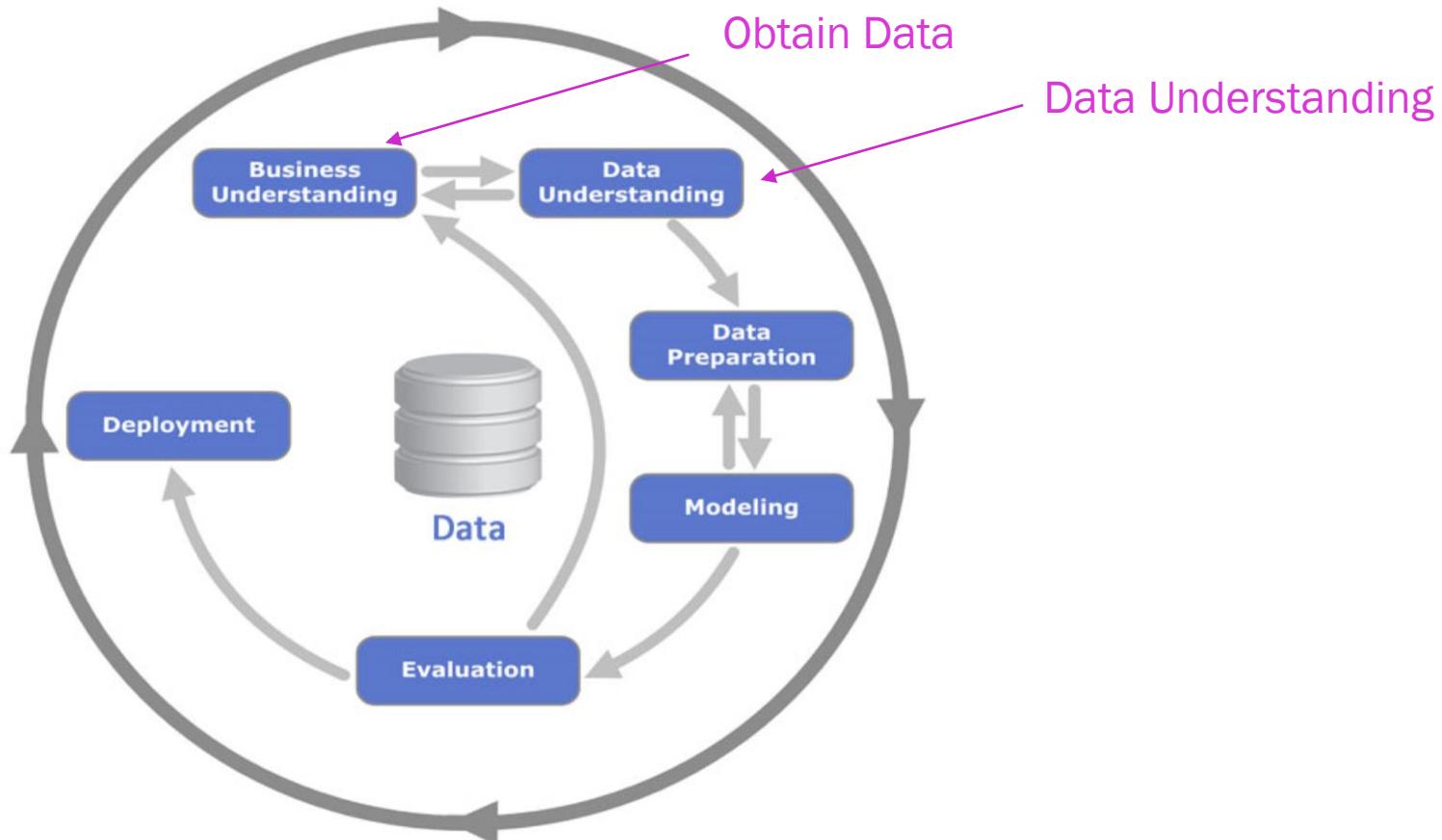
- Given a new complaint comes in, We want to assign it to one of 4 categories. The classifier makes the assumption that each new complaint is assigned to one and only one category.



CRISP-DM Process Diagram

WHAT IS CRISP DM?

