



香港城市大學
City University of Hong Kong

Parallel computing on Bike Sharing Demand Dataset

CS4480 Group Project – Group 14

LI Yiheng, 56641664

LUO Peiyuan, 56642728

ZHOU Xin, 56644501



CONTENTS

1. Introduction & Overview of Dataset
2. EDA
3. Data Pre-processing Stream
4. Data Pre-processing
5. Machine Learning with Scala
6. Deep Learning with Python



Data Preprocessing Demand Dataset

	date	hour	year	month	weekday	season	holiday	workingday	weather	temp	atemp	humidity	windspeed	casual	registered	count
0	2011-01-01	0	2011	1	7	1	0	0	1	9.84	14.395	81	0.0	3	13	16
1	2011-01-01	1	2011	1	7	1	0	0	1	9.02	13.635	80	0.0	8	32	40
2	2011-01-01	2	2011	1	7	1	0	0	1	9.02	13.635	80	0.0	5	27	32
3	2011-01-01	3	2011	1	7	1	0	0	1	9.84	14.395	75	0.0	3	10	13
4	2011-01-01	4	2011	1	7	1	0	0	1	9.84	14.395	75	0.0	0	1	1

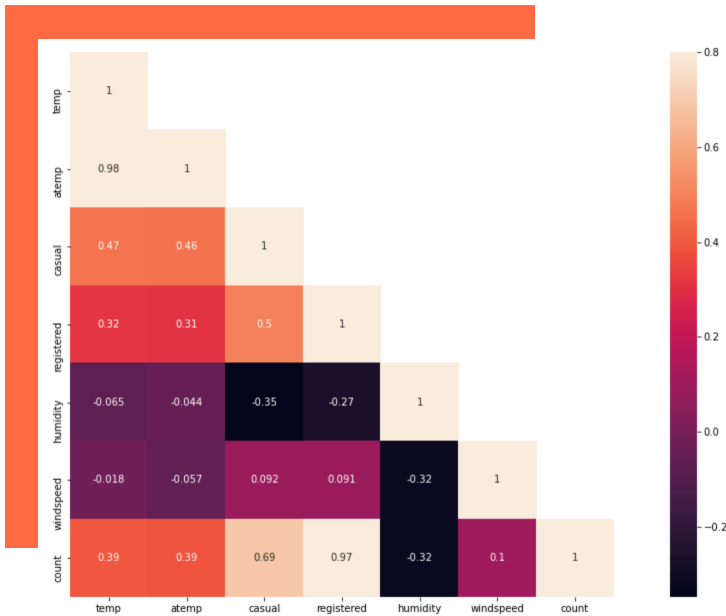
Two-year historical log corresponding to years 2011 and 2012 from Capital Bikeshare system, Washington D.C., USA

Corresponding weather and seasonal information extracted from *freemeteo*



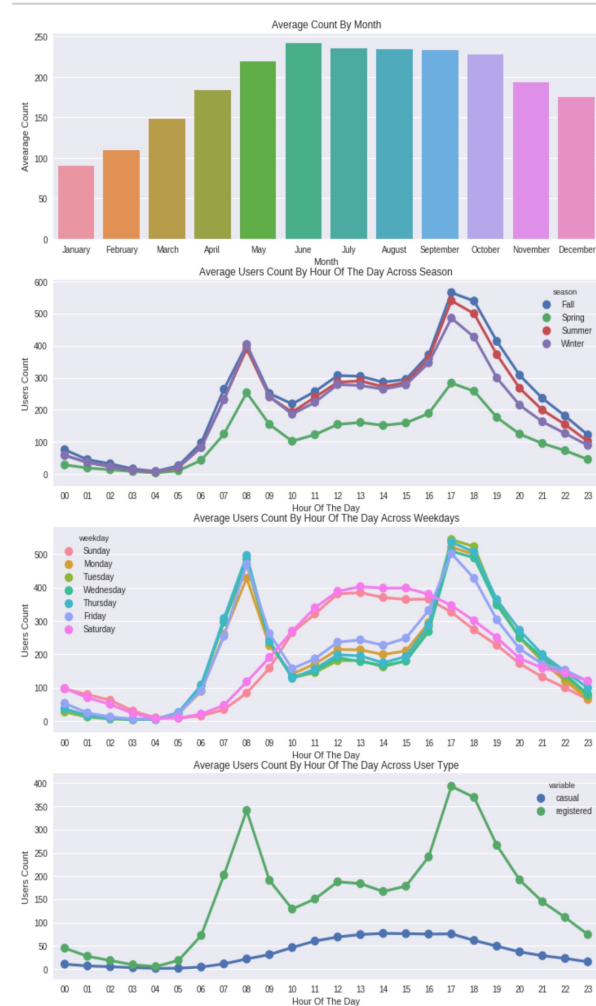
Data Source — <http://capitalbikeshare.com/system-data>.
<http://www.freemeteo.com>.

