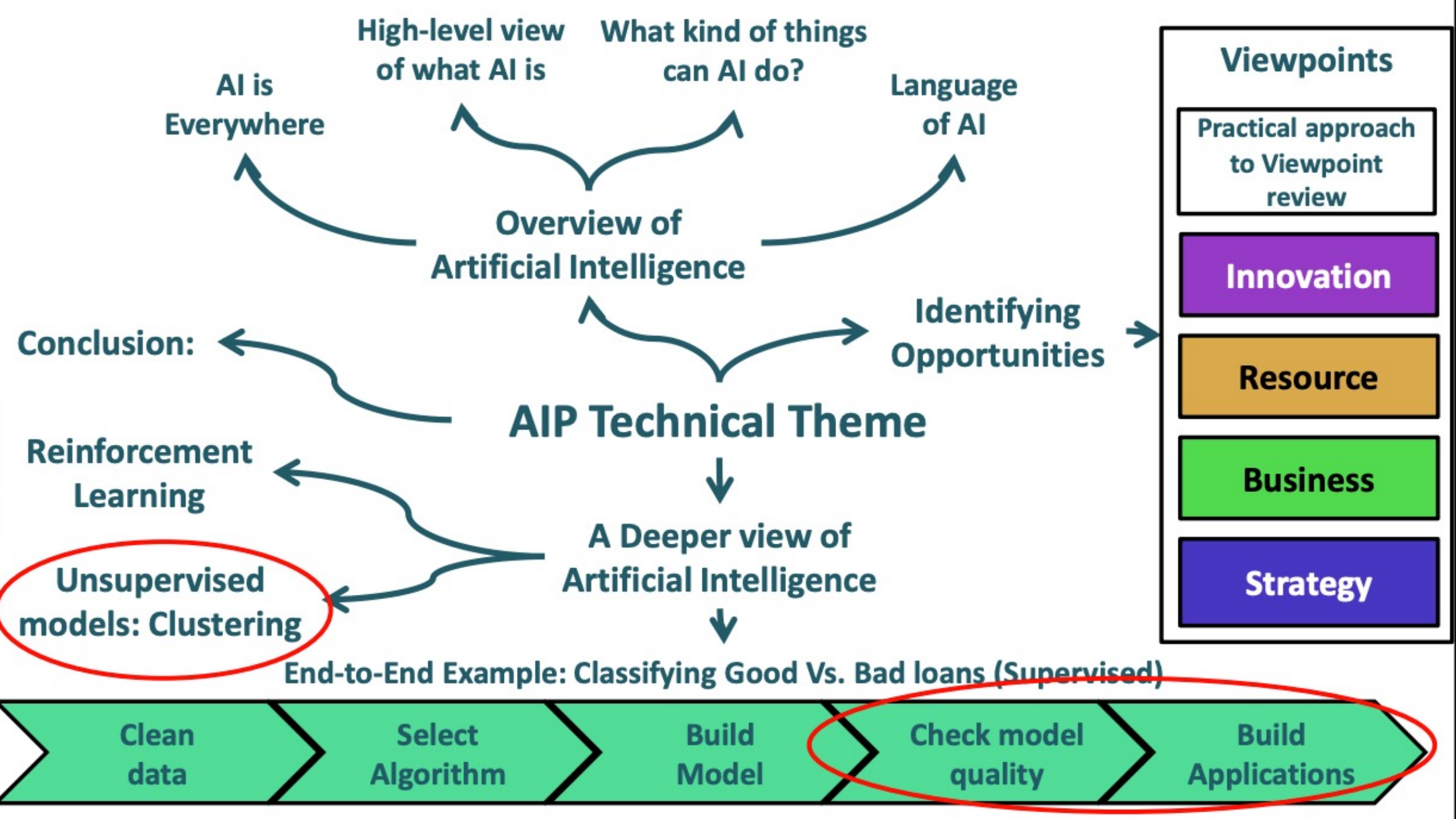
immerse
EDUCATION

Unsupervised Larning

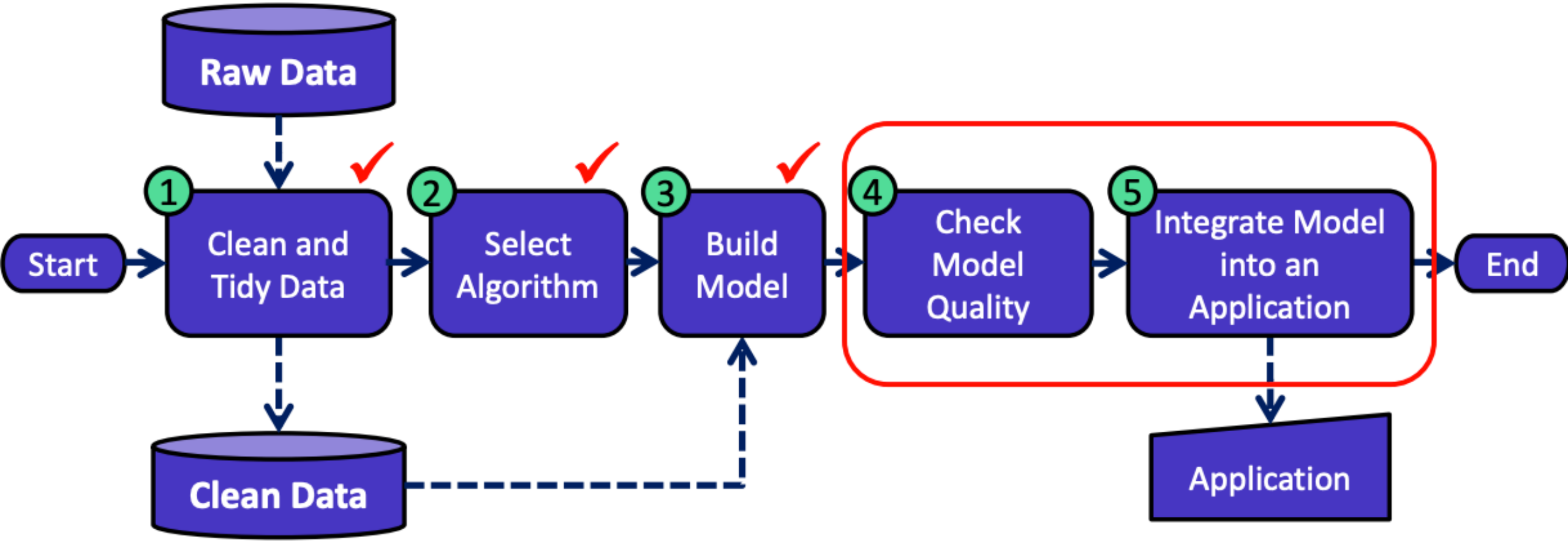
Elizabeth Savochkina | 7th June

Shape of a Day

Registration 9.00am to 9.30am
Unsupervised Learning 9.30am to 11.30am
Break 11.30am to 13.30 pm
Project Discussion: Opportunity Identification & Ideation 13.30am to 15.30pm


