The Impact of Bike-Sharing Ridership on Air Quality: A Scalable Data Science Framework

Nina Hua*, Victoria Suarez*, Rebecca Reilly*, Philip Trinh*, Paul Intrevado, Diane Myung-kyung Woodbridge
{nhua2,vasuarez2,rreilly3,ptrinh,pintrevado,dwoodbridge}@usfca.edu
Data Science Program
University of San Francisco

***Abstract*—This research explores the relationship between daily air quality indicator (AQI) values and the daily intensity of bike-share ridership in New York City. The authors designed and deployed a distributed data science framework on which to process and run Elastic Net, Random Forest Regression, and Gradient Boosted Regression Trees. Nine gigabytes of CitiBike ridership data, along with 1 gigabyte of air quality indicator (AQI) data were employed. All machine learning algorithms identified bike-share ridership intensity as either the most important or the second most important feature in predicting future daily AQIs. The authors also empirically demonstrated that although a distributed platform was necessary to ingest and pre-process the raw 10 gigabytes of data, the actual execution time of all three machine learning algorithms on cleaned, joined, and aggregated data was far faster on a local, commodity computer than on its distributed counterpart.**

***Index Terms*—Distributed computing, Distributed information systems, Distributed databases, Machine learning, Air pollution, Air quality, Intelligent transportation systems**

I. Introduction

Air pollution has a well-documented negative impact on the human respiratory system. A 14-to-16-year mortality follow-up study of 8,111 adults in six U.S. cities confirmed that there are statistically significant associations between air pollution and mortality, causing death from lung cancer and cardiopulmonary disease [1]. A recent study from Guttikunda also demonstrated in 2010 that 695,000 premature deaths in India were caused by continued exposure to air pollutants including particulate matter (PM) and ozone(O_3) [2].

In an effort to improve air quality, both the public and private sectors have developed regulations and recommendations: the legislation of vehicle emission standards, stricter environmental laws, energy conservation policies including encouraging limited driving, and the planting of vegetation [3], [4].

Vehicle emissions are a main cause of increased atmospheric carbon dioxide (CO_2), a major air pollutant. Various efforts to curb CO_2 emissions include fuel economy regulations, technological advances in vehicle emission efficiency, and developments in electric- and hydrogen-powered vehicles [5]. Many local governments have also redesigned and rebuilt public spaces to encourage walking, biking, and the use of public transportation.

A recent study concluded that a 5% increase in walkability and bikeability can contribute to 6.5% fewer vehicle miles traveled. This reduction in vehicle miles in turn results in emissions reductions of both nitric oxides (NOx) and volatile organic compounds (VOCs) by 5.6% and 5.5% respectively [6].

A popular way for municipalities to encourage bikeability is through the introduction of bike-share programs. This free or low-cost service enables residents to borrow/rent bicycles from a docking station and, after use, return the bicycle to a set of designated docking locations. A formalized bike-sharing system was first introduced to the U.S. in 1994 in Portland, Oregon. The number of user rides has been increasing by over 25% yearly. Currently, 25 cities in the U.S. have an established bike-sharing program [7].

Bike sharing has many redeeming qualities, including health benefits to riders [8], a reduction in automobile congestion [9], and providing an additional, highly affordable mode of transportation for those who lack access to other forms of transportation [10]. Bike-sharing programs can also contribute to improving air quality in many cities: a rider on a bicycle will generate 80% less emissions per kilometer than a passenger car [11].

To date several studies have focused on factors influencing bike share usage [12], [13], [14] and its impact on car use [15]. To our knowledge, there is no study that formalizes the relationship between bike-share usage and air quality. Joining daily air quality index (AQI) values with bike-share usage data from CitiBike—New York's bike-sharing program, and the largest bike-sharing service in the United States—the authors leverage several machine learning and distributed computing techniques to analyze 10 GB of data, and clearly demonstrate that the intensity of bike-share usage on a given day is a strong predictor of daily AQI in major urban areas. Additionally, the authors develop a robust framework in which to store and process a large volume of real-time, streaming data from rapidly expanding bike share systems. The framework is sufficiently robust to accommodate growth in air quality data, as well as Internet of Things (IoT) and wearable device data.

A traditional, non-distributed, single machine environment is unable to process the voluminous amount of data required for this type of analysis in an expedient and cost effective manner. The authors therefore develop and design a scalable computational framework for storing and managing both bike-share and air quality data, and efficiently apply machine learning algorithms, all in an economical computational en-

[*] These authors contributed equally.

vironment.

II. Background

A. Cloud computing and Amazon Web Services

Cloud computing provides remote access to data centers over the Internet, and provides information technology infrastructure and management for users to store and process data. Sharing resources among users and providing maintenance services for hardware and software, cloud computing is an economical means by which to manage infrastructure, and has become a powerful tool for individuals and organizations with large data storage and intense data processing needs. Cloud computing services must be easily scalable, offer redundancies, and ensure both data integrity and privacy [16], [17], [18].

