

Introduction to Data Science

Data Science Essentials



Meet your instructors!



Michael Holloway is a mathematician turned data scientist. Michael holds a PhD in Mathematics from the University of Tennessee, Knoxville and a Masters from Tennessee Technological University. Michael loves the challenge and excitement of keeping up with the ever-evolving landscape of data science. Prior to joining Nashville Software School in April 2019, Michael served as an Assistant Professor in the Department of Mathematics at Saint Augustine's University.



Mahesh Rao is a neuroscientist turned data scientist. Mahesh received his PhD in Biological Sciences from Vanderbilt University before starting as a data scientist at Amira Learning, where he worked to develop their data science pipeline and machine learning models. He's a lifelong learner who enjoys taking an interdisciplinary approach to new problems. Mahesh couldn't stay away from NSS and is excited to be a part of the growing data community in Nashville.



- **Your name**
- **The place you call home**
- **Something people are usually surprised to discover about you**



Classroom rules

- Ask lots of questions
- Help each other; learn from each other
- Coding tasks are a guide
 - You **don't** have to get them **all** done
 - You **can** form your own ideas and do your own exploration beyond what has been suggested



Class format

- **Review of previous coding tasks**
- **Concepts/Code Lecture**
- **Coding tasks**
- **Interactive with instruction team!**



Goals for the class

- **Get hands-on experience of what it might be like to work as a data scientist**
- **Get an idea of whether or not this might be a good fit for a career**
- **Make discoveries and have fun**