High-level view of what AI is
What kind of things can AI do?
AI is Everywhere
Language of AI
Overview of Artificial Intelligence
Conclusion:
Identifying Opportunities
AIP Technical Theme
Reinforcement Learning
A Deeper view of Artificial Intelligence
Unsupervised models: Clustering
End-to-End Example: Classifying Good Vs. Bad loans (Supervised)
Clean data
Select Algorithm
Build Model
Check model quality
Build Applications
Viewpoints
Practical approach to Viewpoint review
Innovation
Resource
Business
Strategy

Typical Workflow



Step 4 : Check Model Quality

Evaluate Model Quality

4 Check Model Quality

How can you be sure you have a 'good' model?

I.e. a model that gives predictions that are often correct?

Shoe Size	Hat Size	Body Weight	Supports Bayern Munich	Supports Manchester City	Supports Paris Saint-Germain	Likes pies for lunch	Likes Cats	Likes Dogs	Loan Default
35	30	102	0	0	1	0	0	1	1
41	23	106	0	0	0	0	0	1	0
37	18	136	0	0	0	0	1	0	1
41	16	183	0	0	0	1	0	0	0
44	22	19	0	0	0	1	0	0	1
42	26	7	0	0	0	1	0	0	1
36	30	17	0	0	0	1	1	1	1
36	13	150	0	0	0	1	0	1	1
45	10	105	0	0	0	1	1	0	0
32	27	134	0	0	0	0	1	0	0
37	22	137	0	0	0	1	1	1	1
36	13	149	0	0	0	1	0	1	0

Garbage in → Garbage Out

Different ways of being Wrong

4 Check Model Quality

The ‘confusion matrix’ is a useful way of reviewing model quality for category predictions

		Prediction	
		No Default	Default
Reality	No Default	Computer said "They will not default". We gave them a loan. They did not default. Good business!	Computer said "They will default". We didn't give them a loan. Lost business. Not good
	Default	Computer said "They will not default". We gave them a loan. They did default. Bad!	Computer said "They will default". We did not give them a loan. They would have defaulted in reality Good decision