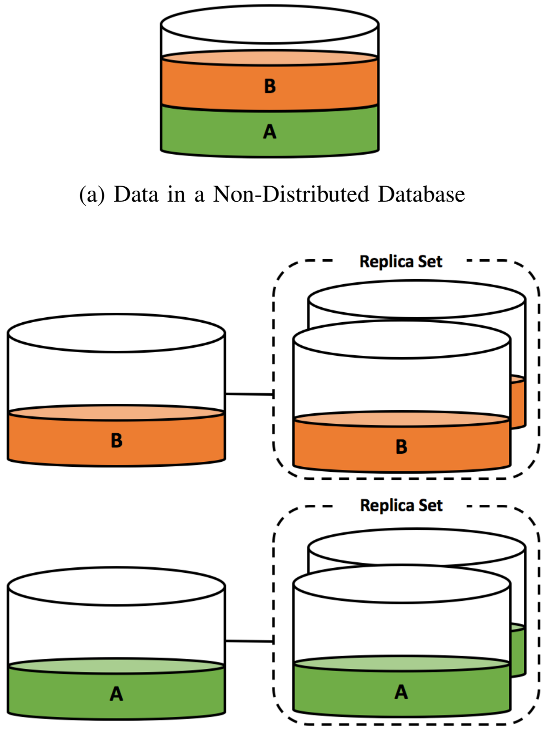
Amazon Web Services (AWS) is the largest public cloud service, offering data storage, management, computing, and analytics services, to name but a few. [19]. AWS facilitates the launching of virtual machines (VMs), and allows users to specify hardware, software, and networking requirements, including speed and quantity of CPUs/GPUs, storage size and speed, live memory, operating systems, as well as other software for web servers and databases. AWS also provides configuration options with a focus on balancing scalability, availability, data integrity, and security. Simple Storage Service (S3) and Elastic Compute Cloud (EC2) are the two most popular AWS services. S3 offers a unified storage solution, and EC2 provides remote computing on demand [20], [21]. AWS Elastic Map Reduce (EMR) was utilized for distributed computing.

B. NoSQL and MongoDB

Owing to the recent increases in available data, traditional data-management techniques and systems are ill-suited to efficiently process and store large volumes of data. Moreover, traditional relational database management systems (RDBMSs) do not support all the data types used in application programs specifically written in object-oriented languages [22]. As the volume and data types used and stored by an application evolve, users require a new database management system whose schema can similarly evolve [23]. Perhaps most importantly, a scalable database ensures both data integrity and redundancy. Unfortunately, most existing relational databases do not offer native mechanisms for data redundancy, nor can they scale the database beyond a single server [24].

An alternative to the traditional RDBMS, a NoSQL (not only SQL) database management system runs on a distributed infrastructure with multiple connected servers. NoSQL supports various data types, has an evolving schema, and is able to store data with far fewer structural restrictions than SQL. NoSQL additionally splits data into multiple shard servers in order to avoid congestion on a single server, significantly improving read and write access. In order to ensure the existence of redundancies and enhance data integrity, each shard server copies the data into its replica servers using various configurations. [25] (Figure 1).

MongoDB is the most popular NoSQL variant, storing data in a JavaScript Object Notation (JSON) format. MongoDB is also highly flexible, allowing for schemas that can differ across documents. Moreover, attributes can be created without formally defining or altering a schema. MongoDB also manages the loading and balancing of data across clusters, and routes user requests to the correct machines. MongoDB supports various data types including arrays, regular expressions, embedded documents, and JavaScript code, in addition to commonly supported data types in other RDBMSs [26].



(a) Data in a Non-Distributed Database

(b) Data Shards and Replica Sets in Distributed NoSQL Database

Fig. 1: Comparison: Distributed vs. Non-Distributed Databases

C. Distributed Computing and Apache Spark

Distributed computing facilitates the processing of large volumes of data by utilizing networked computers. Hadoop MapReduce, introduced by Google, is a programming paradigm for processing large data sets using distributed computing. Hadoop MapReduce is composed of two functions, a map and a reduce function. A map function processes a key/value pair in parallel to generate a set of intermediate key/value pairs. A reduce function merges all intermediate values from the map function associated with the same key and returns final key/value pairs to a driver [27]. This highly-effective model allows users to design programs with successive map and reduce operations, and remains in production today.

Apache Spark is a variant of MapReduce, improving upon performance for interactive algorithms by reusing a working set of data across multiple parallel operations. Spark runs up to 100 times faster than Hadoop MapReduce, utilizing in-memory computing and an advanced task-execution engine [28]. Spark uses resilient distributed datasets (RDDs), which is an abstract of a read-only collection of objects partitioned across a set of machines. RDDs can be rebuilt if data is lost or a system fails and achieves fault tolerance using RDD lineage. RDD lineage tracks RDD dependency information including its parents and operations used to create the current RDD in a directed, acyclic graph. Spark provides built-in machine learning libraries called MLlib, optimized for iterative machine learning algorithms [29].

To use Apache Spark, the authors employed AWS Elastic Map Reduce (EMR) solution. EMR is a cluster of EC2 instances that is optimized for running distributed computing frameworks including Hadoop MapReduce and Apache Spark. By default, EMR uses YARN (Yet Another Resource Negotiator) for resource management, which was introduced with Hadoop, but is also supported by Spark. EMR automatically provisions hardware resources, installs the required software, and provides an accessible monitoring dashboard.

