STEVENS INSTITUTE of TECHNOLOGY

School of Business

FE 582

Lecture 1: Introduction to Financial Data Science

Dragos Bozdog

For academic use only.

Data Science

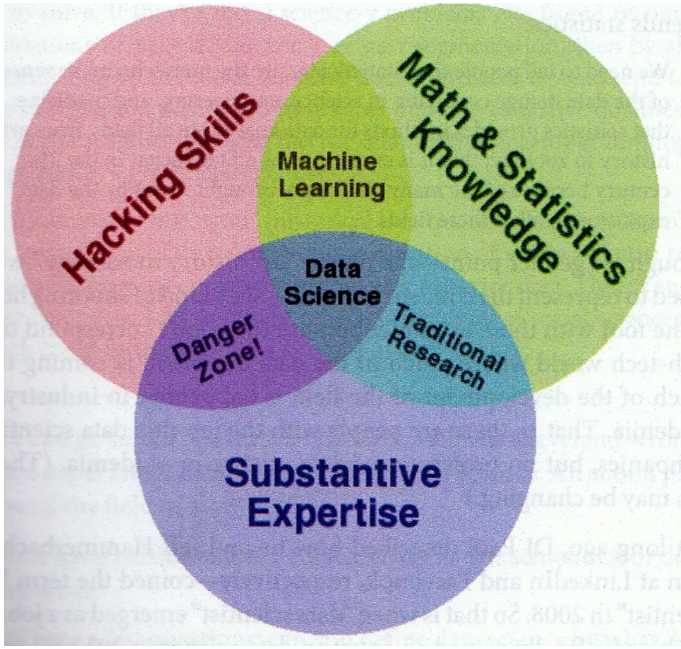
Refers to the collection of definitions, rules, methods, and analysis concerned with the collection, preparation, protection, management, analysis and synthesis of data.

Based on:

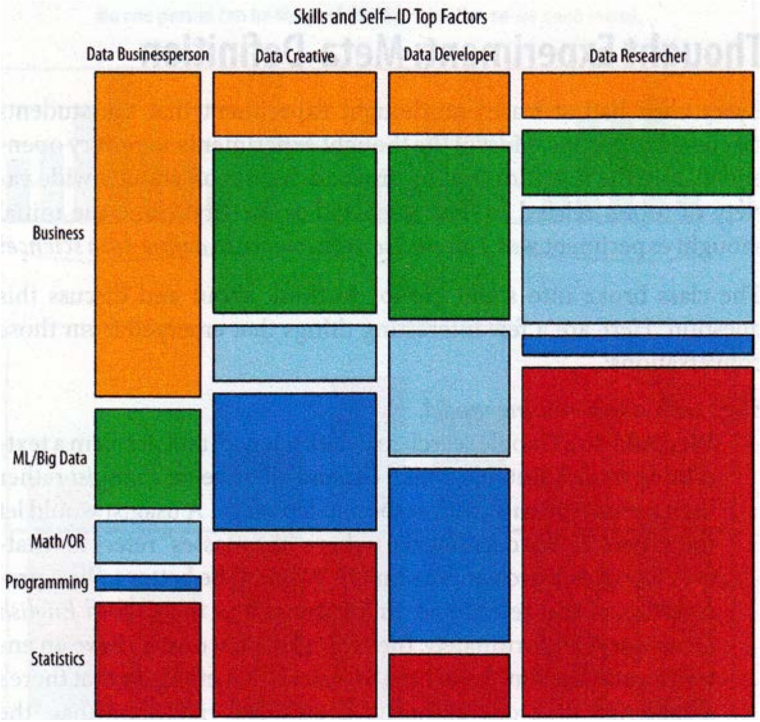
- Mathematical representations
- Computer science.

There are several skill sets needed:

- Mathematics and Statistics knowledge
- Hacking Skills (parsing, scraping, and formatting data)
- Substantive expertise


Hacking Skills
Math & Statistics Knowledge
Machine Learning
Data Science
Danger Zone!
Traditional Research
Substantive Expertise

Results of a survey in the subfields of data science



Data Science Process

Data Collection

Feature Extraction and Data Cleaning

Analytical Processing and Algorithms

Data Science Process

Data Collection

Specialized hardware

Manual labor

Software tools

Data Collection Phase

Application-specific

Critically important because good choices at this stage may significantly impact the data mining process.

After the collection phase, the data is often stored in a database or *data warehouse* for processing.