Cross Industry Standard Process for Data Mining



Source: Kenneth Jensen



LAB - NLP DEMO

SPACY

In [77]:



Slide Type

Slide

```
1 # Visualise the input text with displacy
2
3 displacy.render(nytimes, style = "ent",jupyter = True)
```

"But this our purpose now is twelve month old DATE ,""And bootless 'tis to tell you we will go:""Therefore we meet not now. Then let me hear""Of you, my gentle cousin Westmoreland PERSON ,""What yesternight our council did decree""In forwarding this dear expedience.""My liege, this haste was hot in question,""And many limits of the charge set down""But yesternight: when all athwart there came""A post from Wales GPE loaden with heavy news,""Whose worst was, that the noble Mortimer PERSON ,""Leading the men of Herefordshire GPE to fight""Against the irregular and wild Glendower,""Was by the rude hands of that Welshman ORG taken,"" A thousand CARDINAL of his people butchered,""Upon whose dead corpse there was such misuse,""Such beastly shameless transformation,""By those Welshwomen done as may not be""Without much shame retold or spoken of."It seems then that the tidings of this broil""Brake off our business for the Holy Land ORG .""This match'd with other did, my gracious lord,""For more uneven and unwelcome news""Came from the north and thus it did import:"" On Holy-rood day WORK_OF_ART , the gallant Hotspur there,""Young Harry Percy and WORK_OF_ART brave Archibald PERSON ,""That ever-valiant and approved Scot PERSON ,""At Holmedon GPE met,""Where they did spend a sad and bloody hour TIME ,""As by discharge of their artillery,""And shape of likelihood, the news was told,""For he that brought them, in the very heat""And pride of their contention did take horse,""Uncertain of the issue any way."Here is a dear, a true industrious friend,""Sir Walter Blunt PERSON , new lighted from his horse."Stain'd with the variation of each soi"" Betwixt NORP that Holmedon GPE and this seat of ours,""And he hath brought us smooth and welcome news."The Earl of Douglas WORK_OF_ART is discomfited:"" Ten thousand CARDINAL bold Scots ORG , two CARDINAL and twenty CARDINAL knights,""Balk'd in their own blood did Sir Walter PERSON see""On Holmedon GPE 's plains. Of prisoners, Hotspur took"" Mordake the Earl of Fife WORK_OF_ART , and eldest son""To beaten Douglas PERSON , and the Earl of Athol WORK_OF_ART ,""Of Murray, Angus PERSON , and Menteith:""And is not this an honourable spoil?""A gallant prize? ha, cousin, is it not?""In faith,""It is a conquest for a prince to boast of."Yea PERSON , there thou makest me sad and makest me sin""In envy that my Lord Northumberland PERSON ""Should be the father to so blest a son,""A son who is the theme of honour's tongue,"" Amongst WORK_OF_ART a grove, the very straightest plant,""Who is sweet Fortune ORG 's minion and her pride?"" Whilst I, by looking on the praise of him WORK_OF_ART ,""See riot and dishonour

SPACY

In [34]:

```
1 # Find named entities, phrases and concepts
2
3 for entity in doc.ents:
4     print(entity.text, entity.label_)
```

```
ACT ISCENE I. London ORG
KING HENRY PERSON
the EARL of WESTMORELAND ORG
WALTER BLUNT PERSON
hoofsOf PRODUCT
one CARDINAL
one CARDINAL
shockAnd ORG
March DATE
one CARDINAL
kindred PERSON
Christ ORG
crossWe GPE
Forthwith PERSON
English LANGUAGE
fourteen hundred years ago DATE
the bitter cross ORG
```

Slide Type ▾



ALTERYX



ALTERYX DESIGNER

- Excel on steroids
- Self service analytics – you do not need to code
- What can it do?
 - Workflow Optimization – compress file size, speed up processing
 - Data Preparation and Data Blending (joins)
 - Writing fast and accurate expressions
 - Using Macros in workflows
 - Containers
 - Data Parsing – Text to Columns
 - Outputs files to csv, an Alteryx workflow and even a Tableau hyper file
 - Predictive analytics

The screenshot shows the Alteryx Designer product page. At the top, there are buttons for 'WATCH DEMO', 'FREE TRIAL', and a language selector set to 'ENGLISH'. Below these are navigation links for 'PRODUCTS', 'SOLUTIONS', 'PARTNERS', 'RESOURCES', 'COMMUNITY', and 'COMPANY'. The main title 'ALTERYX DESIGNER' is prominently displayed in large red letters. A sub-headline reads 'Repeatable workflows for self-service data analytics'. A 'FREE DESIGNER TRIAL' button is visible. Below the main title, there are tabs for 'PREP + BLEND', 'VISUALYTICS', 'PREDICTIVE', 'SPATIAL', 'SHARING INSIGHTS', and 'TECH SPECS'. The page footer includes a breadcrumb trail ('Products > Alteryx Platform > Alteryx Designer'), a section titled 'EXPERIENCE THE THRILL OF SOLVING', and a paragraph about streamlining data analysis processes.