Exploratory Data Analysis



Correlation Analysis:

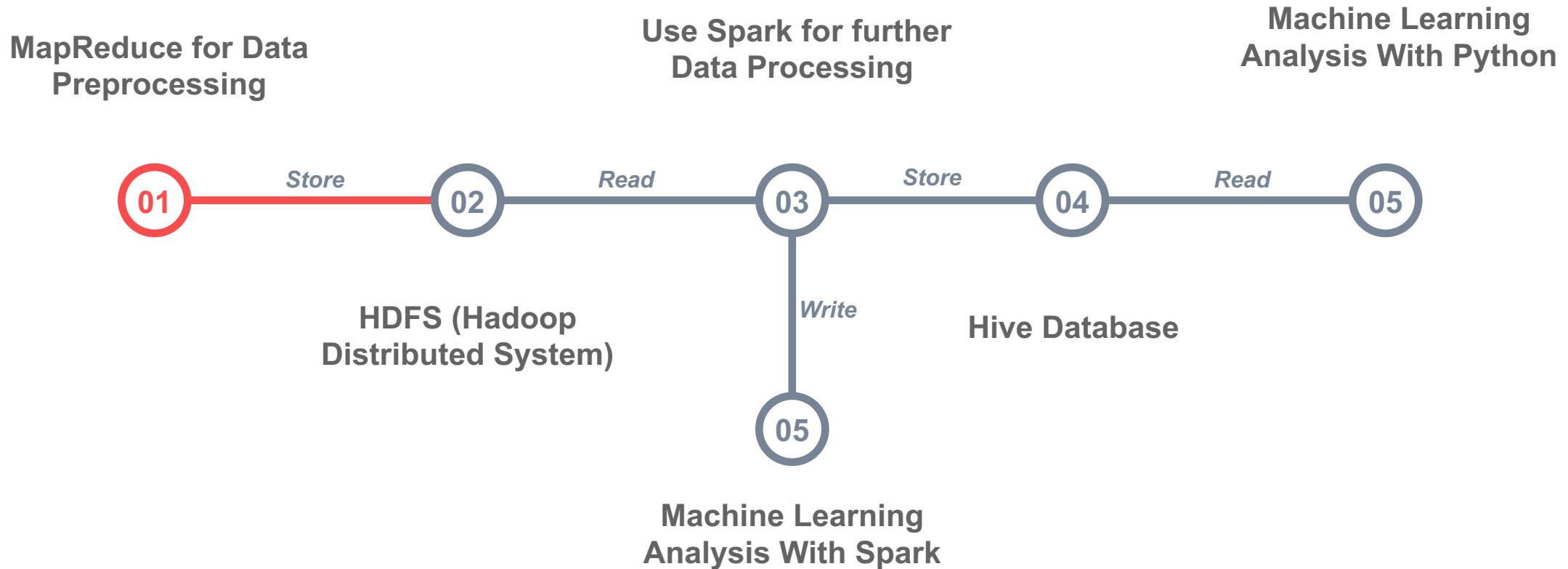
- ❑ Correlations between “Count” and other features
- ❑ Multicollinearity – “temp” & “atemp”



Rental Pattern Analysis:

- ❑ Higher demand in summer months
- ❑ Peak rental time of one day
- ❑ Rentals in weekend
- ❑ Peak user count of one day

Data Processing Stream



Data Preprocessing With MapReduce

1. Input Data Splitting:

Input data is divided into multiple blocks.

2. Mapping:

- Map tasks process data blocks.
- Mapping transforms data into intermediate key-value pairs.
- Intermediate results are stored temporarily.

