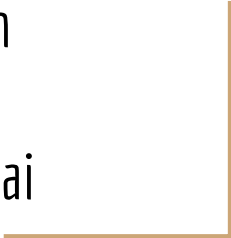




# PyPandas

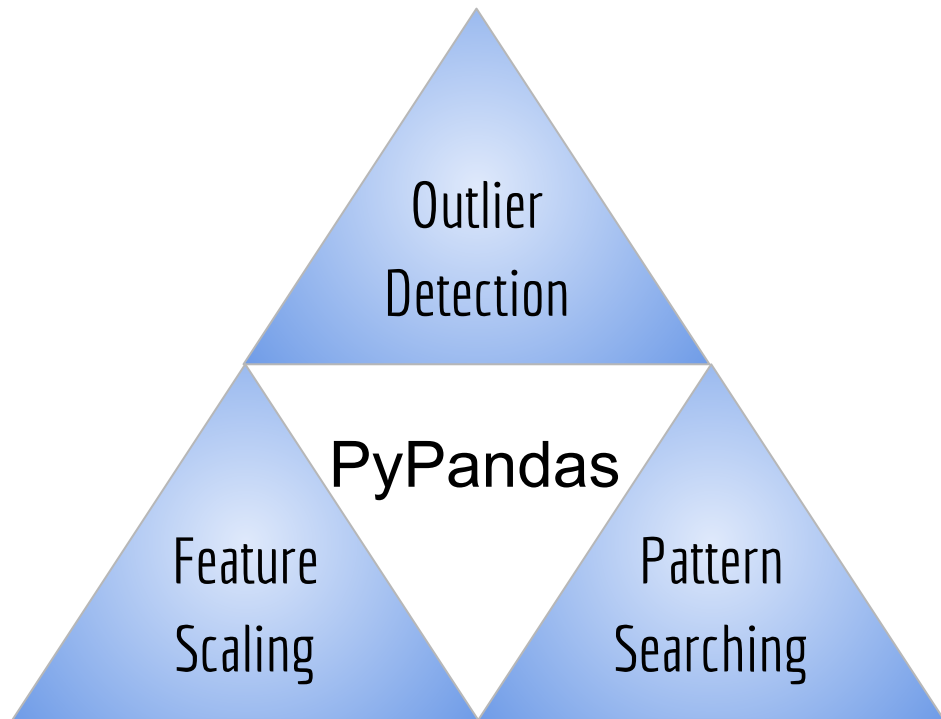
Chia-Hsien Lin  
Pei-Lun Liao  
Shang-Hung Tsai



# Introduction

PyPandas is a data cleaning toolkit built specifically for the Spark framework.

- PySpark ML module integration
- Work with PySpark Dataframe
- Easy installation
- User-friendly APIs
- Scalable



# Outlier Detection

- clustering algorithms to detect outliers
  - KMeans
    - Euclidean distance
  - Gaussian Mixture Model
    - Mahalanobis distance

# Outlier Detection

## Generic API for different OutlierRemovers

```
model = OutlierRemover.factory("kmeans")
```

```
model.fit(dataframe, ["Initial Cost",  
                      "Total Est Fee"])
```

```
model.summary()
```

cluster index	size	cluster center	avg(distance to cluster center)
0	1506910	86153.79855969973	115623.10175225197
1	11	3.7746609272727275E8	9.78881126369113E7
2	3	9.55543933E8	3.703218131733947E7
3	1206	1.6625062420529801E7	7289962.118664694
4	47	1.1278780606382978E8	3.6478952700282976E7

```
model.get_cluster(3)
```

prediction	distance to cluster center	Initial Cost	Total Est Fee
3	4775293.760860619	1.185E7	122143.5
3	2216883.564916434	1.4408275E7	148501.2
3	8191841.731979836	8433633.0	86958.7
3	8186474.485411539	8439000.0	87010.2

```
model.filter(3, 10000000)
```

# Normalization and Scaling

```
df = spark.createDataFrame(  
  [(1, 1.22, 2.36, 0.11),  
   (2, 0.22, 1.34, 0.12),  
   (3, 0.32, 0.37, 1.12)],  
  ["id", "col1", "col2", "col3"])
```

```
scaled_df = standard_scale(df, ["col1", "col2", "col3"])  
scaled_df.show()
```

```
scaled_df = min_max_scale(df, ["col1", "col2", "col3"])  
scaled_df.show()
```

```
scaled_df = max_abs_scale(df, ["col1", "col2", "col3"])  
scaled_df.show()
```

```
normalized_df = normalize(df, ["col1", "col2", "col3"])  
normalized_df.show()
```

- Standard scale

$$S(X) = \frac{X - \mu}{\sigma} \in \text{proper deviation range}$$

where  $\mu$  is the mean of  $X$  and  $\sigma$  is the standard deviation of  $X$

- Min-Max scale

$$S(X) = \frac{X - \min(X)}{\max(X) - \min(X)} \in [0.0, 1.0]$$

- Max-Abs scale

$$S(X) = \frac{X}{\max(X)} \in [-1.0, 1.0]$$

- P-norm Normalization

$$S(X) = \frac{X}{\|X\|_p} \in [-1.0, 1.0]$$

# Pattern Searching and Replacement

Common Pattern	Regular Expression
URL	URL regular expression[6]
Leading space	' ^ +'
Trailing space	' +\$'
Consecutive space	' +'
Number	' \d+'
Not a word	' [^\w\d\s]+'
Blank	' _+'

# Pattern Searching and Replacement

Generic API for pattern searching and replacement

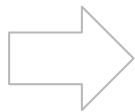
```
# text cleaning with common patterns  
clean_text(dataframe, columns)
```

```
# text cleaning with customized regular expression  
sub_with_pattern(dataframe, columns, regexp, to_replace)
```