III. System Overview

Owing to the large amount of data required for this research, the data science pipeline was designed around scalability, cloud resources, and distributed methods. Technologies were selected to build an ingestion and prediction engine leveraging data from both CitiBike and air quality index (AQI) data from the Environmental Protection Agency (EPA). The authors selected Amazon Web Services (AWS) as the primary platform to host storage, data extraction, transform and load (ETL) processes, and machine learning tasks. See Figure 2 for a visual representation of the complete data pipeline.

A. System Workflow

1) Data & Storage: CitiBike, the provider of New York City's bike-sharing program, has a fleet of 12,000 bicycles and 750 unique pick-up/drop-off docking stations across Manhattan, Brooklyn, Queens, and Jersey City. Most CitiBike docking stations are located within five-minutes (walking) from public transportation stations, providing a last-mile solution for commuters [30], [31]. The authors analyzed ridership data from 2016 to 2018, inclusive.

To analyze the relationship between bike-share ridership intensity and air quality index (AQI)—an aggregate measure of air pollution—the authors obtained and merged CitiBike ridership data with daily pollution emissions data in New York City.

The data was stored in AWS S3 to achieve high scalability, redundancy and security. As there is no limit on the amount of data nor the repository size, S3 is a reliable and cost-efficient option for storing voluminous data such as AQI and bike-share information. S3 also provides a high level of ease of interoperability with other AWS services including EC2, which was also employed in this research.

2) Data Management and Pre-Processing: The data was loaded and distributed from S3 into MongoDB, which is installed across several AWS EC2 instances. MongoDB is configured to have sharded clusters, where each cluster has one primary (master), multiple secondary (slave) nodes, a configuration server cluster with one primary and multiple secondaries, and a routing server (Figure 2). Configuration servers store the metadata for a sharded cluster, containing the state and data organization of each shard. A routing server takes a query request and determines the location to direct the request, using metadata stored in the configuration server to route read and write request to the correct shard. Lastly, data from MongoDB is loaded to the AWS Elastic Map Reduce (EMR) cluster, installed with Apache Spark.

B. Algorithms

The authors seek to formally establish the relationship between bike-share ridership intensity and air quality index (AQI). Moreover, using historical AQI, ridership data and machine learning algorithms, the authors aspire to accurately predict future daily AQI values.

To achieve the aforementioned objectives, the authors have conducted their analysis on a broad spectrum of supervised learning techniques, including Elastic Net (regularized linear regression), Random Forest Regression, and Gradient-Boosted Regression Trees.

1) Elastic Net: Elastic Net takes the traditional loss function for linear regression, and adds two additional penalties, resulting in the following loss function:

$$\sum_{i=1}^{n}(Y-\hat{Y})^2+\lambda\Big[\alpha\sum_{j=1}^{p}|\beta_j|+(1-\alpha)\sum_{j=1}^{p}\beta_j^2\Big] \qquad (1)$$

where the data contains n, observations and $p-1$ explanatory variables, Values for the scaling parameters λ and α can vary between 0 and 1 inclusive. The scaling parameter λ controls how much the penalty factors into the loss function, whereas α controls the tradeoff between the L1 ($|\beta_j|$) and L2 (β_j^2) penalties. Loosely speaking, increasing both $lambda$ and $alpha$ result in a model with fewer explanatory variables, with the most important variables being retained.

2) Random Forest Regression: A Random Forest Regression is an ensemble of decision trees, where the output is a mean prediction of the constituent trees. Each individual tree is exposed to a (potentially bootstrapped) subset of the observations and variables, and consists of a series of binary splits that look to optimize the loss function. This inherent randomness within the trees avoids overfitting, a scenario where the model is overly sensitive to training data and insufficiently sensitive to test data. The compartmentalized nature of the algorithm with independent trees is especially well-suited to a distributed computing framework.

Random Forests have become an increasingly popular technique that has the added benefit of requiring a minimal amount of architecture and hyper-parameter tuning, making them