Data Science Process

Feature Extraction and Data Cleaning

Data Collection output

- Various types of data
- Free-form documents

Requirement:

Transform data into a standard format for analysis, such as

- multidimensional,
- time series,
- semi-structured format,
- other

Data Science Process

Feature Extraction and Data Cleaning

Example: Multidimensional format

Fields of the data correspond to the different measured properties that are referred as

- *features*,
- *attributes*,
- *dimensions*.

Feature Extraction Phase

- often performed in parallel with data cleaning
- missing and erroneous parts: estimated or corrected.

After the Feature Extraction Phase, data can be stored in a database for processing.

Data Science Process

Analytical Processing and Algorithms

- Final part of the process

  Design effective analytical methods from the processed data.

- Standard data mining problems:

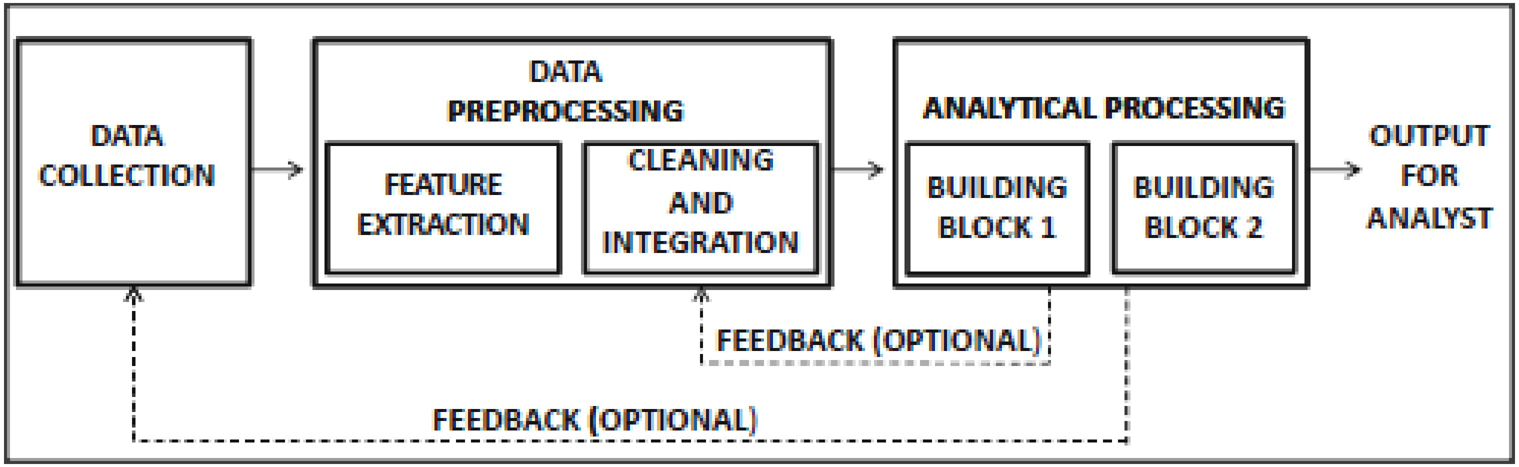
  Association pattern mining
  Clustering
  Classification
  Outlier Detection

If not possible to apply any of these methods directly

- Many applications can be broken up into components
- Use these different building blocks.

Data Science Process

The data processing pipeline



Examples of different types of data

WWW

- Number of documents in order of billions
- Web access logs at servers and customer behavior
- Linked structure of Web: *Web graph*

Financial Interactions

- ATM and credit card transactions
- Financial Statements
- Financial News

Sensor Technologies and the Internet of Things

- Implemented in low-cost sensors, smartphones, etc…

Sample Data Processing

Consider the following scenario for a retailer that has

- Web logs corresponding to customer access.
- Indication of interest in that particular product.
- Demographic profiles for the different customers.

Objective:

- Make targeted product recommendations to customers using the customer demographics and buying behavior.

Sample Solution Pipeline

- Collect relevant data from two different sources.

  Web logs at the site

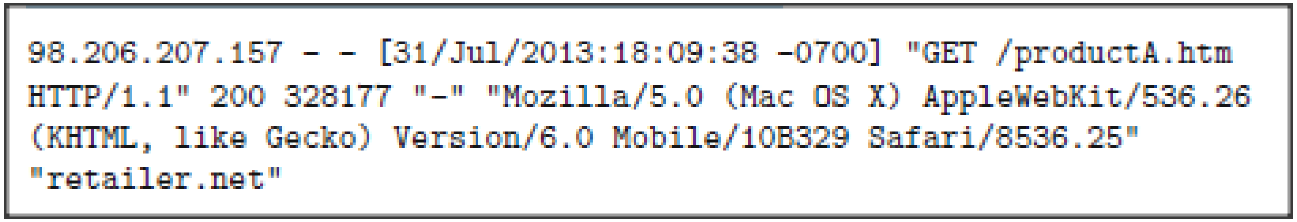
  Demographic information within the retailer database.