Reviewing the Confusion Matrix

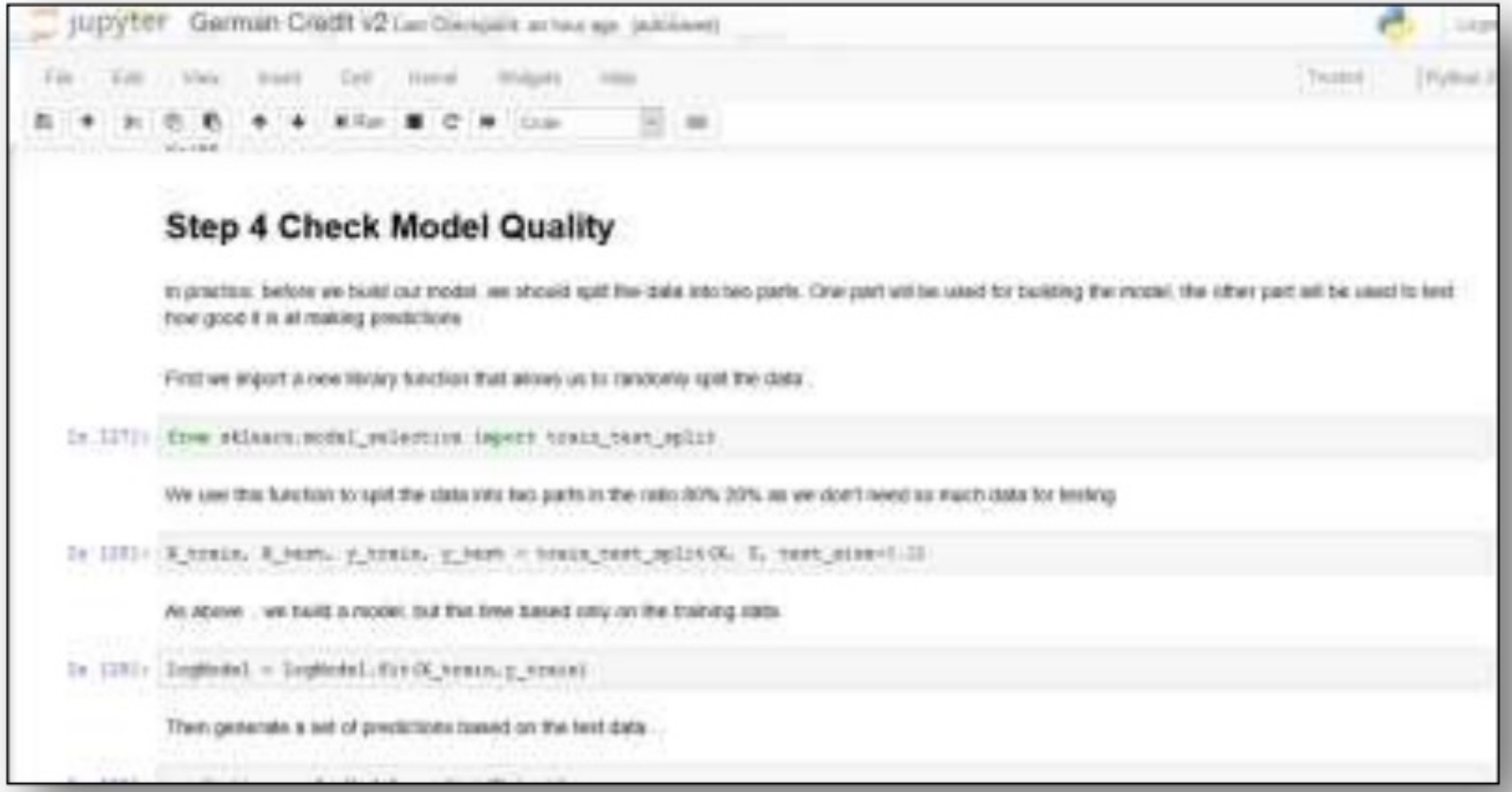
4 Check Model Quality

- In this case, a total of 155 predictions were correct (78%)
- A total of 45 were wrong ..
- Of which 15 were bad failures

		Prediction: No Default	Prediction: Default	
Reality	No Default	75	30	105
	Default	15	80	95
		90	110	200

Model Quality Check : Practical Example

Check Model Quality



Demonstration: “04-01 German Credit Model Quality Check.mp4”

Step 5 : Building an Application

Options for Building Applications

5 Integrate Model into an Application

- Building models often requires large amounts of computing power
- But running them as applications may need far fewer resources ..
  - Enabling ML apps to run on smartphones and laptop computers

Run the model directly on a desktop computer or smartphone

Run the model on a server – and access remotely

Building Models into Applications

5 Integrate Model into an Application

- Regression models are 'formulae' that can be copied into software code and evaluated
- Decision trees might be implemented as a series of 'If .. Then .. Else' statements
- Library code is available to evaluate Neural Networks on smaller processors

$$Y = k + A.x_1 + B.x_2 + C.x_3 \ldots Z.x_n$$

Variable Name	Constant	Number of Rooms	Number of Bathrooms	Garden area	Detached	Semi-detached	Terrace
Model Parameters	50,000	20000	3760	4.67	47577	32432	23543
Applicant data	1	6	2	1200	1	0	0
Result	50000	120000	7520	5604	47577	0	0

House predicted value £230,701

'linear' (straight-line) model built by multiplying and adding terms

Build the Application

5 Integrate Model into an Application

- In this case the model is simply a list of coefficients for a relatively simple equation ..

```
print( logModel.coef_)
print( logModel.intercept_)
```

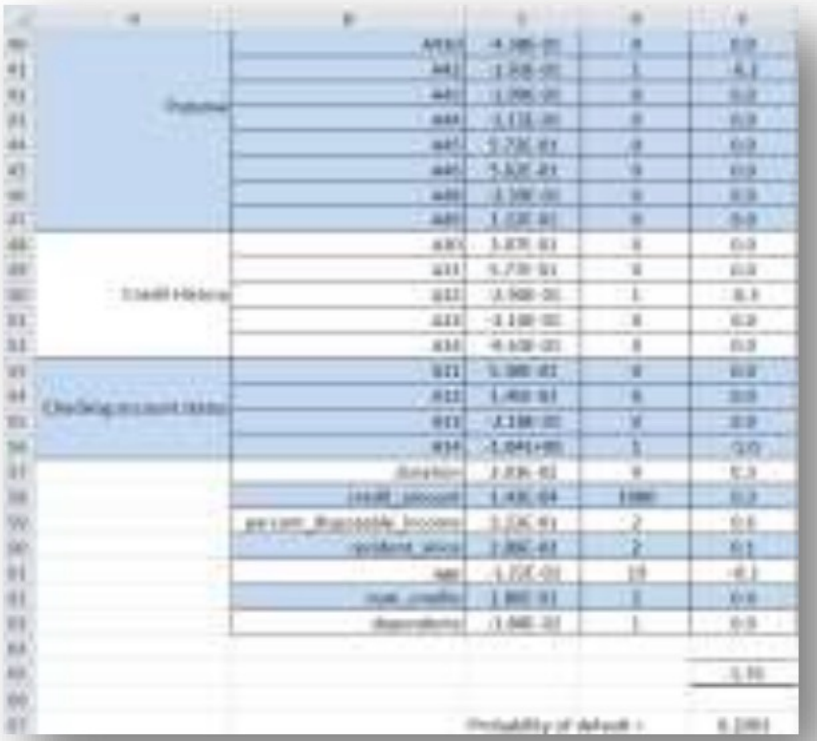
```
[[ 1.76294369e-01 -8.56571126e-01 -2.10532764e-01 -4.67743993e-01
  -5.32855166e-01 -8.92711403e-02 -1.48284746e-02 -4.13219760e-02
   2.09198704e-01 -2.92896634e-01 -5.94578827e-01 -1.15009343e-02
  -4.76317246e-02 -6.19144098e-01 -4.00699728e-01 -1.99907975e-01
  -3.53242688e-01  2.75573634e-01 -1.03105672e-01  6.34713643e-02
  -6.38642449e-01  1.75219375e-01 -1.83209646e-01 -4.74459892e-01
  -1.95826594e-01  3.47136666e-01  1.02540582e-01 -9.87155603e-02
  -6.88921888e-01 -3.40316556e-01  3.29246259e-01 -1.40511096e-01
  -2.91350624e-03 -4.58538784e-01 -4.05559630e-01  5.88994290e-01
  -1.03965156e+00 -4.37697991e-01 -1.92838329e-01 -1.99360467e-01
  -3.14801370e-01  5.71761336e-01  5.62232677e-01 -3.39359854e-01
   1.22444517e-01  3.07173187e-01  5.76979945e-01 -2.89642473e-01
  -3.09780167e-01 -9.63007250e-01  5.29671486e-01  1.45249041e-01
  -3.17664937e-01 -1.03553235e+00  3.03348301e-02  1.42773085e-04
   3.22047587e-01  2.86347603e-02 -1.21770781e-02  1.80129775e-01
  -3.87951340e-02]]
[-0.67827676]
```

Demonstration: “04-02 - Export model params.mp4”