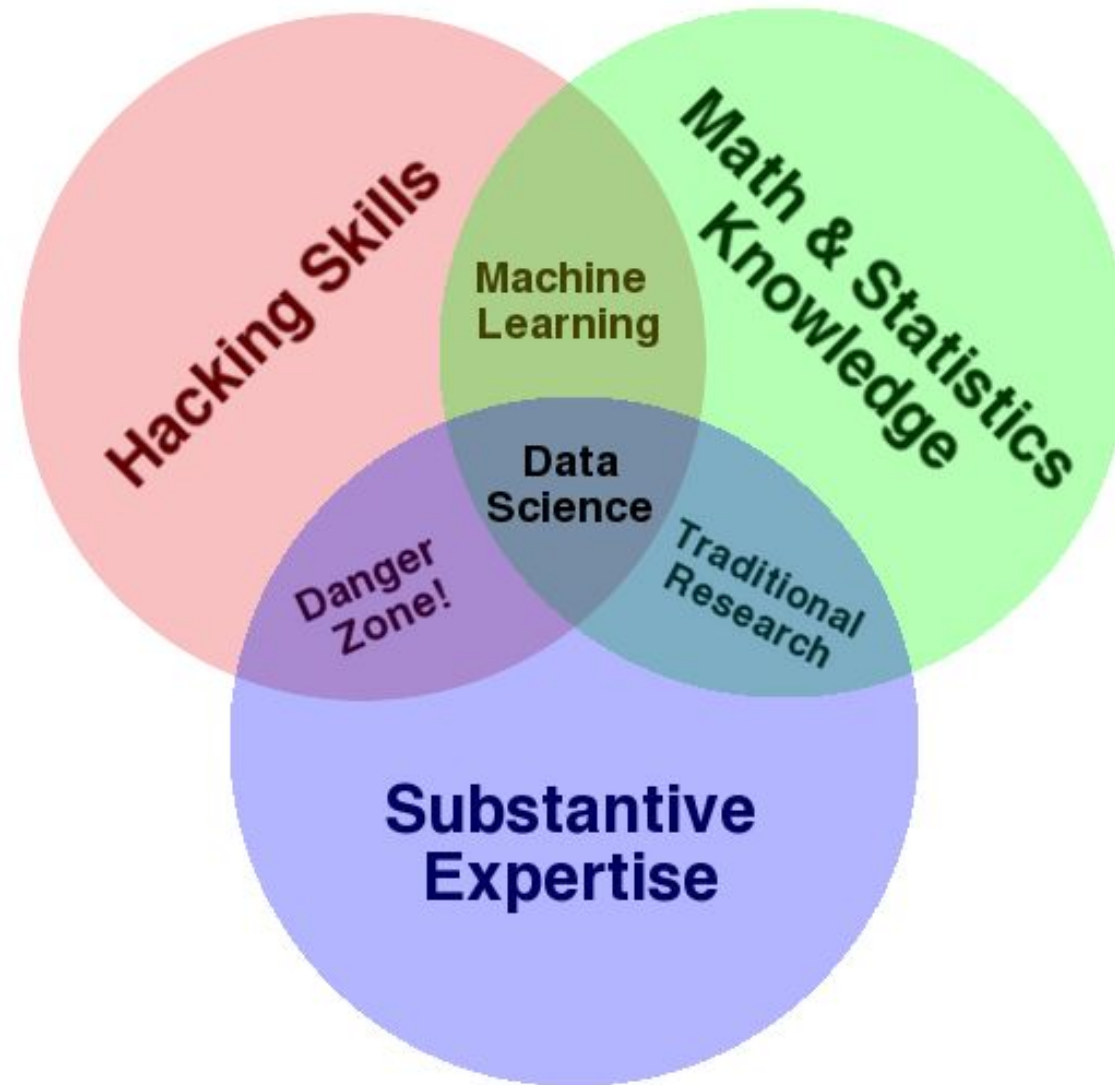
Goals for today

- **Define Data Science and the Data Science Process**
- **DM us your github account name on Slack**
- **Learn a little *pandas***
- **Understand the project questions**
- **Work in teams to complete coding tasks**



What is Data Science?





- **Data science produces insights**
- **Machine learning produces predictions**
- **Artificial intelligence produces actions**

VARIANCE EXPLAINED



David Robinson

*Chief Data Scientist at
DataCamp, works in R and
Python.*

Data science produces insights

Data science is distinguished from the other two fields because its goal is an especially human one: to gain insight and understanding. Jeff Leek has an excellent definition of the types of insights that data science can achieve, including descriptive (“the average client has a 70% chance of renewing”) exploratory (“different salespeople have different rates of renewal”) and causal (“a randomized experiment shows that customers assigned to Alice are more likely to renew than those assigned to Bob”).

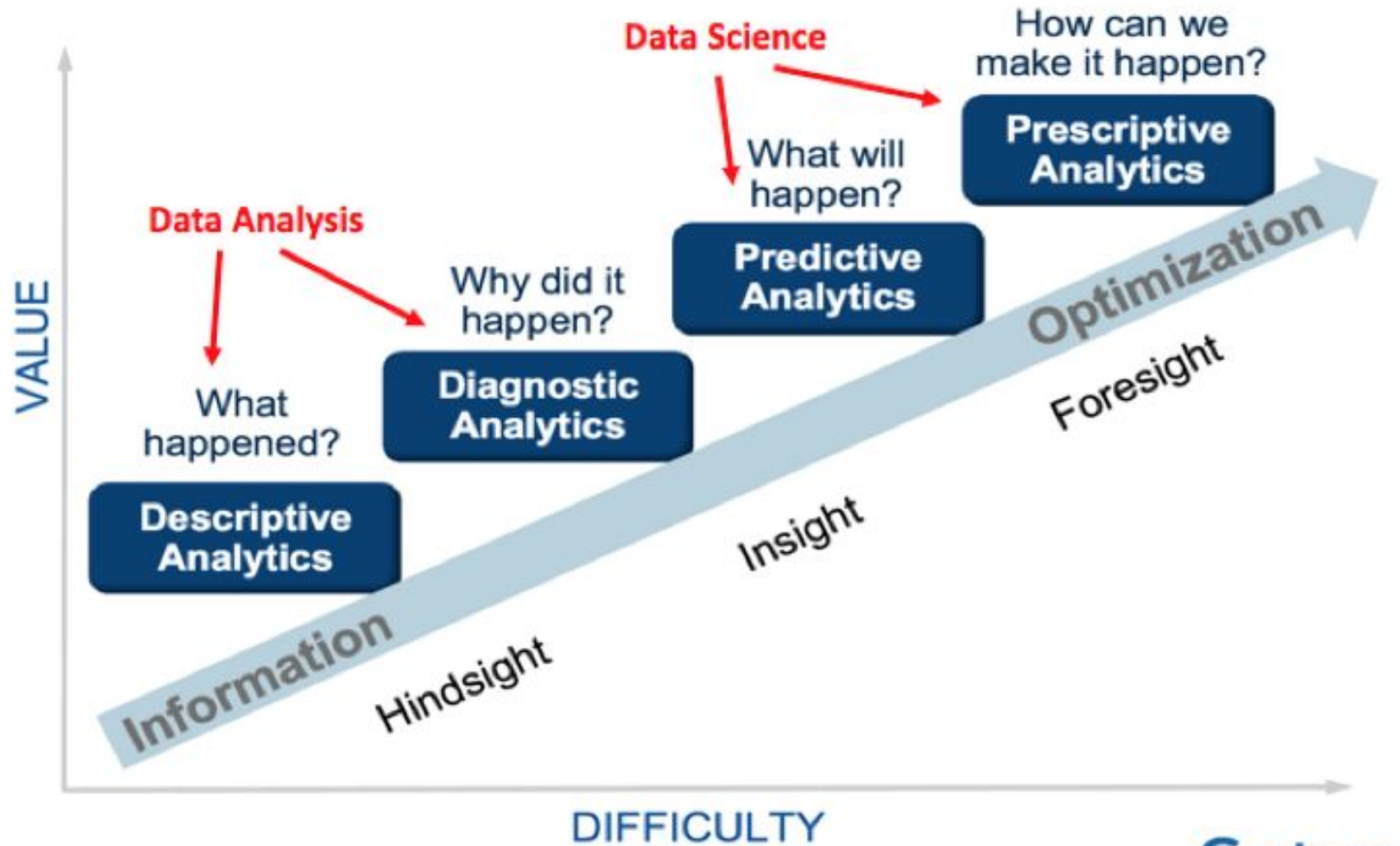
Again, not everything that produces insights qualifies as data science (the classic definition of data science is that it involves a combination of statistics, software engineering, and domain expertise). But we can use this definition to distinguish it from ML and AI. The main distinction is that in data science there’s always a human in the loop: someone is understanding the insight, seeing the figure, or benefitting from the conclusion. It would make no sense to say “Our chess-playing algorithm uses data science to choose its next move,” or “Google Maps uses data science to recommend driving directions”.