Products > Alteryx Platform > Alteryx Designer

EXPERIENCE THE THRILL OF SOLVING

For many analysts throughout marketing, sales, finance, or customer insight, the process involved in prepping, blending, and analyzing data is slow and painful. It requires many tools and people to gather, cleanse, and join data from different sources, then more tools to build and publish analytic models – and then even more effort to get those models and insights into the hands of business decision makers.

Alteryx Designer streamlines the process by delivering a repeatable workflow for self-service data analytics, leading to deeper insights in hours, not weeks. Alteryx Designer empowers data analysts by combining data preparation, data blending, and analytics – predictive, statistical, and spatial – using the same intuitive user interface.



Machine Learning Platforms



Fig. 1: Gartner 2019 Magic Quadrant for Data Science and Machine Learning Platforms (as of Nov 2018)

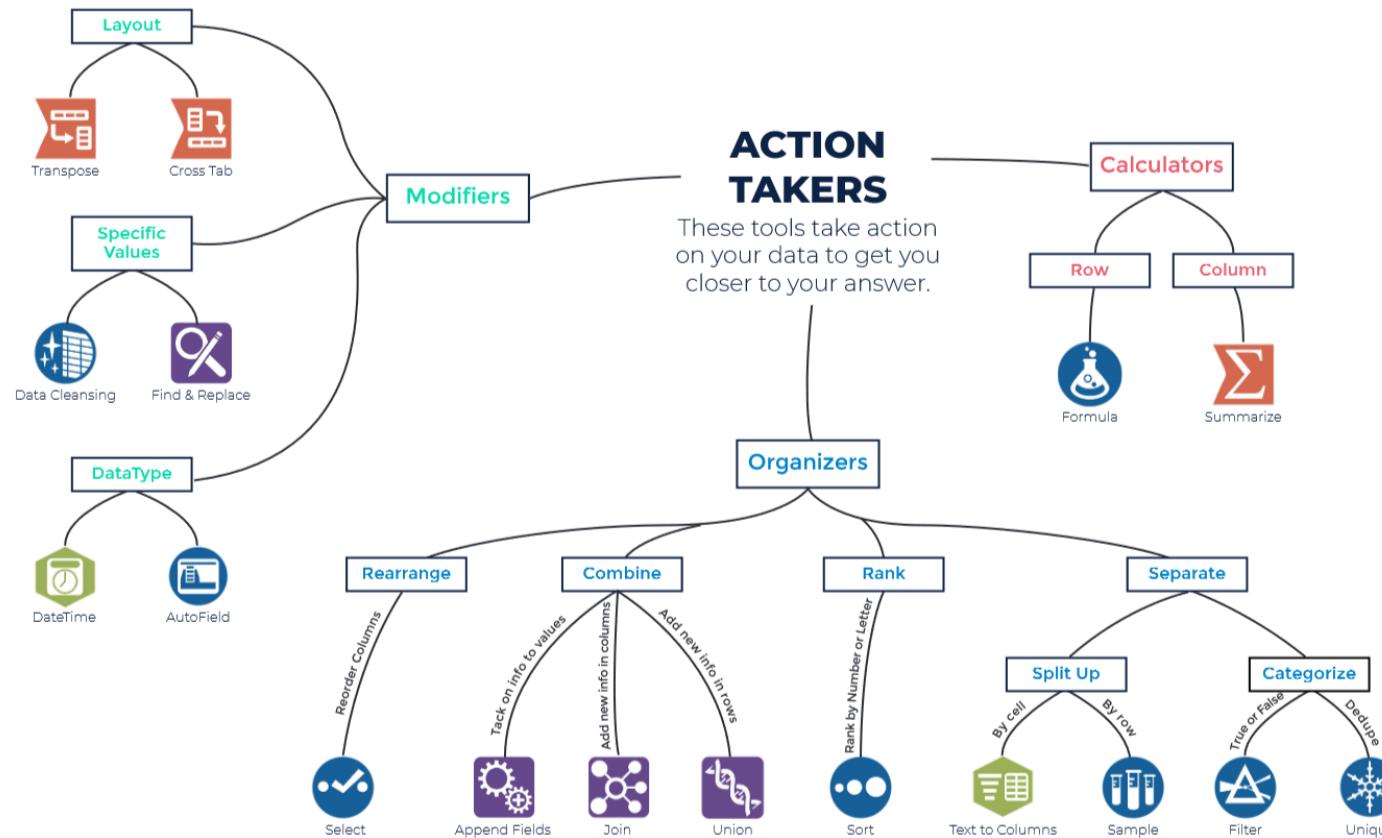
- Open source languages Python and R do not appear in Gartner's 2018 magic quadrant
- Alteryx is a 'challenger' in machine learning and data science