Embed the Model into an Application

5 Integrate Model into an Application

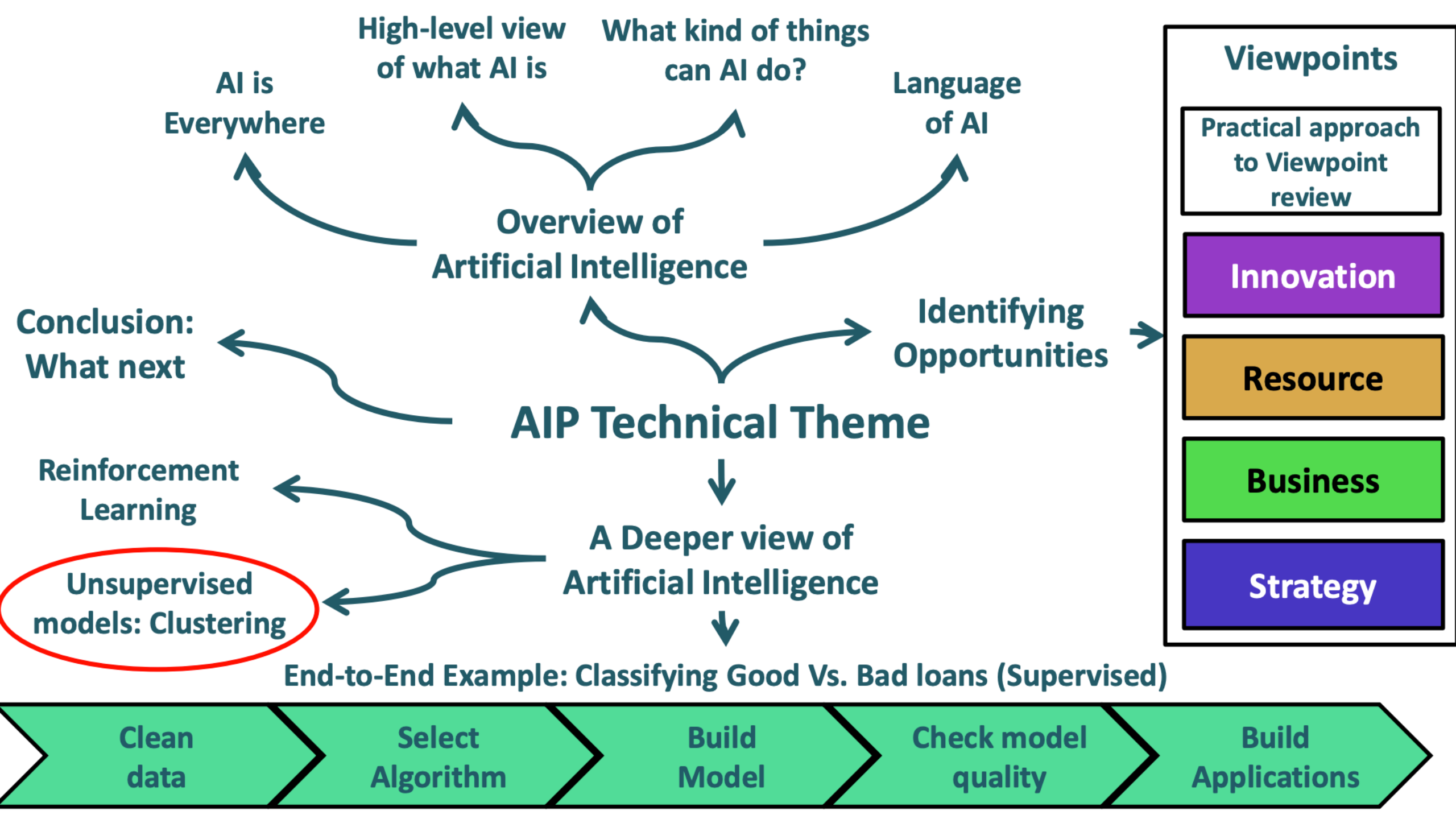
- Once you have the model, you can embed it into other software
- In this case, I have reproduced the model in Excel



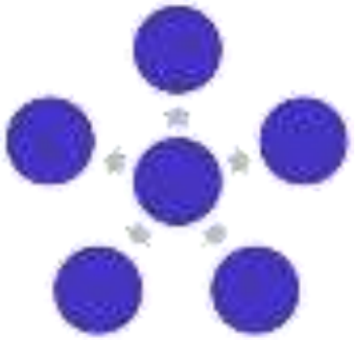
Demonstration: "04-03 - Build an Application.mp4"

Microsoft Excel Worksheet

Unsupervised Models : Clustering


High-level view of what AI is
What kind of things can AI do?
AI is Everywhere
Language of AI
Overview of Artificial Intelligence
Conclusion: What next
Identifying Opportunities
AIP Technical Theme
Reinforcement Learning
Unsupervised models: Clustering
A Deeper view of Artificial Intelligence
End-to-End Example: Classifying Good Vs. Bad loans (Supervised)
Clean data
Select Algorithm
Build Model
Check model quality
Build Applications
Viewpoints
Practical approach to Viewpoint review
Innovation
Resource
Business
Strategy

A Brief Reminder: What is Clustering?



Clustering

Dividing a large group into smaller groups containing members that are similar to each other

- Segmenting customers into groups
- Grouping products into families that customers will relate to
- Grouping complaint types