This definition of data science thus emphasizes:

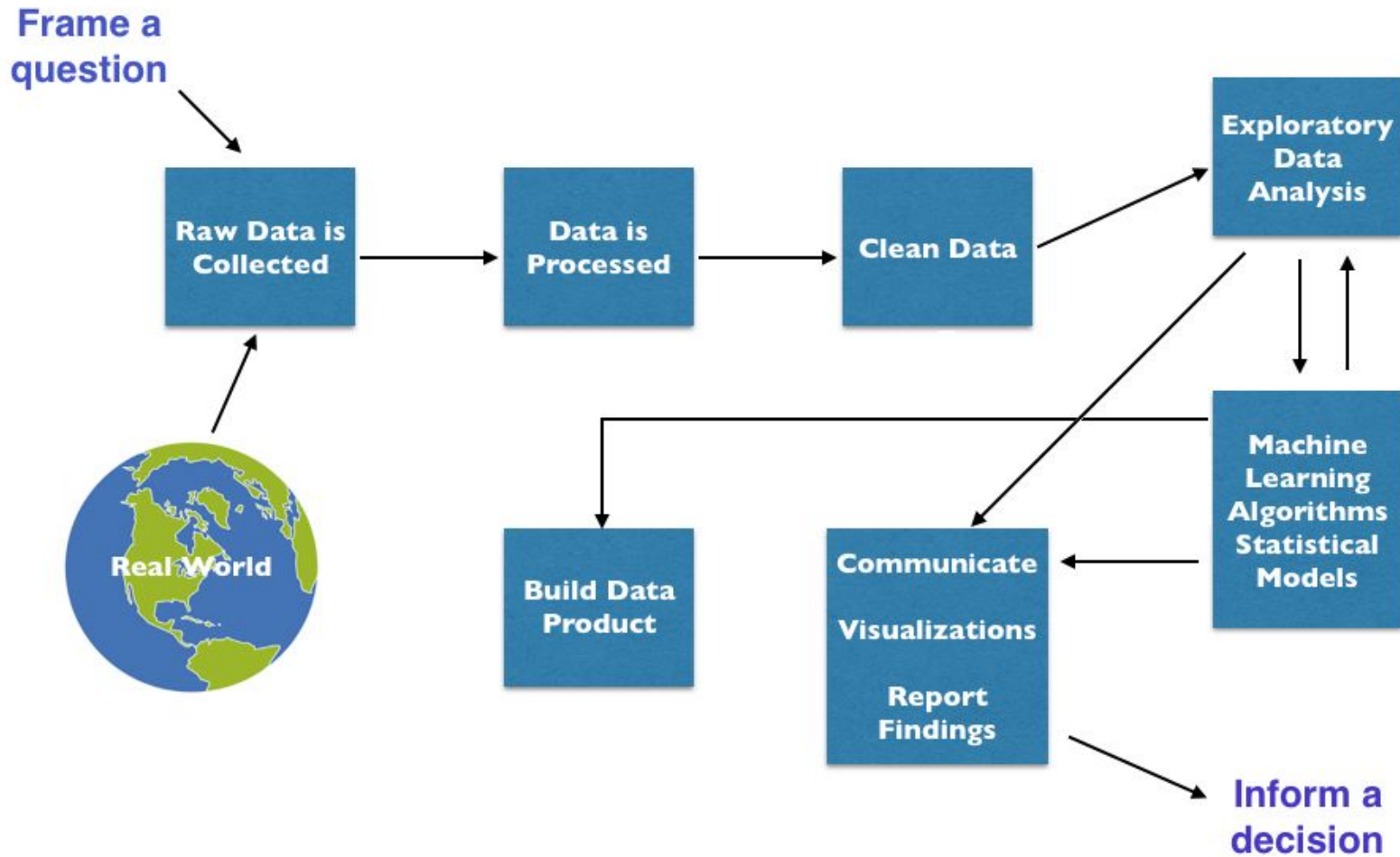
- Statistical inference
- Data visualization
- Experiment design
- Domain knowledge
- Communication