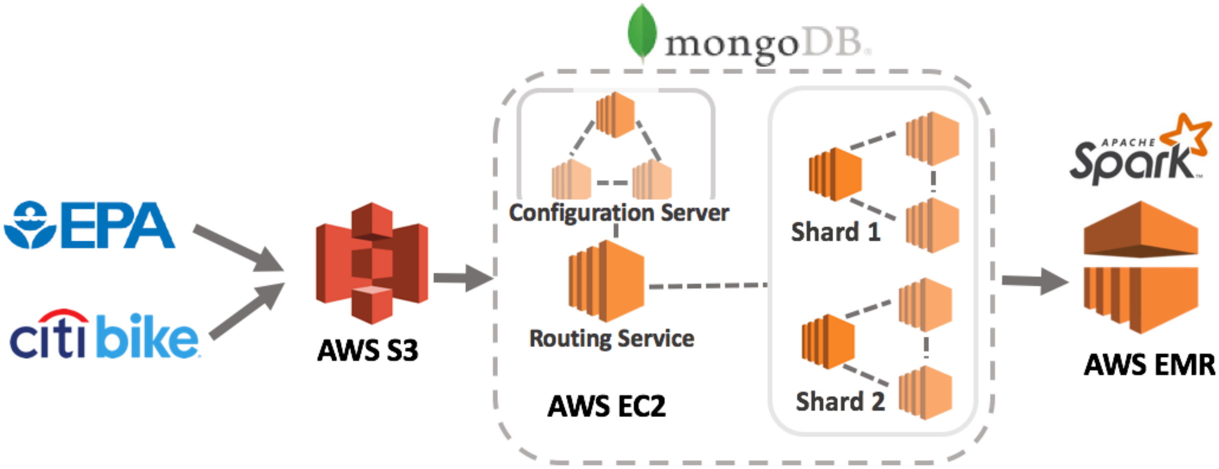


Fig. 2: System workflow

TABLE I: Air Quality Index Levels

Air Quality Index Levels of Health Concern	Numerical Value	Meaning
Good	0 to 50	Air quality is considered satisfactory, and air pollution posse little or no risk.
Moderate	51 to 100	Air quality is acceptable; however, for some pollutants there may be a moderate health concern for a very small number of people who are unusually sensitive to air pollution.
Unhealthy for Sensitive Groups	101 to 150	Members of sensitive groups may experience health effects. The general public is not likely to be affected.
Unhealthy	151 to 200	Everyone may begin to experience health effects; members of sensitive groups may experience more serious health effects.
Very Unhealthy	201 to 300	Health alter : everyone may experience more serious health effects.
Hazardous	301 to 500	Health warnings of emergency conditions. The entire population is more likely to be affected.

relatively easy to train on a data set. Tree depth is an important hyper-parameter of a random forest algorithm, which we tune and evaluate in this research. Although increasing tree depth improves a model's predictive accuracy, it requires a longer training time. Moreover, it may also cause overfitting. Spark's MLlib provides a method to specify the maximal depth of any individual tree within the forest, allowing for the analysis of the relationship between tree depth and the size of data used to train the random forest.

Another benefit of using Random Forests is ease of interpretability. We can easily quantify the feature importance in any model by observing the effect that randomizing each feature has on model prediction quality.

3) Gradient Boosted Regression Trees: Gradient Boosted Regression Trees are similar to their Random Forest Regression counterparts. They benefit from a collection of decision trees, subsequently making a prediction based on the weighted scoring from each of those trees. The primary difference in Gradient Boosted Regression Trees is that the first tree is used to make a prediction, and, once evaluated, an additional tree is added such that it minimizes error, i.e., minimizes the loss of the first tree. Trees are added, one at a time, each minimizing the loss of the preceding tree, until a robust model is developed. As trees are added sequentially and not in parallel—owing to the dependency of earlier predictions—this reduces the performance benefit of distributed computing.

4) Baseline: A naive prediction using the historical mean is used to establish a baseline for comparison.

IV. Experiment Output

A. Data & Computing Specifications

In an effort to predict AQI based on bike-sharing intensity, New York City bike-share ridership data for all CitiBike rides was downloaded from 2016 to 2018 inclusive, with 46,779,707 observations and 10 variables, totalling 9 gigabytes. This data included the following trip information: trip duration, start time, stop time, date, station ID and longitude/latitude of the beginning and ending docking stations, user type (24-hour pass, 3-day pass, or annual member), rider gender, and year of birth. Bike-ridership data was subsequently aggregated at the daily level, computing the total rides in a given day for a given location. Figure 3 depicts the aggregate ride-share data over all three years.

Daily pollution emissions data in New York City was also obtained in an effort to reconstruct daily air quality index (AQI) values, an aggregate measure of air pollution that is computed in two stages. Firstly, a given monitoring station reports levels of air pollution for the following pollutants: ground-level ozone (O_3), particulates (PM), sulfur dioxide (SO_2), carbon monoxide (CO), lead (Pb) and nitrogen dioxide (NO_2) [32]. With the levels of each of the aforementioned pollutants reported, the second step selects the highest reported level of an individual pollutant as the daily AQI. To ease interpretation for the public, AQI levels are mapped to six discrete *AQI Levels of Health Concern* (see Table I).

Air Quality Index data is not directly available from the United States Environmental Protection Agency (EPA), however levels of individual pollutants are readily available [33]. Therefore, to reconstruct daily AQI levels, the authors downloaded the daily pollution levels for CO, Pb, NO_2, O_3, PM_{10}, $PM_{2.5}$, SO_2 across all proximal New York City pollution monitoring stations. The AQI was subsequently computed by selecting the maximal pollution level across all of the aforementioned pollutants. This resulted in a 1 gigabyte data set that includes: date, monitoring site ID, longitude/latitude, and AQI. Figure 4 depicts the AQI levels for New York City from 2016–2018. Yearly correlation values between mean monthly AQI and total bike-share ridership were -0.788, -0.479 and -0.333 for 2016, 2017 and 2018, respectively.