Issue:

- These data sets are in a very different format and cannot easily be used together for processing.

Sample Data Processing

For example, consider a sample log entry of the following form:



```
98.206.207.157 - - [31/Jul/2013:18:09:38 -0700] "GET /productA.htm
HTTP/1.1" 200 328177 "-" "Mozilla/5.0 (Mac OS X) AppleWebKit/536.26
(KHTML, like Gecko) Version/6.0 Mobile/10B329 Safari/8536.25"
"retailer.net"
```

- The design algorithm for filtering is a *cleaning and extraction* process.
- Retain only the relevant information is the *feature extraction* process.
- Attributes are added to these records for the retailer’s database containing demographic information in a *data integration phase*.
- Missing entries from the demographic records need to be estimated further *data cleaning*.

Result: a single data set containing attributes for the customer

- Demographics
- Customer access

Sample Data Processing

Next Steps:

- Determine similar groups of customers
- Make recommendations on the basis of the buying behavior of these similar groups.

In particular, the *building block* of clustering is used to determine similar groups.

- For a given customer, the most frequent items accessed by the customers in that group are recommended.

Sample Data Processing

The entire data mining process is both a scientific and an art form

Based on

- skill of the analyst
- cannot be fully captured by a single technique or building block.

In practice, this skill can be learned only by working with a *diversity of applications* over *different scenarios* and *data types*.

The Data Preprocessing Phase

Begins after data collection. Steps:

Feature extraction:

Examples:

- Large volumes of raw documents, system logs, or commercial transactions with little guidance on how these raw data should be transformed for processing.

Relevant features:

Example:

- in a credit-card fraud detection application good indicators of fraud: amount of a charge, frequency, and location .

However, many other features may be poorer indicators of fraud.

Need: *specific application domain knowledge* !!

The Data Preprocessing Phase

Data cleaning:

Issues: extracted data may have erroneous or missing entries.

Solution: Some records may have to be

- Dropped
- Missing entries estimated.
- Inconsistencies removed.

The Data Preprocessing Phase

Feature selection and transformation:

- Issue: data mining algorithms do not work effectively on *high dimensional* data
- Many of the high-dimensional features are *noisy* and may add *errors* to the data mining process.
- Methods are used to either remove irrelevant features or *transform* the current set of features to a *new data space*.
- Data set with a particular set of attributes may be transformed into a data set with another set of attributes of the same or a different type.

The Analytical Phase

Major challenges:

- Each data mining application is unique
- Difficult to create general and reusable techniques across different applications, but some data mining formulations are repeatedly used in the context of different applications:

  Major *superproblems* or *building blocks* of the data mining process.
- It is dependent on the *skill* and *experience* of the analyst to determine how these different formulations may be used in the context of a particular data mining application.

The Basic Data Types

Two broad types of data:

Nondependency-oriented data:

- Simple data types such as multidimensional data or text data.
- In these cases, the data records do not have any specified dependencies between either the data items or the attributes.

Example:

- Set of demographic records about individuals containing their age, gender, and ZIP code.

Dependency-oriented data:

- Implicit or explicit relationships may exist between data items.

Example:

- A social network data set contains a set of *vertices* (data items) that are connected together by a set of *edges* (relationships).
- Time series contains implicit dependencies.

Nondependency-Oriented Data

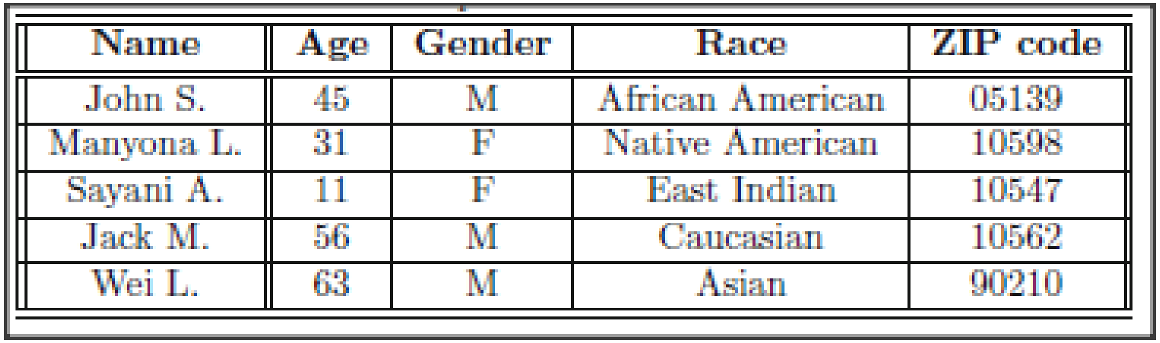
The simplest form of data: *multidimensional data*.