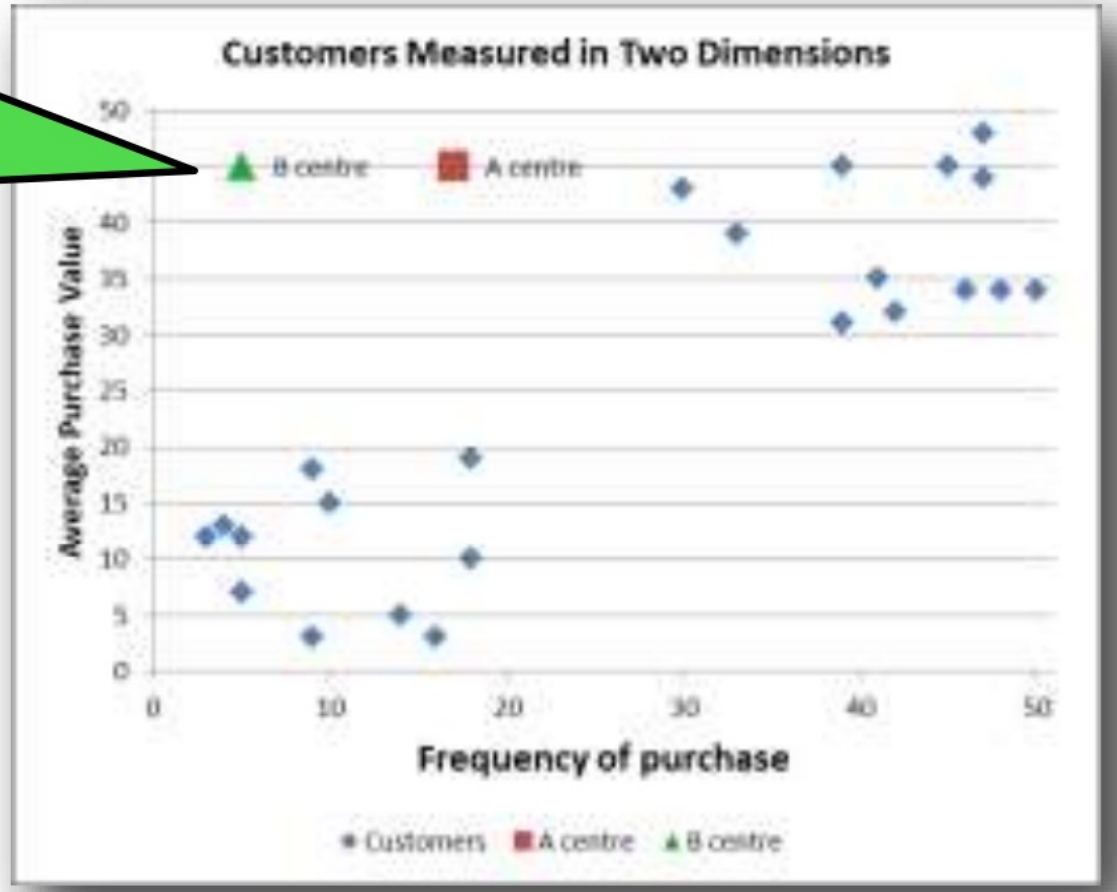
A Simple and Fast Clustering Algorithm

- Here we have a simple example:
  - There are only two dimensions
  - We can easily see two groups
- But the following algorithm, whilst simple, can:
  - Deal with many dimensions ..
  - Divide data into as many groups as we decide
  - Make a decision even if the groups are similar

Step 1 : Take a Guess

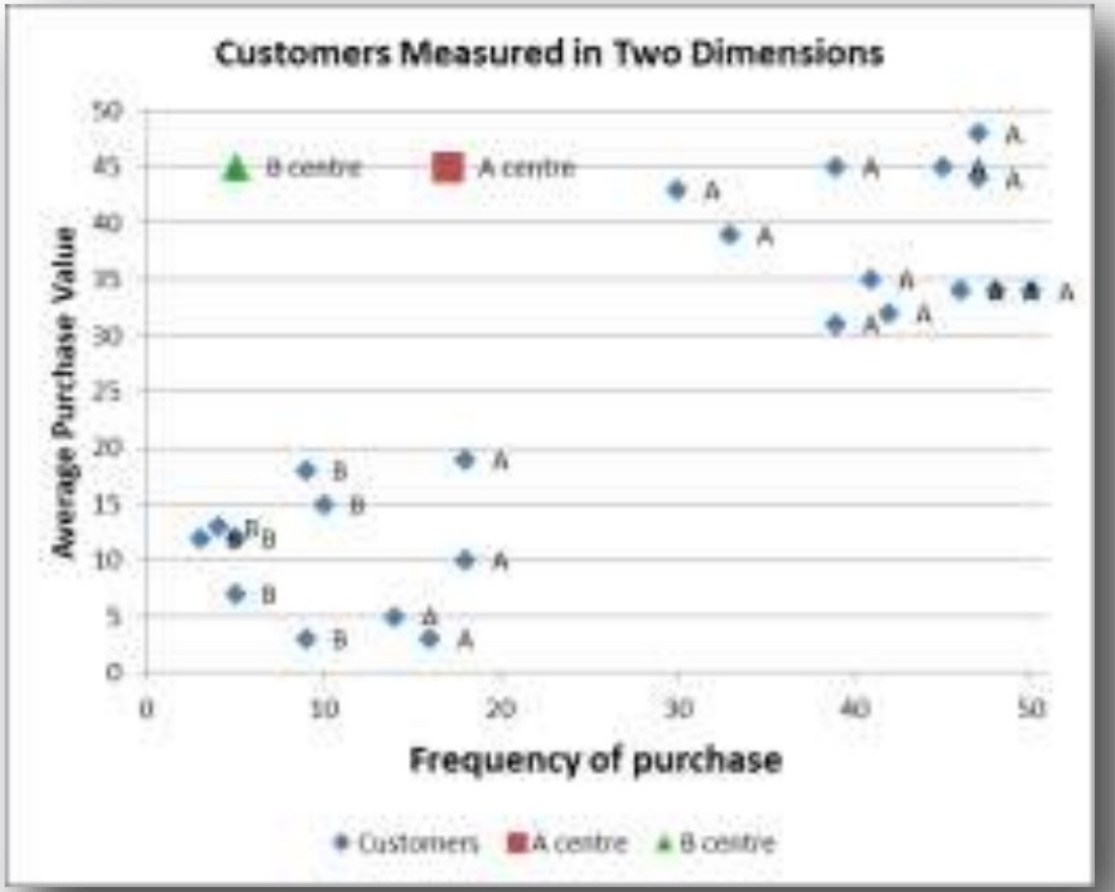
- We want to divide this data into two groups which we will call ‘A’ and ‘B’
- First, we select two, completely random points as a guess at the centre of each group