The data was subsequently joined and merged into a final data set that contained 1,095 rows (365 days * 3 years) and five features, outlined in Table II. A feature vector of a given day's ride count, the previous day's AQI, and the seasonal indicator were calculated, aggregated, and created to be used as inputs to various machine learning algorithms.

TABLE II: Machine Learning Features

Feature	Definition
1	Date (mm/dd/yyyy)
2	Total number of bike-sharing rides on that day
3	Previous day's AQI
4	Seasonal Indicator $\in$ {Fall, Winter, Spring, Summer}
5	Current day's AQI

Although the final data used as input to the machine learning models is of a trivial size, the manipulation of the original 10 gigabytes of data (9 GB CitiBike + 1 GB AQI) required a distributed computing framework. The specification of EC2 and EMR clusters of MongoDB and Apache Spark are given in Table III and Table IV, respectively.

TABLE III: AWS EC2 MongoDB Specifications

Role	EC2 Type	CPUs	Memory	Disk
Shard (2 shards with a primary and two secondaries)	t3.large	2	8 GB	16 GB
Configuration Server (one primary and two secondaries)	t2.small	1	2 GB	8 GB
Routing Server	t3.medium	2	4 GB	8 GB

B. Results

The Root Mean Squared Error (RMSE) is a typical metric by which predictions are evaluated, and will be used as the metric of comparison across the machine learning algorithms herein. It is computed as the root of the squared sum of distances of predicted and true values. For the i^{th} record, the difference between predicted and actual AQI is calculated. Then the differences over n total observations are squared, averaged, and the square root taken (see Equation 2).

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(Y_i - \hat{Y_i}\right)^2} \qquad (2)$$

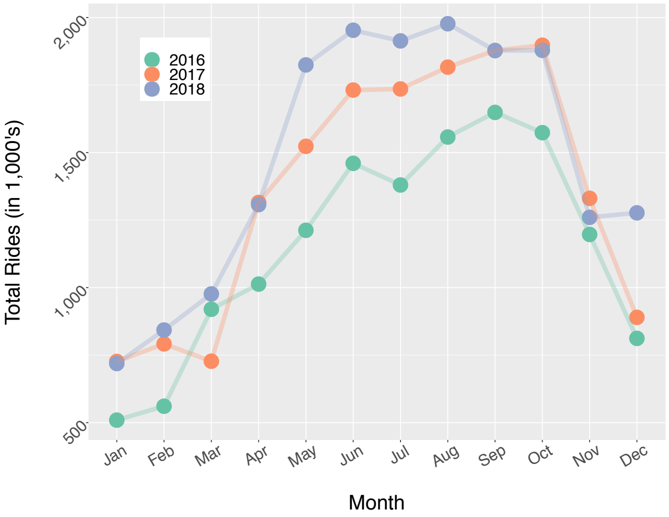


Fig. 3: CitiBike Rides Per Month, 2016–2018

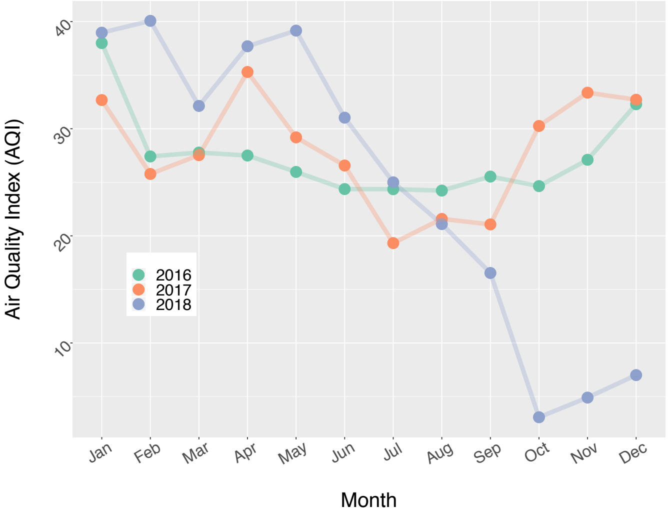


Fig. 4: NYC AQI Per Month, 2016–2018

TABLE IV: AWS EMR Cluster Specifications for Apache Spark

Role	EC2 Type	CPUs	Memory
Master	r5.2xlarge	8	64 GB
Slave (5 Nodes)	r5.2xlarge	8	64 GB

The Elastic Net, Random Forest Regression and Gradient Boosted Regression Trees were run under several parameter settings, outlined in Table V. Models were compared using using k-fold cross validation—a technique employed to compare and select models —with $k = 10$ folds, using a random 80/20 train/validation split for the 2016–2017 data. The models were then tested on 2018 data. RMSE was reported for both the validation and test sets. Owing to the small size of the final input data, all algorithms were also run both on a distributed framework, using Spark's MLlib, as well as on a local machine with Python's Scikit-Learn. RMSE results are reported in Figure 5.