Choosing the right Alteryx Tool

CHOOSING THE RIGHT TOOL

Hover or click the icons for more info.



CHOOSING THE RIGHT ALTERYX TOOL

ACTIONS YOU MAY WANT TO TAKE...

Change Datatype	Select Formula Autofield
Row to Column	Transpose
Column to row	Cross Tab
Split one cell into multiple cells	Text to Column Formula
Combine lists by adding rows	Union
Combine lists by adding columns	Join Find & Replace Append Fields
Group Information	Union Find & Replace Summarize
Rank Data	Sort Summarize
Get rid of columns	Select
Get rid of empty values	Formula Filter
Get rid of rows	Formula Filter Sample
Get rid of punctuation or whitespace	Data Cleansing
Perform a calculation	Summarize Formula
Work with dates	Datetime Formula
Find a value (unique, min, max)	Summarize Unique Find & Replace
Identify records with a unique ID	Record ID
Replace a value	Find & Replace Formula
Input Data	Input Data
Rename Fields	Select
Reorder Fields	Select
View Results	Browse
Output Results	Output Data

TOOLS THAT CAN DO THAT IN DESIGNER

FUNCTIONS

When using functions in Designer, keep in mind that datatype is very important. The table on the right shows the function category and an X indicates that functions in that category are compatible with that column's corresponding datatype. This is not an exhaustive list. Rather, use this table to match your data's type and find a category that is compatible with that datatype to ensure the function will work. Note that you may need to change your data's datatype if you wish to use it with a particular function.

	String	Numeric	Datetime	Boolean	Spatial
Conditional	X	X	X	X	X
Conversion	X	X			
Datetime	X		X		
File	X				
Finance		X			
Math		X			
Math: Bitwise		X			
Min/Max		X			
Operators	X	X	X	X	X
Spatial		X			X
Specialized	X	X	X	X	X
String	X				
Test	X	X	X	X	X

TERMINOLOGY

Blend - merging data from different sources into one dataset, such as data from different spreadsheets, databases, or other sources into one complete dataset.

Concatenate - joining one or more text strings together.

Datatype - an attribute of data which lets the computer know how to interpret that value. There are 5 main datatypes in Designer (string, numeric, DateTime, Boolean, Spatial). Datatypes can be changed for particular values.

Delimiter - a sequence of one or more characters that creates a boundary between values. Common delimiters include commas, pipes, and quotes.

Filter - filtering separates your data into two streams: True containing the data met your criteria, and False containing the data that did not meet your criteria.

Flag - flagging data is a technique used to categorize data. This is usually accomplished with a conditional statement which checks values against a set of criteria and creates a corresponding flag in another column.

Parse - parsing separates values based on delimiters. Examples include: separating keywords from phrases, separating numbers from letters, or area codes from phone numbers.

Sort - ranking items in ascending or descending order.