These starting points can be anywhere



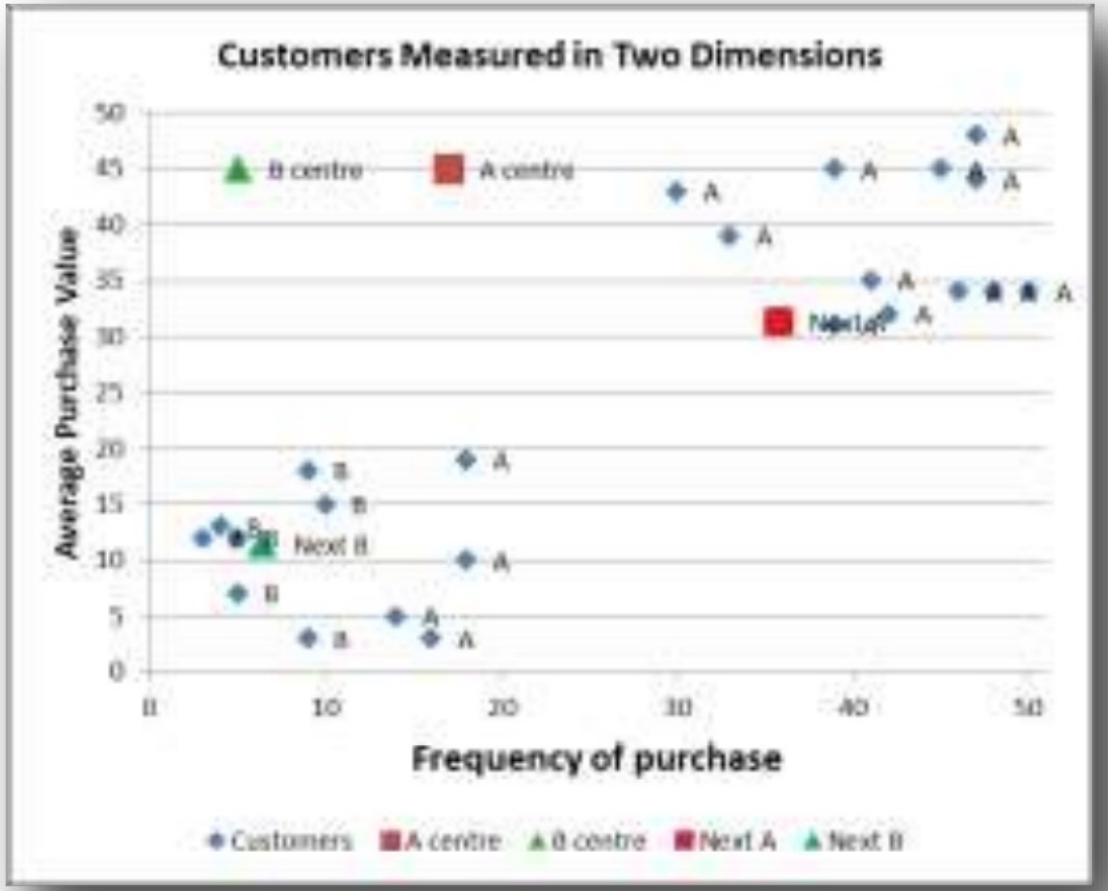
Step 2 : Which Guess is each Point Closest to?

- Measure the distance every point and each of the guesses
- If a point is closest to guess ‘A centre’ – then label it ‘A’
- If the point is closest to guess ‘B centre’ – then label it ‘B’



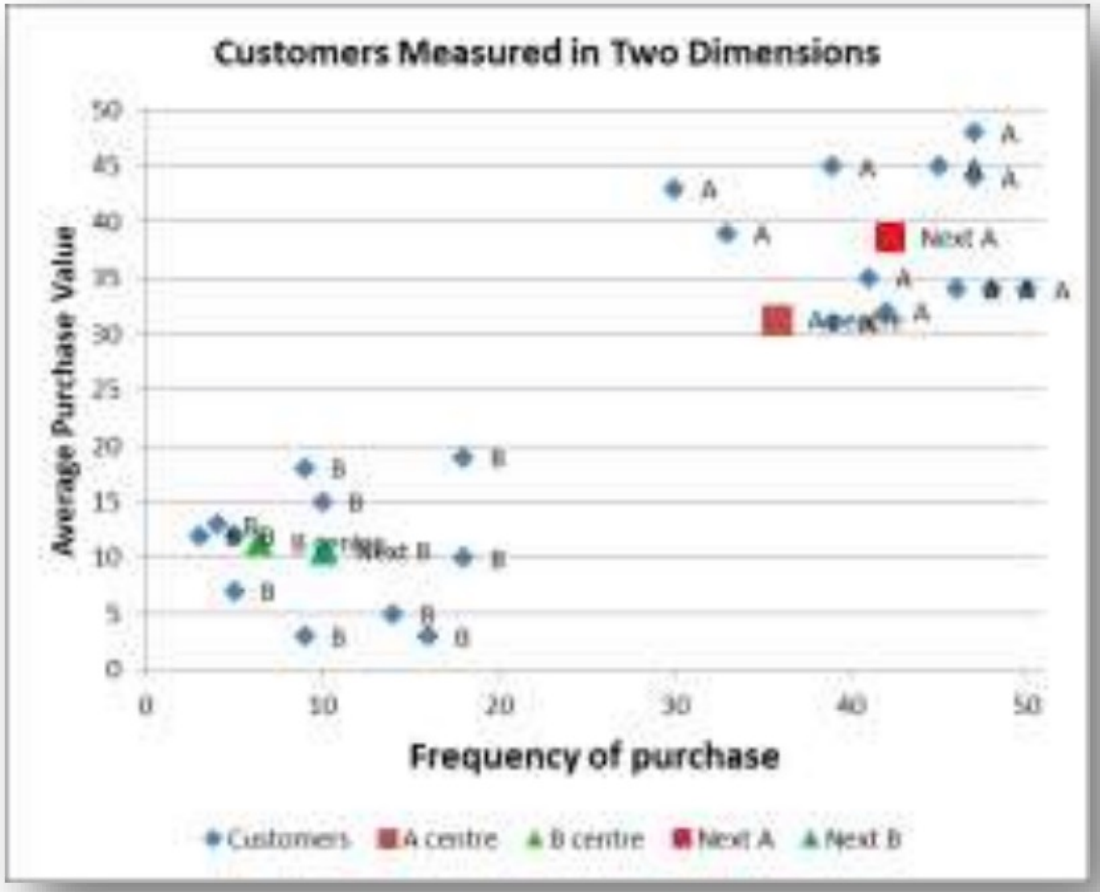
Step 3 : Find the Centre of Each Group

- Find the centre of all of the ‘A’s
  - Call this ‘Next A’
- Find the centre of all of the ‘B’s
  - Call this ‘Next B’