TABLE V: Modeling Parameters

ML Algorithm	Parameters
Elastic Net	$\lambda = 1.0$, $\alpha \in \{0, 0.1, \ldots, 0.9, 1.0\}$
Random Forest Regressor	Max Depth $\in \{2, 9, 16, 23\}$ Number of Trees $\in \{2, 10, 18, 26, 34, 42\}$
Gradient Boosted Regression Trees	Max Depth $\in \{2, 6, 10\}$ Max Iterations $\in \{2, 6, 10\}$

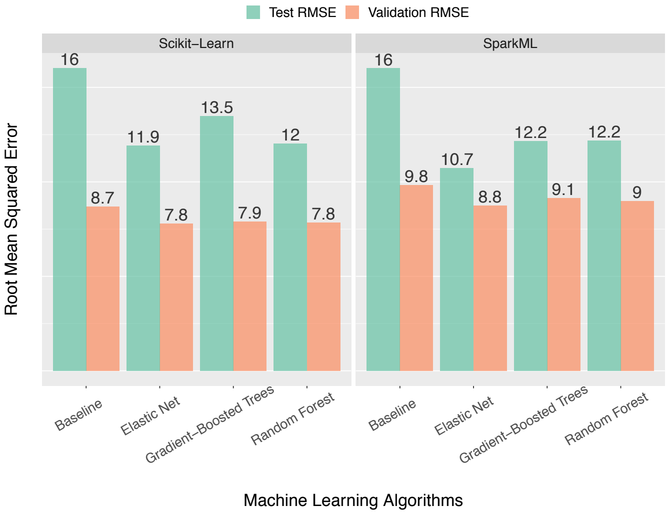


Fig. 5: Validation and Test RMSE for all Machine Learning Models in Spark MLlib and Scikit-Learn

The results from Figure 5 confirm that all three machine learning algorithms preform significantly better than the baseline. Moreover, the Elastic Net generates the lowest test and validation RMSE on both Spark MLlib and Scikit-Learn. Although the resulting RMSEs generated by Spark MLlib and Scikit-Learn aren't an exact match for each machine learning algorithm—owing to the randomness of the train/validation/test procedure—the consistency in the implementation algorithms across both platforms is reassuring.

All algorithms confirm that bike-share *ride count* is one of the top two features when predicting the following day's AQI.

Figures 6 and 7 report the execution time for Random Forest Regression and Gradient Boosted Regression Trees across various parameter settings on Spark MLlib. As expected, computational time increases non-linearly as maximum depth increases.

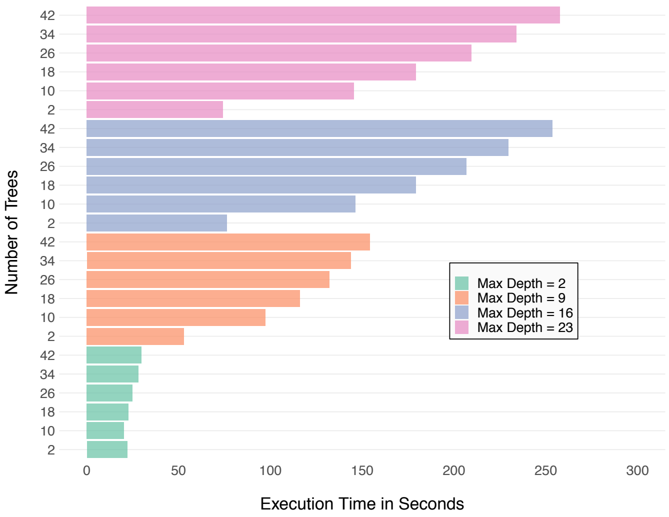


Fig. 6: Random Forest Regression Execution Times on Spark MLlib

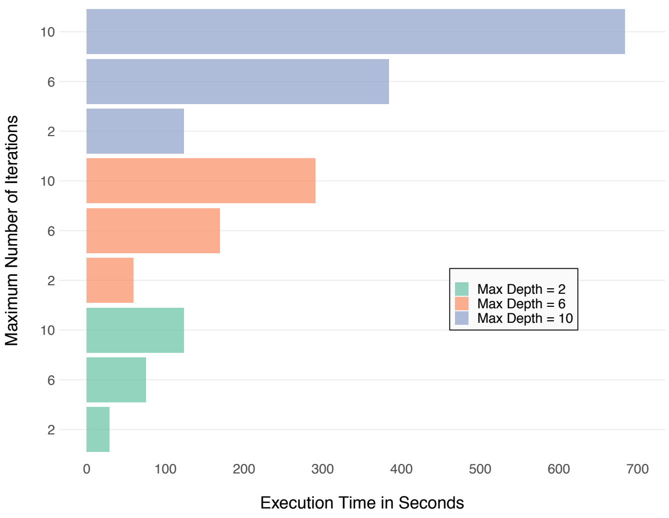


Fig. 7: Gradient Boosted Regression Tree Execution Times on Spark MLlib

Although it was necessary to leverage a distributed computing framework to merge and manipulate all 10 gigabytes of data for this analysis, using that same distributed platform to execute the machine learning algorithms resulted in diminishing returns to scale, even after additional pre-processing/data munging steps intended to speed up the Spark MLlib execution times. The raw data, once cleaned, joined, and aggregated, was orders of magnitude smaller than the raw data (KB versus 10 GB).