Data typically contains a set of *records*.

A record is also referred to as a

- *data point*, *instance*, *example*, *transaction*, *entity*, *tuple*, *object*, or *feature-vector*, depending on the application at hand.

- Each record contains a set of *fields*, which are also referred to as *attributes*, *dimensions*, and *features*.



Name	Age	Gender	Race	ZIP code
John S.	45	M	African American	05139
Manyona L.	31	F	Native American	10598
Sayani A.	11	F	East Indian	10547
Jack M.	56	M	Caucasian	10562
Wei L.	63	M	Asian	90210

Nondependency-Oriented Data

Definition (Multidimensional Data)

- A multidimensional data set D is a set of *n* records, $\overline{X_1} \cdots \overline{X_n}$, such that each record $\overline{X}_i$ contains a set of *d* features denoted by $\left(x_i^1 \cdots x_i^d\right)$.

Quantitative Multidimensional Data

- Data in which all fields are quantitative is also referred to as *quantitative data* or *numeric data*.

Categorical Data

- Many data sets in real applications may contain categorical attributes that take on *discrete unordered* values.
- If each value of x_i^j is categorical, then such data are referred to as *unordered discrete-valued* or *categorical*.

Mixed Attribute Data

- In the case of *mixed attribute* data, there is a combination of categorical and numeric attributes.

Nondependency-Oriented Data

Binary and Set Data

Can be considered a special case of

- Multidimensional categorical data
  - each categorical attribute may take on one of at most two discrete values
- Multidimensional quantitative data.
  - ordering exists between the two values

Binary data is also a representation of setwise data

- Each attribute is treated as a set element indicator.
- A value of 1 indicates that the element should be included in the set.

$$I(x) = \begin{cases} 1 & \text{if} \quad x \in X \\ 0 & \text{otherwise} \end{cases}$$

Nondependency-Oriented Data

Text Data

Text data can be represented:

- String Format
- Multidimensional Data Format

String Format

- In raw form, a text document corresponds to a *string*. This is a dependency-oriented data type.
- Each string is a sequence of characters (or words) corresponding to the document.
- However, text documents are rarely represented as strings.

Nondependency-Oriented Data

Text Data

Multidimensional Data Format

- *Vector-space representation*

  - frequencies of the words (or *terms*) in the document are used for analysis.

  - frequencies are typically normalized with statistics such as the length of the document, or the frequencies of the individual words in the collection.

  $n \times d$ data matrix for a text collection with *n* documents and *d* terms is referred to as a *document-term matrix*.

- Issue: Most attributes take on zero values, and only a few attributes have nonzero values.

  - This phenomenon is referred to as *data sparsity*.

Dependency-Oriented Data

In practice, the different data values may be (implicitly) related to each other

- Temporally
- Spatially
- Through explicit network relationship links between the data items.
- Knowledge about *preexisting* dependencies greatly changes the data analysis process.

Dependency-Oriented Data

Types of dependencies:

- *Implicit dependencies:* dependencies between data items are not explicitly specified but are known to *typically* exist in that domain.
- *Explicit dependencies:* This typically refers to graph or network data in which edges are used to specify explicit relationships.

Dependency-Oriented Data

Time-series data

- Values are typically generated by continuous/discrete measurement over time.

- Typically have *implicit* dependencies built into the values received over time.

The attributes are classified into two types:

- *Contextual attributes*

  - These are the attributes that define the *context* on the basis of which the implicit dependencies occur in the data

- *Behavioral attributes*

  - These represent the values that are measured in a particular context.

Dependency-Oriented Data

Definition (Multivariate Time-Series Data)

A time series of length *n* and dimensionality *d* contains *d* numeric features at each of *n* time stamps $t_1 \cdots t_n$. Each timestamp contains a component for each of the d series. Therefore, the set of values received at time stamp $t_i$is $\overline{Y_i} = (y_i^1 \cdots y_i^d)$. The value of the j^{th}series at time stamp t_i is y_i^j.