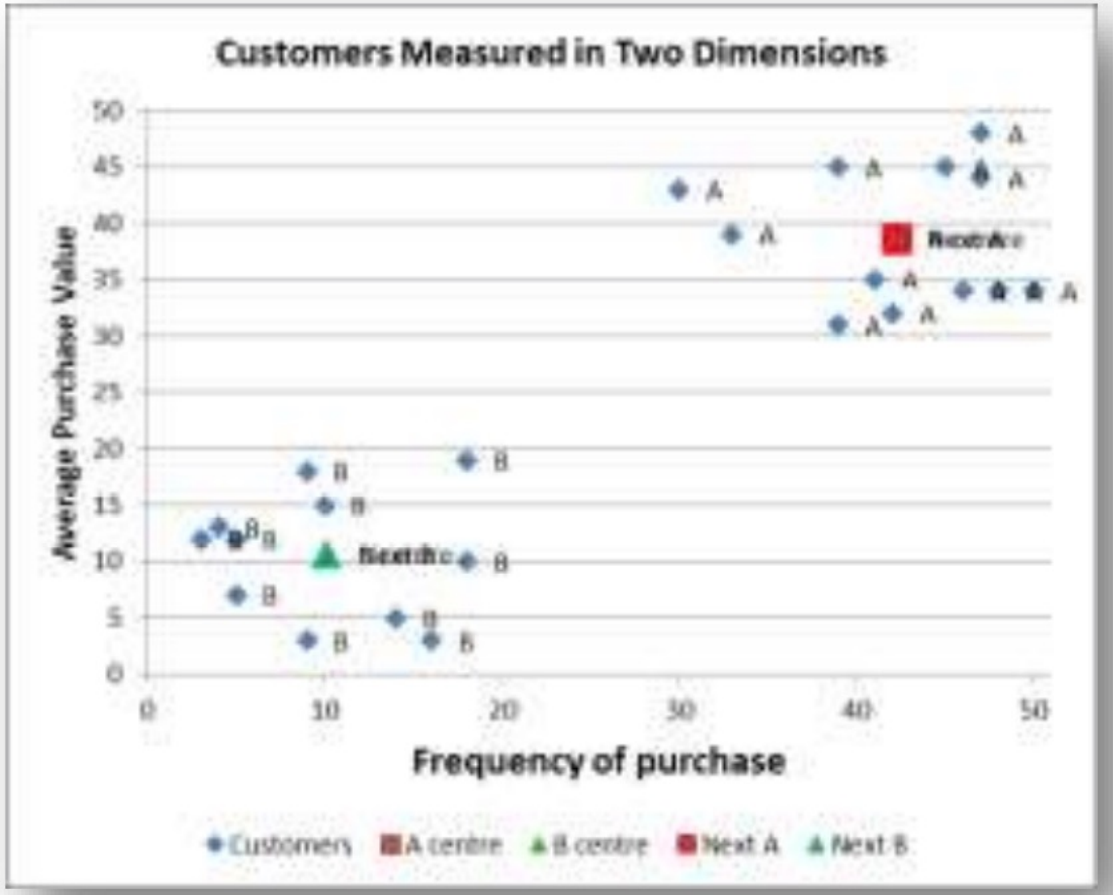
Step 4 : Use those Points as your New Best Guess

- Those centre points become the new ‘best guesses’



Repeat Steps 2 .. 4

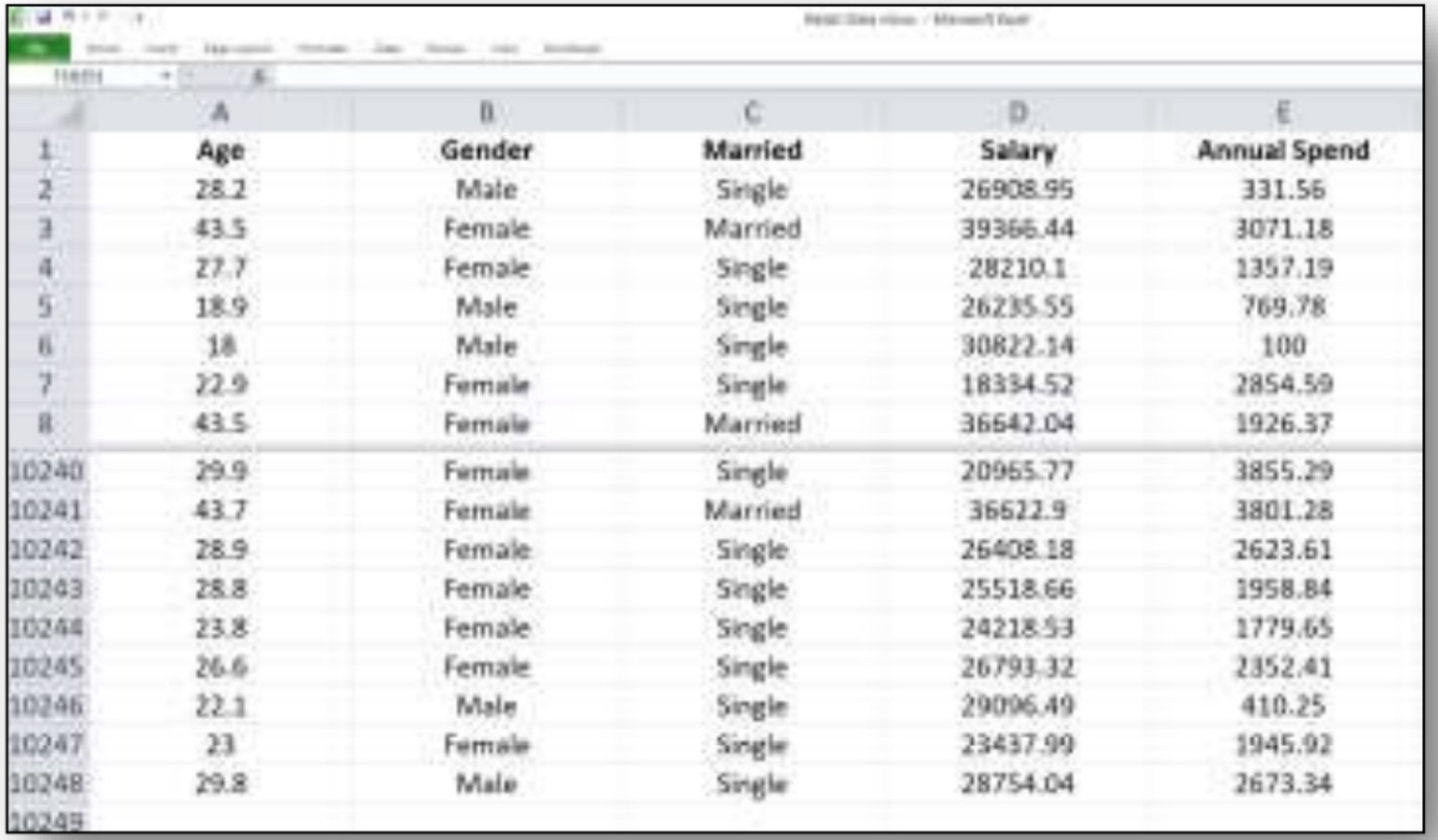
- This simple procedure is then repeated
- Each time it is repeated the data falls into more distinct groups
- After a few repeats our ‘guesses’ are actually at the centre of the group and the set has been divided