Using Spark MLlib to execute the machine learning algorithms on such a small data set was akin to killing a fly with a cannon, deploying a far too complex solution to solve a relatively simple problem. Figure 8 depicts the excessive computational time required by Spark MLlib to run all three machine learning algorithms in comparison to running those same algorithms on a local machine using Scikit-Learn. These significant discrepancies in execution time across platforms are attributed to two main causes.

Firstly, Spark MLlib excels for algorithms whose performance improves when distributed across a cluster. Algorithms that tend to work poorly within the distributed framework rely on boosting techniques, i.e., iteratively combining weak learners to form a single, stronger learner. This process requires a large amount of data-shuffling between individual nodes in a cluster, which is detrimental to performance.

Secondly, the massive overhead required to run a machine learning algorithm on Spark MLlib, on such a relatively small, aggregate data set is also a likely contributor to the excessive processing times. Comparatively, running those same algorithms on a local machine with Scikit-Learn generates near-instant results with no algorithm taking more than 10 seconds to complete.

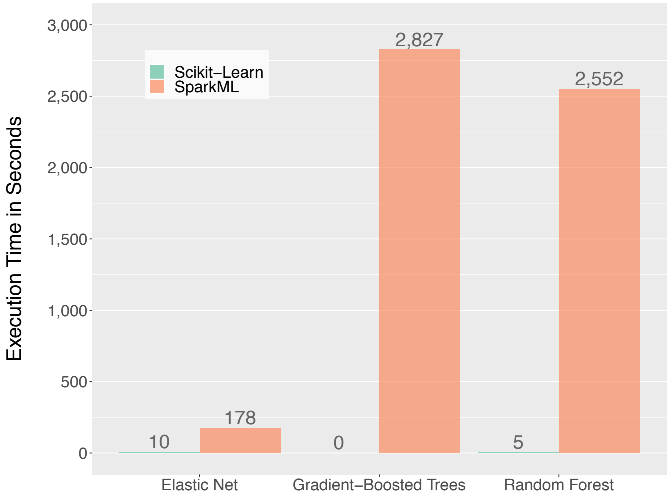


Fig. 8: Algorithm Execution Time by Platform

V. Conclusion

A variety of ride-sharing networks are popping up in cities around the U.S.: from cars to bikes to scooters, the public can easily access temporary transportation for relatively small sums. As ride-sharing systems grow and evolve, it behooves us to measure the impact these various systems have on society, so that municipal and/or state governments can enact policies that restrict or encourage certain behavior.

This research establishes an inverse relationship between daily air quality indicator (AQI) values and the daily intensity of bike-share ridership. The authors designed and deployed a distributed data science framework on which to process and run machine learning algorithms. 10 gigabytes of CitiBike ridership data, joined with air quality indicator (AQI) data for New York City, were input into three machine learning algorithms on distributed systems with varying characteristics. All machine learning algorithms tested identified bike-share ridership intensity as either the most important or the second most important feature in predicting future daily AQIs.

The authors also empirically demonstrated that although a distributed platform was necessary to ingest and pre-process the raw 10 gigabytes of data, the actual execution time of all three machine learning algorithms on the aggregated data was far faster on a local, commodity computer than on its distributed counterpart. This result is attributed to the high computational overhead costs associated with operating a distributed data science framework, relative to the small size of the aggregate data necessary to run the algorithms.

Future research should include using additional data sources that may impact pollution areas in major urban environments to better predict future AQIs.

References

[1] D. W. Dockery, C. A. Pope, X. Xu, J. D. Spengler, J. H. Ware, M. E. Fay, B. G. Ferris Jr, and F. E. Speizer, "An association between air pollution and mortality in six us cities," *New England journal of medicine*, vol. 329, no. 24, pp. 1753–1759, 1993.

[2] S. K. Guttikunda, R. Goel, and P. Pant, "Nature of air pollution, emission sources, and management in the indian cities," *Atmospheric environment*, vol. 95, pp. 501–510, 2014.

[3] H. Akbari, M. Pomerantz, and H. Taha, "Cool surfaces and shade trees to reduce energy use and improve air quality in urban areas," *Solar energy*, vol. 70, no. 3, pp. 295–310, 2001.

[4] New Hampshire Department of Environmental Services. (2019) What can i do to help reduce air pollution? New Hampshire Department of Environmental Services. [Online]. Available: https://www.des.nh.gov/organization/divisions/air/tsb/ams/aqmdp/share.htm

[5] P. Poudenx, "The effect of transportation policies on energy consumption and greenhouse gas emission from urban passenger transportation," *Transportation Research Part A: Policy and Practice*, vol. 42, no. 6, pp. 901–909, 2008.

[6] L. D. Frank, J. F. Sallis, T. L. Conway, J. E. Chapman, B. E. Saelens, and W. Bachman, "Many pathways from land use to health: associations between neighborhood walkability and active transportation, body mass index, and air quality," *Journal of the American planning Association*, vol. 72, no. 1, pp. 75–87, 2006.