3. Partitioning and Sorting:

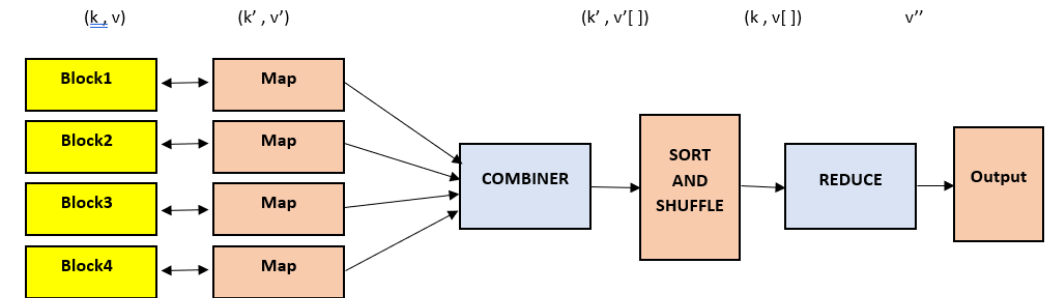
- Partitioning groups key-value pairs based on keys.
- Sorting arranges key-value pairs within partitions.

4. Reducing:

- Reduce tasks process intermediate results.
- Reduction merges key-value pairs with the same key.
- Final results are written to output.

Data Preprocessing:

- Missing Value Analysis
- Remove Outliers



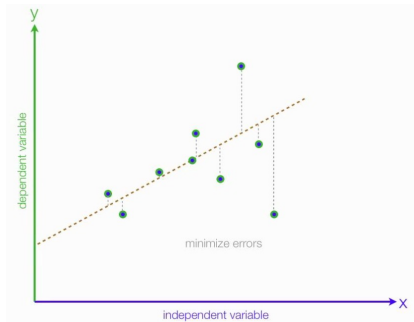
Further Data Processing with Spark and Hive



- ❑ Defined the schema to specify the structure of the data.
- ❑ Read the training and test data from HDFS using the specified schema.
- ❑ Conducted data preprocessing on the training and test data, including type conversion, date extraction, and feature selection.
- ❑ Saved the processed data into Hive tables for Machine Learning Analysis.

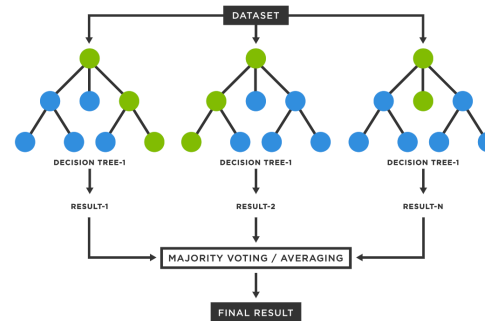
Machine Learning with Scala

Forecasting by ML Methods



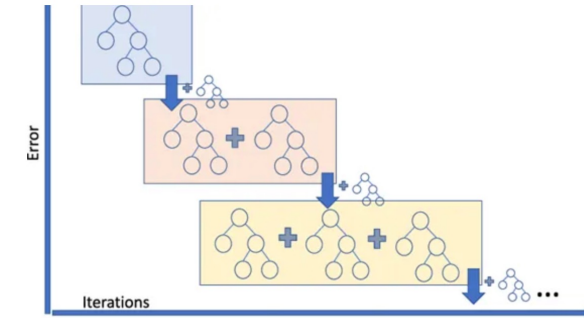
Linear Regression

- Simple and interpretable
- Low Memory Requirements
- Linear Assumption
- Bad performance on complex data



Random Forest

- Strong Performance
- Parallel Processing
- Computationally Intensive
- Long training time



Gradient Boosting

- High Predictive Power
- Parallel Processing
- Computationally Intensive
- Long training time

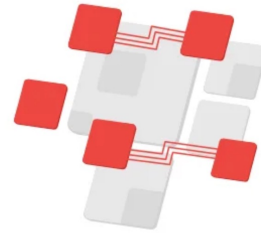
Machine Learning with Scala

Advantages of Scala



Concurrency

- Applying special structure JVM to achieve parallel training
- Significantly accelerates the training phases for large dataset.



Scalability

- Distributing the workload across multiple machines.
- Faster computing when dataset becomes larger

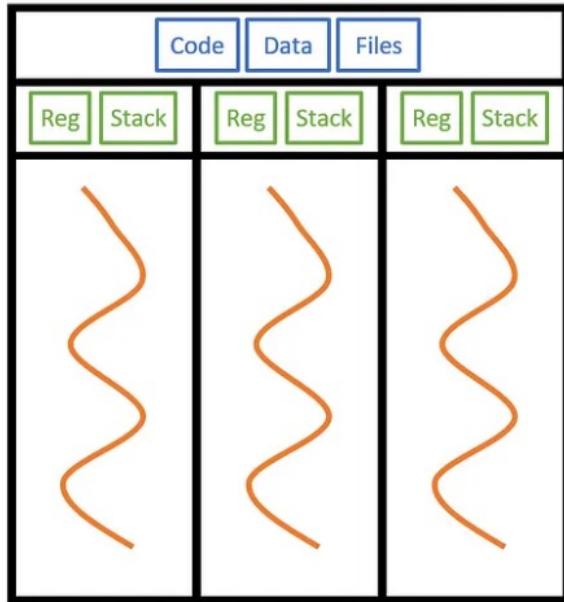


In-Memory processing

- Integrates well with Spark
- Spark could Minimize disk I/O operations, leading to faster execution