Clustering : Practical Example

Identifying Customer Groups using Clustering

- Our source data is over 10,000 customer records
- 5 factors for each customer
- We want to identify groups of customers based on the data



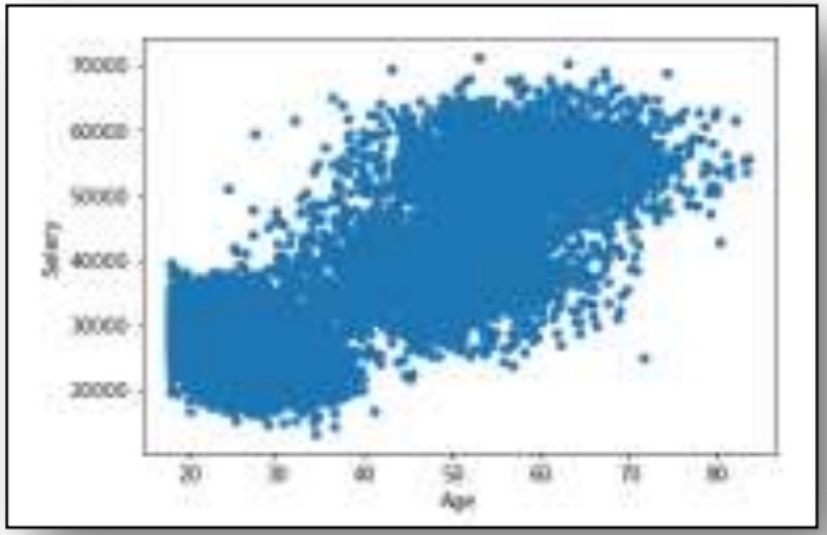
	A	B	C	D	E
1	Age	Gender	Married	Salary	Annual Spend
2	28.2	Male	Single	26908.95	331.56
3	43.5	Female	Married	39366.44	3071.18
4	27.7	Female	Single	28210.1	1357.19
5	18.9	Male	Single	26235.55	769.78
6	18	Male	Single	30822.14	100
7	22.9	Female	Single	18334.52	2854.59
8	43.5	Female	Married	36642.04	1926.37
10240	29.9	Female	Single	20965.77	3855.29
10241	43.7	Female	Married	36622.9	3801.28
10242	28.9	Female	Single	26408.18	2623.61
10243	28.8	Female	Single	25518.66	1958.84
10244	23.8	Female	Single	24218.53	1779.65
10245	26.6	Female	Single	26793.32	2352.41
10246	22.1	Male	Single	29096.49	410.25
10247	23	Female	Single	23437.99	1945.92
10248	29.8	Male	Single	28754.04	2673.34
10249					

Import the Data and Take a look …

```
customer_data = pd.read_csv("Retail Data v4.csv")
```

```
print(customer_data[0:5])
```

	Age	Gender	Married	Salary	Annual Spend
0	28.2	Male	Single	26908.95	331.56
1	43.5	Female	Married	39366.44	3071.18
2	27.7	Female	Single	28210.10	1357.19
3	18.9	Male	Single	26235.55	769.78
4	18.0	Male	Single	30822.14	100.00




```
customer_data.plot.scatter('Age', 'Salary')
```

Build the Model and Review Results

```
# create kmeans object
kmeans = KMeans(n_clusters=4)

# fit kmeans object to data
kmeans.fit(standardized_customer_data_df)

# save new clusters for chart
y_km = kmeans.fit_predict(standardized_customer_data_df)
```

```
print(customer_data[y_km==3][0:10])
```

	Married	Single	Female	Male	Age	Salary	Annual Spend
2	0	1	1	0	27.7	28210.10	1357.19
5	0	1	1	0	22.9	18334.52	2854.59
9	0	1	1	0	27.2	21901.08	566.21
11	0	1	1	0	31.0	24944.10	1661.97
12	0	1	1	0	34.9	23136.51	1363.36
14	0	1	1	0	32.8	21680.78	2600.72
15	0	1	1	0	28.7	27930.64	3717.44
17	0	1	1	0	28.0	25404.68	756.29
19	0	1	1	0	30.4	30677.96	1120.23
20	0	1	1	0	29.9	25182.12	1391.35

Chart the Results