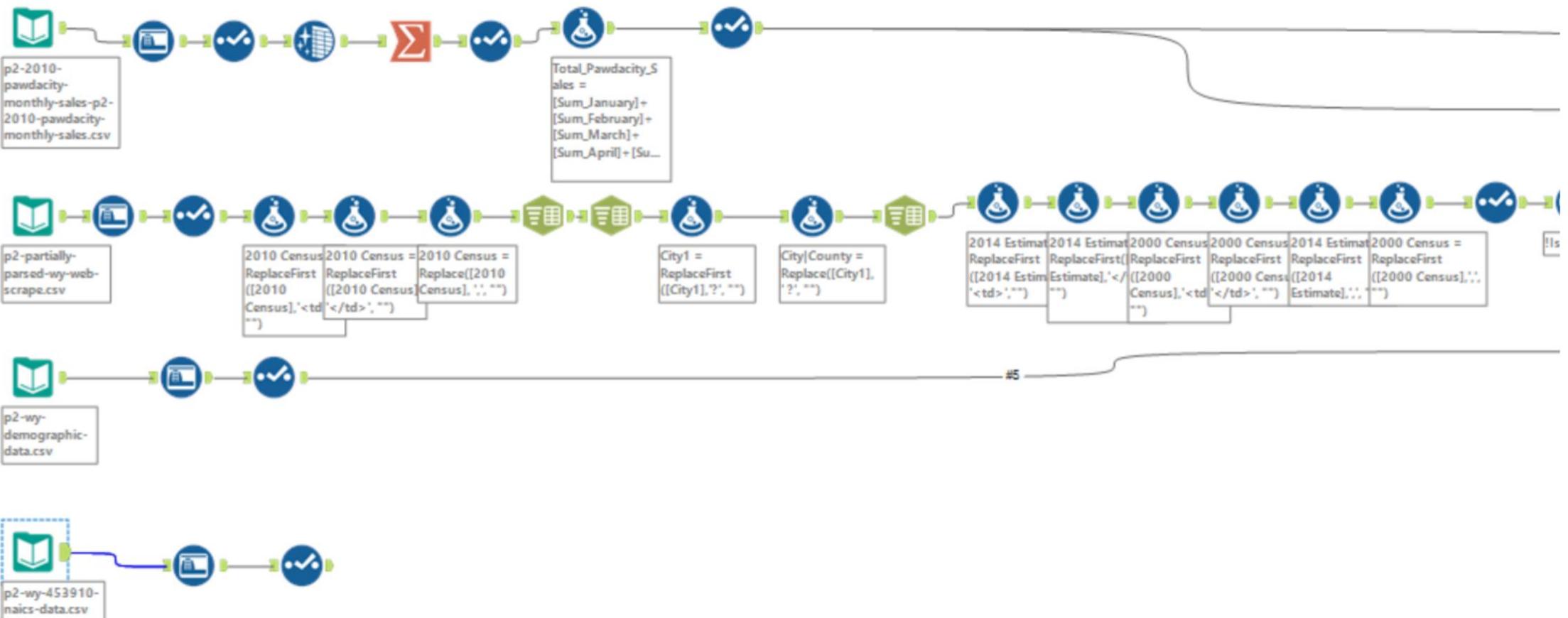
WEBSRAPING

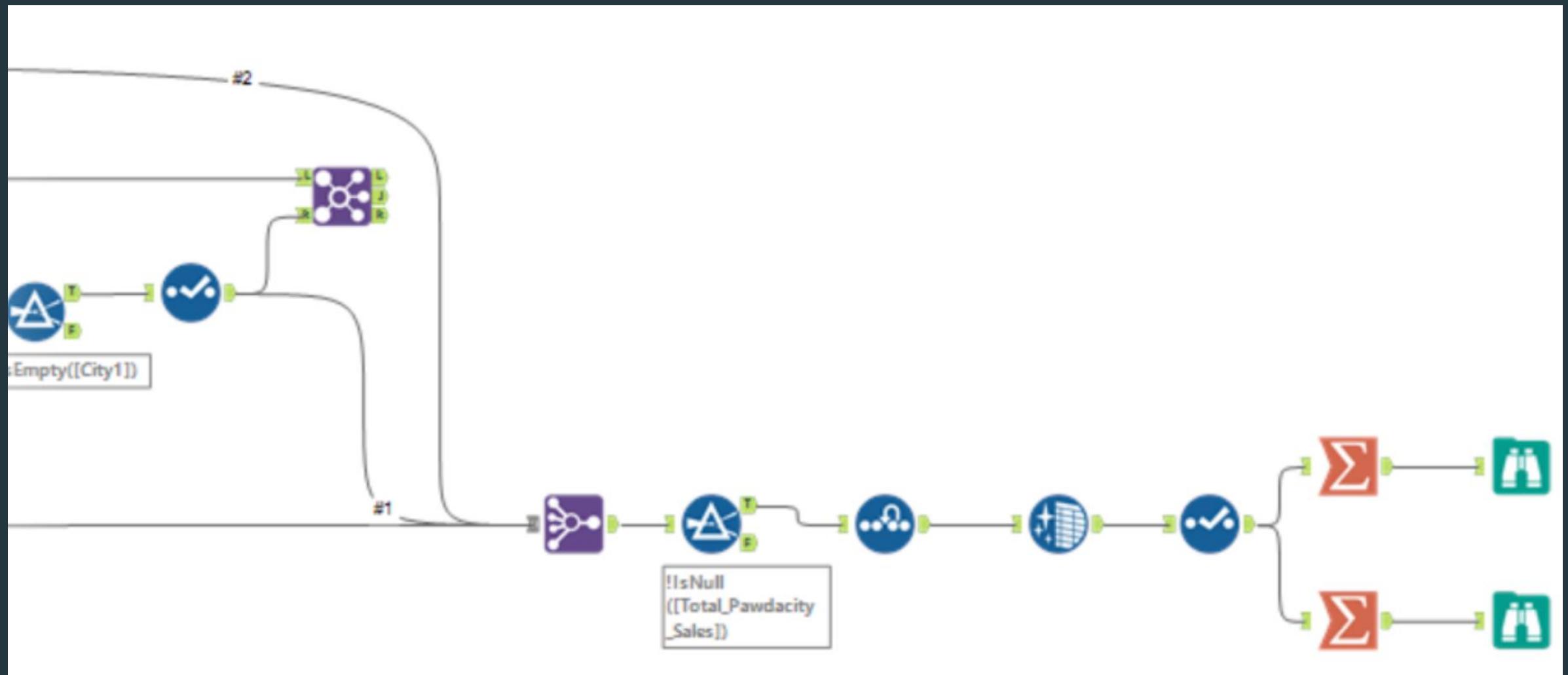
- Python and Alteryx
- We know you want to impress your boss and do web scraping
- Check firstly if your company allows you to scrape someone else's data
- Speak with your business risk team
- You may be breaching copyright laws under some company terms and conditions
- Some websites are locked down by organizations



LAB - ALTERYX

LIVE DEMO







CAREER ADVICE



MY TWO CENTS – DATA SCIENCE CAREER TIPS

- Meetup.com e.g data science, data engineering, Rladies, Sydney Women in Machine Learning and Data Science, Women who code, PyData, Docker, Kubernetes, AWS
 - Lunch & Learn
 - Hackathons - Data Science is a team sport
 - Data Science community – Get Involved
 - Data Science blogs
 - Learn Python, R and SQL
 - Internship

https://en.wikipedia.org/w/index.php?title=Michelle_Payne&oldid=900000000

Not logged in | Talk | Contributions | Create account | Log in

WIKIPEDIA The Free Encyclopedia

Article Talk Read Edit View history Search Wikipedia

DNA helix logo

Participate in an international science photo competition!

Michelle Payne

From Wikipedia, the free encyclopedia

Michelle J. Payne (born 29 September 1985)^[3] is an Australian jockey. She won the 2015 Melbourne Cup, riding Prince of Penzance, and was the first female jockey to win the event.

Contents

- 1 Early life
- 2 Career
 - 2.1 Melbourne Cup 2015
 - 2.2 2016–2017 seasons
- 3 Legacy
- 4 References
- 5 Further reading
- 6 External links

Early life

The youngest child of ten of Paddy and Mary Payne, Payne grew up on a farm at **Miners Rest**, a locality near **Ballarat** in central **Victoria, Australia**.^[4] Her mother Mary died in a motor vehicle crash when Payne was six months old, leaving her father Paddy to raise their ten children as a single father.^[5] Payne dreamt of being a winning jockey as a child, and, aged seven, told friends she would one day win the Melbourne Cup.^[6] She attended Our Lady Help of Christians primary school and Loreto College, Ballarat,^[6] and entered racing aged 15, the eighth of the Payne children to do so.^[6] She has Irish New Zealand heritage.