Deep Learning with Python

Multithreading



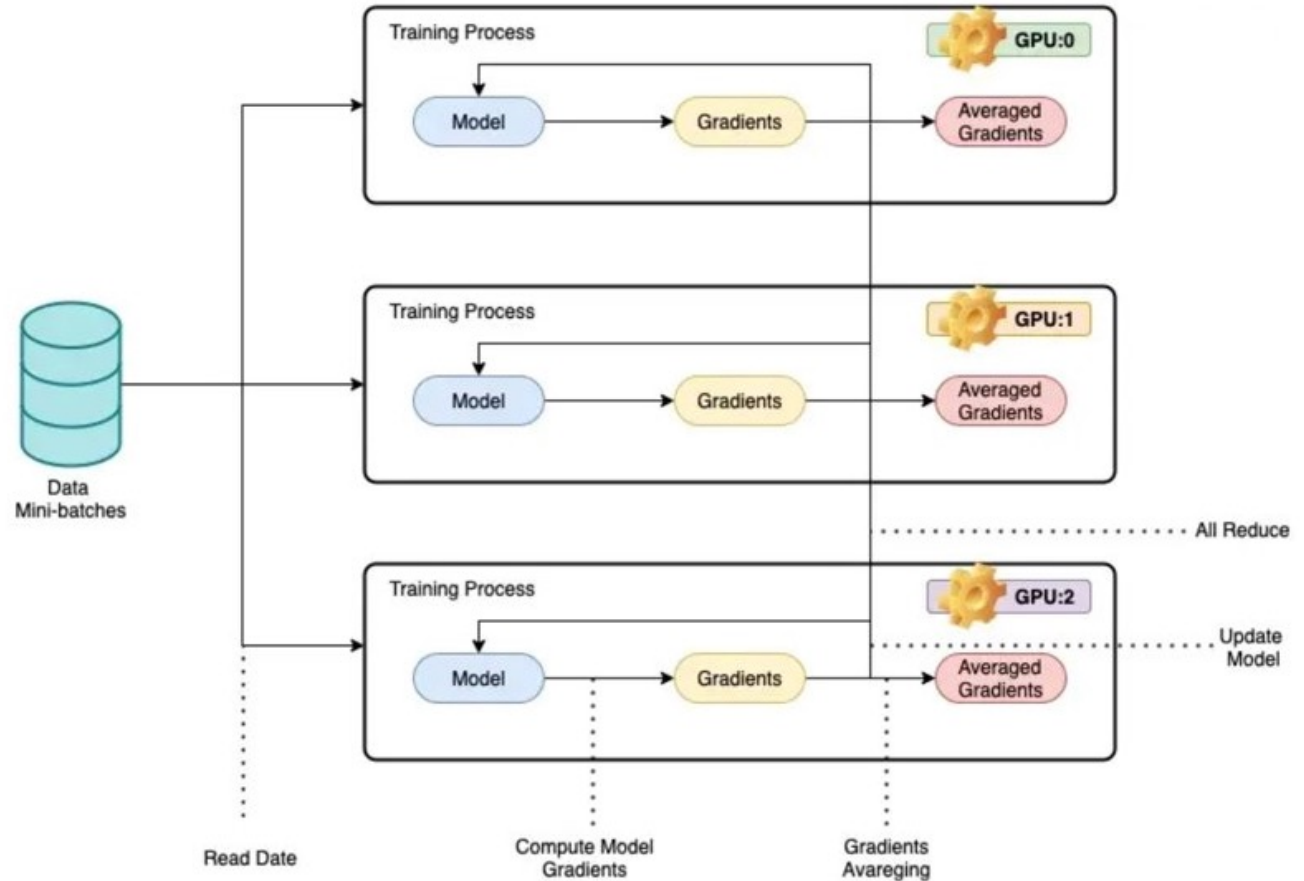
```
1 print("Time cost with parallel method:")
```

Time cost with parallel method: 0.84s

```
1 print("Time cost without parallel meth")
```

Time cost without parallel method: 2.27s

Model parallel



Much Faster!

Conclusion & Future Work

**Thank you for
watching.**