[7] National Association of City Transportation Officials. (2018) Bike share in the u.s.: 2017. National Association of City Transportation Officials. [Online]. Available: https://nacto.org/bike-share-statistics-2017/

[8] D. Rojas-Rueda, A. De Nazelle, O. Teixidó, and M. Nieuwenhuijsen, "Replacing car trips by increasing bike and public transport in the greater barcelona metropolitan area: a health impact assessment study," *Environment international*, vol. 49, pp. 100–109, 2012.

[9] M. Wang and X. Zhou, "Bike-sharing systems and congestion: Evidence from us cities," *Journal of transport geography*, vol. 65, pp. 147–154, 2017.

[10] L.-Y. Qiu and L.-Y. He, "Bike sharing and the economy, the environment, and health-related externalities," *Sustainability*, vol. 10, no. 4, p. 1145, 2018.

[11] B. Blondel, C. Mispelon, and J. Ferguson, "Cycle more often 2 cool down the planet," *Quantifying CO2 savings of cycling. European Cyclistss Federation ECF, Brussel*, 2011.

[12] E. Fishman, S. Washington, N. Haworth, and A. Watson, "Factors influencing bike share membership: an analysis of melbourne and brisbane," *Transportation research part A: policy and practice*, vol. 71, pp. 17–30, 2015.

[13] K. Gebhart and R. B. Noland, "The impact of weather conditions on bikeshare trips in washington, dc," *Transportation*, vol. 41, no. 6, pp. 1205–1225, 2014.

[14] X. Wang, G. Lindsey, J. E. Schoner, and A. Harrison, "Modeling bike share station activity: Effects of nearby businesses and jobs on trips to and from stations," *Journal of Urban Planning and Development*, vol. 142, no. 1, p. 04015001, 2015.

[15] E. Fishman, S. Washington, and N. Haworth, "Bike shares impact on car use: Evidence from the united states, great britain, and australia," *Transportation Research Part D: Transport and Environment*, vol. 31, pp. 13–20, 2014.

[16] B. Furht and A. Escalante, *Handbook of cloud computing*. Springer, 2010, vol. 3.

[17] Z. Chaczko, V. Mahadevan, S. Aslanzadeh, and C. Mcdermid, “Availability and load balancing in cloud computing,” in *International Conference on Computer and Software Modeling, Singapore*, vol. 14, 2011.

[18] D. Zissis and D. Lekkas, “Addressing cloud computing security issues,” *Future Generation computer systems*, vol. 28, no. 3, pp. 583–592, 2012.

[19] Amazon Web Services. (2019) Amazon web services (aws). [Online]. Available: https://aws.amazon.com/

[20] Amazon Web Services . (2019) Amazon s3. [Online]. Available: https://aws.amazon.com/s3/

[21] Amazon Web Services. (2019) Amazon ec2. [Online]. Available: https://aws.amazon.com/ec2/

[22] S. W. Ambler, “Mapping objects to relational databases,” *On the World Wide Web: http://www. AmbySoft. com*, 2000.

[23] C. A. Curino, L. Tanca, H. J. Moon, and C. Zaniolo, “Schema evolution in wikipedia: toward a web information system benchmark,” in *In International Conference on Enterprise Information Systems (ICEIS*. Citeseer, 2008.

[24] mongoDB. (2018) Rdbms to mongodb migration guide. mongoDB. [Online]. Available: https://webassets.mongodb.com/_com_assets/collateral/RDBMStoMongoDBMigration.pdf

[25] K. Chodorow, *MongoDB: The Definitive Guide: Powerful and Scalable Data Storage*. ” O’Reilly Media, Inc.”, 2013.

[26] MongoDB. (2019) Mongodb for giant ideas. [Online]. Available: https://www.mongodb.com/

[27] J. Dean and S. Ghemawat, “Mapreduce: simplified data processing on large clusters,” *Communications of the ACM*, vol. 51, no. 1, pp. 107–113, 2008.

[28] L. Gu and H. Li, “Memory or time: Performance evaluation for iterative operation on hadoop and spark,” in *High Performance Computing and Communications & 2013 IEEE International Conference on Embedded and Ubiquitous Computing (HPCC_EUC), 2013 IEEE 10th International Conference on*. IEEE, 2013, pp. 721–727.

[29] M. Zaharia, M. Chowdhury, M. J. Franklin, S. Shenker, and I. Stoica, “Spark: Cluster computing with working sets.” *HotCloud*, vol. 10, no. 10-10, p. 95, 2010.

[30] CitiBike. (2019) Citi bike: Nycs official bike sharing system — citi bike nyc. CitiBike. [Online]. Available: https://www.citibikenyc.com/

[31] S. M. Kaufman, L. Gordon-Koven, N. Levenson, and M. L. Moss, “Citi bike: The first two years,” 2015.

[32] US EPA. (2011) Air quality index(aqi) a guide to air quality and your health. [Online]. Available: https://www.airnow.gov/index.cfm?action=aqibasics.aqi

[33] United States Environmental Protection Agency. (2019) Download daily data — us epa. United States Environmental Protection Agency. [Online]. Available: https://www.epa.gov/outdoor-air-quality-data/download-daily-data