Career

She won in her first race at Ballarat, aboard **Reigning**—a horse trained by her father.^[7] In March 2001, Payne fell heavily at a race in Sandown Racecourse in Melbourne, fracturing her skull and bruising her brain. As a result of her prolonged recovery period—including a further fall where she fractured her wrist—Payne was granted a three-month extension to her apprenticeship to allow her time to ride out her claim.^[8]

Payne won her first Group One race, the **Tocra Handicap** at Caulfield Racecourse aboard **Allez Wonder** on 10 October 2009, and trainer **Bart Cummings** offered her the ride at the Caulfield Cup the following week. Payne was the third female jockey to ride in the Caulfield Cup.^[9] As a first-timer in the 2009 Melbourne Cup, she rode Cummings' **Allez Wonder**^[10] with a riding weight of 50.5 kg. The horse was placed 16th in the field of 23. In 2010 Payne rode **Yosef** to victory in the Thousand Guineas at Caulfield.^[11]

Melebourne Cup 2015

In 2015, she gained national attention when she rode the winning horse in two races at Melbourne Cup carnival at the Flemington Racecourse. One of them was the Hilton Hotels

Michelle Payne on Yosef, upon winning the 2010 Thousand Guineas at Caulfield

Occupation Jockey
Born 29 September 1985 (age 34)
Weight 50.5 kg (7.95 st; 111 lb)
Major racing wins Tocra Handicap (2009) - Allez Wonder Sires' Produce Stakes (2010) - Yosef The Thousand Guineas (2010) - Yosef Tattersalls' Trial (2011) - Yosef Melbourne Cup (2015) - Prince of Penzance

Racing awards • The Don Award (2016) - **Prince of Penzance**



Roles and responsibilities

- Design and development of solutions to greatly increase the efficiency of our internal operations
- Producing visualisations and automated reporting that can be presented or emailed to key stakeholders
- Designing automated notifications for internal and external stakeholders using real time analytics
- Operational analytics and fraud detection
- Helping to design any IT implementation required to enable the above to be executed

Skills & Experience

- Advanced programming skills, including in Python, REST APIs
- Advanced knowledge and experience in SQL
- Advanced knowledge and experience using AI / Machine Learning algorithms
- Advanced knowledge of analytics and data processing technologies and architectures.
- Strong analytical and problem solving skills
- Understanding of geospatial data related technologies.
- Experience scoping/running analytics projects
- Experience in analysing large volumes of unstructured data
- Experience in building analytics applications for deployment in real or near real time
- Clear communicator (verbal and written) in order to discuss complex information, analysis and findings to technical, and non-technical audiences
- Familiar with CSS, HTML, IMAP, Visual Basic, Node JS, XML, Tableau, Tosca or Selenium