Data scientists might use simple tools: they could report percentages and make line graphs based on SQL queries. They could also use very complex methods: they might work with distributed data stores to analyze trillions of records, develop cutting-edge statistical techniques, and build interactive visualizations. Whatever they use, the goal is to gain a better understanding of their data.

Gartner Analytic Ascendancy Model



Data Science Process



We will work a lot with Python's *pandas* library, which provides methods for working with **DataFrames** and **Series**.

A **DataFrame** is a two-dimensional (tabular) data structure.

A **Series** is a one-dimensional data structure – could be a row or a column of data, but usually when we work with a series it is a column of data.

The diagram illustrates a pandas DataFrame with the following structure:

- Index:** A red arrow points to the index values (0 to 6) on the left.
- Columns:** Blue arrows point to the column headers: Name, Team, Number, Position, and Age.
- Rows:** Orange arrows point to the rows of the table.
- Data:** A purple box highlights a specific cell (Jonas Jerebko, 8.0) and its corresponding row and column.

	Name	Team	Number	Position	Age
0	Avery Bradley	Boston Celtics	0.0	PG	25.0
1	John Holland	Boston Celtics	30.0	SG	27.0
2	Jonas Jerebko	Boston Celtics	8.0	PF	29.0
3	Jordan Mickey	Boston Celtics	NaN	PF	21.0
4	Terry Rozier	Boston Celtics	12.0	PG	22.0
5	Jared Sullinger	Boston Celtics	7.0	C	NaN
6	Evan Turner	Boston Celtics	11.0	SG	27.0

GG

<https://www.geeksforgeeks.org/python-pandas-dataframe/>

pandas – <https://pandas.pydata.org/pandas-docs/stable/api.html>

Importing Data

- **pd.read_csv()** – read a comma delimited file; good practice is to look at the raw file in a text editor (like Visual Studio Code, not Excel); additional arguments may be needed to handle extra rows at the top and extra data (footnotes) at the bottom.

Inspecting

- **df.info()** – method to get information about the DataFrame
- **df.dtypes** – datatypes attribute for the Data Frame
- **df.head()** – looks at the top of the DataFrame; 5 rows by default
- **df.tail()** - looks at the bottom of the DataFrame; 5 rows by default
- **df.shape** – returns a tuple with the number of rows and number of columns



pandas – <https://pandas.pydata.org/pandas-docs/stable/api.html>

Modifying

- **df.columns** – column labels attribute
- **df.rename()** – rename values (can pass in a dictionary with existing columns as the key and new ones as the values)
- **df.drop()** – drop the specified labels (either rows or columns) from the DataFrame

Summarizing

- **.unique()** – returns the unique values in a column
- **.nunique()** - returns the *number* of unique elements in a column
- **.value_counts()** - returns the unique elements in a column and the number of appearances of each

Slicing/Filtering

- **df.loc[]** – pass in row name and column name to access data at that location
- **df[[]]** - creates a slice (subset) of the DataFrame including just the columns passed in