The cheap thing on **www.amazon.com**

See, iPhoneX only cost **\$20!!!**

What the \_\_\_\_\_ :))



The cheap thing on **\_url\_**

See iPhoneX only cost **\_number\_**

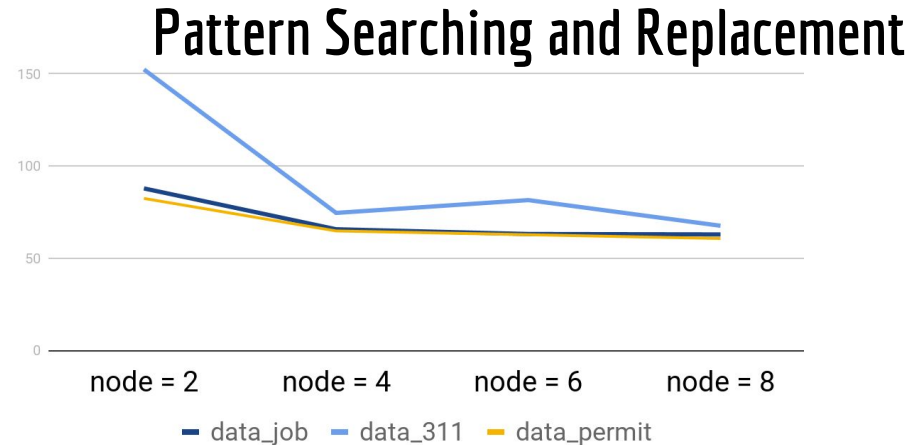
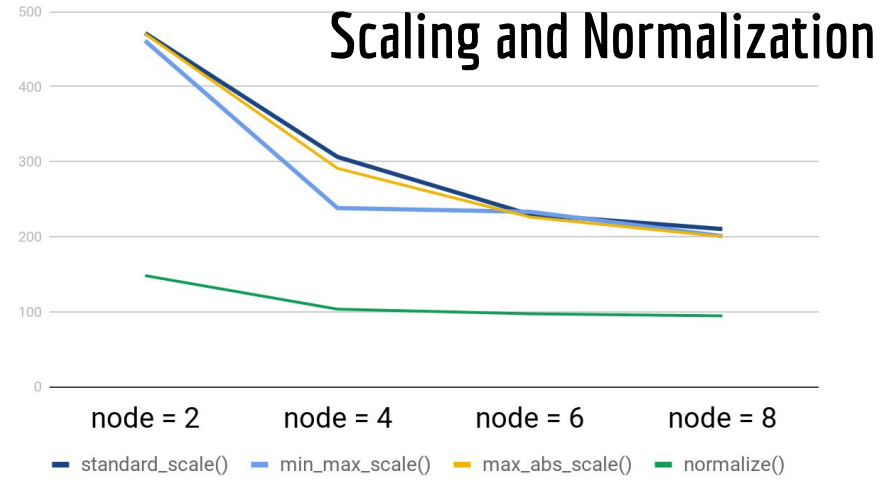
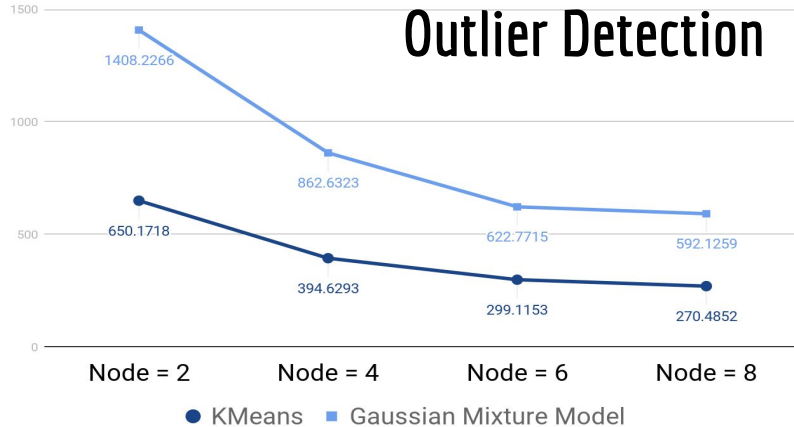
What the \_

# Features comparison

	PySpark DataFrame[20]	PyPandas	Optimus[15]	SparkingPandas[17]	Dask[24]
Basic data cleaning					
Missing value filling	✓	✓	✓		✓
Missing value removing	✓	✓	✓		✓
Duplication removing	✓	✓	✓		✓
Outlier Detection					
Deviation with median			✓		
Clustering algorithms		✓			
Scaling and Normalization					
Standard scaling		✓			✓
Scale in range		✓	✓		✓
Normalization		✓	✓		
Text Cleaning					
Special character removing	✓	✓	✓		
Pattern searching	✓	✓	✓		
Pattern searching and replacement		✓	✓		
Common patterns replacement		✓			



# Scalability Experiments



# Dataset

	(Row, Col)	Size	Property
311 Service Requests	(9M, 53)	6.01 GB	Text data
Permit Issuance	(3M, 60)	1.43 GB	Mixed Type data
Job Application Filings	(5M, 89)	2.88 GB	Numerical data

# Runtime Experiments

Normalization and Scaling		
	Min-Max Scaling	Normalization
PyPandas	391.11s	105.90s
Optimus	142.46s	61.92s

Special characters Cleaning			
	311 Service Requests	Permit Issuance	Job Application Filings
PyPandas	104.18s	62.35s	60.07s
Optimus	86.19s	43.67s	43.04s

# Conclusion

- Useful and commonly used data cleaning features
- Easy installation and usage.
- Good performance and scalability.
- <https://github.com/shtsai7/PyPandas>