Skills & Experience

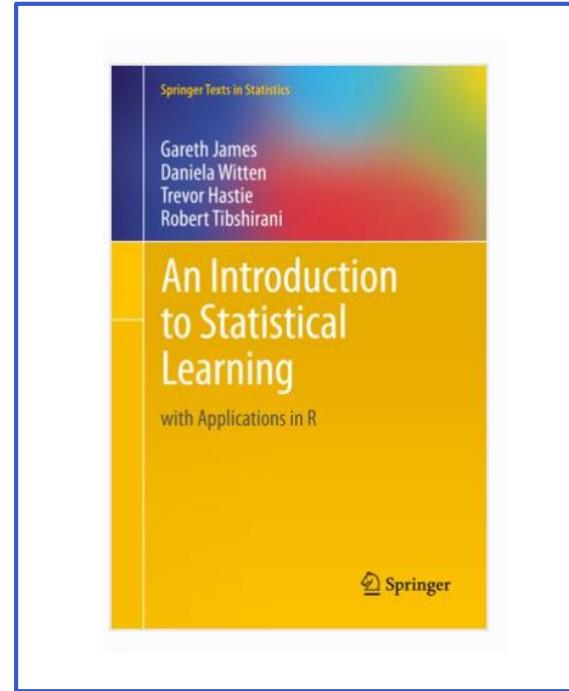
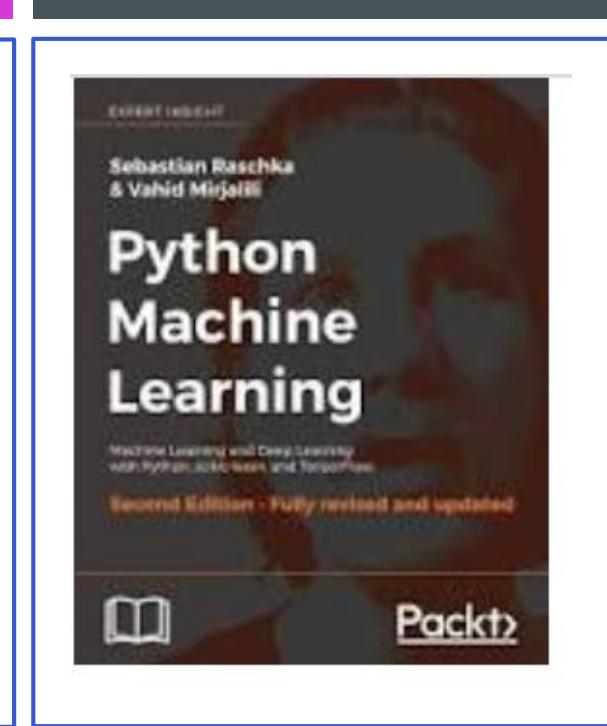
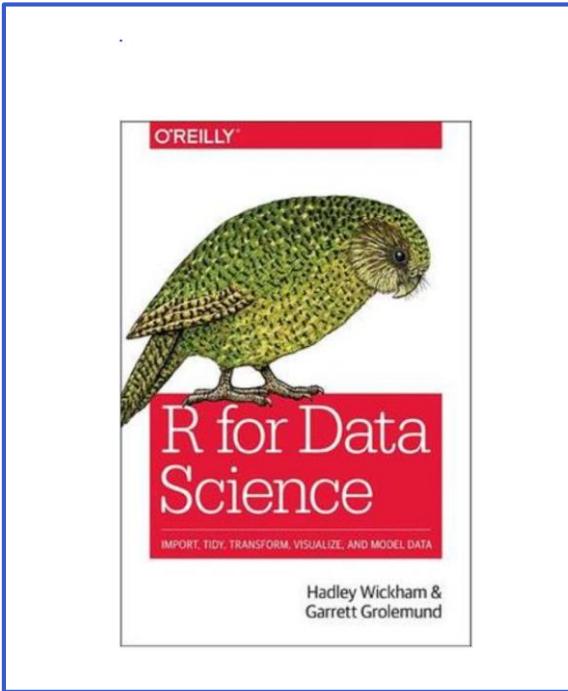
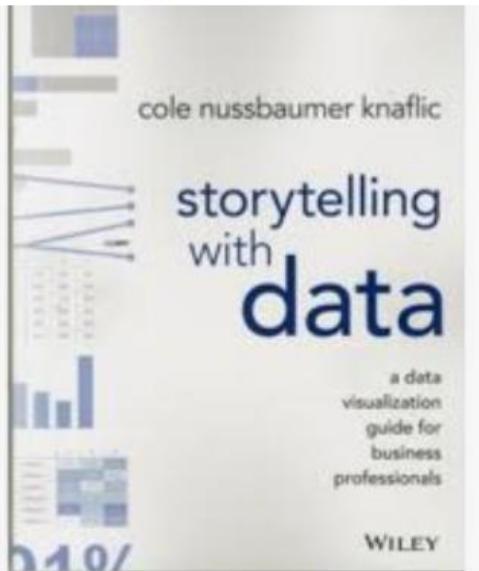
Data Science Interviews

Job Boards- Buyer Beware

- Ask the **recruiter** what duties you will be performing in your data scientist role
- Ask the **hiring manager** what your duties you will be performing in your data scientist role
- Do not proceed to a further interview if the hiring manager says '**yes you will be doing reporting and we may do data science in 18 months time**' - Run for your life!
- Keep on applying for data science roles, get a mentor, speak to the community, do you have peace?



RESOURCES



BOOKS I RECOMMEND



Data Science Central

Towards Data Science



GitHub

kaggle

HANDY RESOURCES

- <https://github.com/REMitchell/python-scraping>
- Tutorials
- Blogs
- Meetups
- Data Sets – AWS, Data.gov

Turn your Jupyter Notebook into slides

- File -> Download as -> Slides (slides.html)

In Windows

- A short tutorial for slide conversion in Mac:
- <https://medium.com/learning-machine-learning/present-your-data-science-projects-with-jupyter-slides-75f20735eb0f>

In Mac

TURN JUPYTER NOTEBOOK INTO SLIDES



THANK YOU



<https://www.linkedin.com/in/wendywong7/>



@abc_wendsss





Q & A