Dependency-Oriented Data

Discrete Sequences and Strings

- Categorical analog of time-series data.
- Contextual attribute:

  - time stamp

  - position index in the ordering.
- Behavioral attribute is a categorical value.

Dependency-Oriented Data

Definition (Multivariate Discrete Sequence Data)

A discrete sequence of length *n* and dimensionality *d* contains *d* discrete feature values at each of *n* different time stamps $t_1 \cdots t_n$. Each of the *n* components $\overline{Y_i}$ contains *d* discrete behavioral attributes $\left(y_i^1 \cdots y_i^d\right)$ collected at the i^{th} time-stamp.

- A particularly common case in sequence data is the *univariate* scenario, in which of *d* = 1. Such sequence data are also referred to as *strings*.

Dependency-Oriented Data

Spatial Data

Many non-spatial attributes are measured at spatial locations.

- Example: credit card transactions collected to forecast the spending behavior. In such cases:

  - the spatial coordinates correspond to contextual attributes

  - transaction amounts correspond to the behavioral attributes.

Dependency-Oriented Data

Definition (Spatial Data)

A *d*-dimensional spatial data record contains *d* behavioral attributes and one or more contextual attributes containing the spatial location. Therefore, a *d*-dimensional spatial data set is a set of d dimensional records $\overline{X_1} \cdots \overline{X_n}$, together with a set of n locations $L_1 \cdots L_n$, such that the record $\overline{X_i}$ is associated with the location L_i.

Dependency-Oriented Data

Spatio-Temporal Data

- Contains both spatial and temporal attributes.
- The precise nature of the data also depends on which of the attributes are contextual and which are behavioral.

Both spatial and temporal attributes are contextual:

- Spatial and temporal dynamics of particular behavioral attributes are measured simultaneously.

The temporal attribute is contextual, whereas the spatial attributes are behavioral:

- Similar to time-series data, but the spatial nature of the behavioral attributes also provides better interpretability and more focused analysis in many scenarios.

Dependency-Oriented Data

Network and Graph Data

- Data values may correspond to nodes in the network
- Relationships among the data values corresponds to the edges in the network.

Definition (Network Data)

A network *G = (N,A)* contains a set of nodes *N* and a set of edges *A*, where the edges in *A* represent the relationships between the nodes. In some cases, an attribute set $\overline{\mathrm{X}_\mathrm{i}}$ may be associated with node *i*, or an attribute set $\overline{\mathrm{Y}_{\mathrm{ij}}}$ may be associated with edge (i,j).

Dependency-Oriented Data

Network and Graph Data

Examples of graph data representation:

Web graph:

- Nodes correspond to the Web pages
- Edges correspond to hyperlinks.

Social networks:

- Nodes correspond to social network users
- Edges correspond to friendship links.

Email or chat-messenger networks:

- Nodes correspond to email addresses
- Edges may have content associated with them.

The Major Building Blocks: A Bird's Eye View

Four problems in data mining are considered fundamental to the data analysis process:

- *Clustering*
- *Classification*
- *Association pattern mining*
- *Outlier detection*

The Major Building Blocks: A Bird’s Eye View

Consider a multidimensional database *D (data matrix)*, with *n* records, and *d* attributes.

- General Task: Find relationships between the entries in the data matrix that are either *unusually frequent* or *unusually infrequent*.

Relationships between data items:

- *Relationships between columns*
- *Relationships between rows*

The Major Building Blocks: A Bird’s Eye View

Example: Relationships between columns

- Frequent or infrequent relationships between the values in a particular row are determined.

  - This maps into either the positive or negative *association pattern* problem.

- Relationships can be used to predict the value of a column, when the value of that column is unknown.

  - This problem is referred to as *data classification*.

A process is referred to as *supervised* when it is based on treating a particular attribute as special and predicting it.

The Major Building Blocks: A Bird’s Eye View

Example: Relationships between rows

- Determine subsets of rows, in which the values in the corresponding columns are related.

- In cases where these subsets are similar, the corresponding problem is referred to as *clustering*.

- If the entries in a row are very different from the corresponding entries in other rows, then this problem is referred to as *outlier analysis*.

Association Pattern Mining

Definition (Association Pattern Mining)

Given a binary $n \times d$ data matrix *D*, determine all subsets of columns such that all the values in these columns take on the value of *1* for at least a fraction *s* of the rows in the matrix.

- Relative frequency of a pattern is referred to as its support.
- Fraction *s* is referred to as the minimum support.

Patterns that satisfy the minimum support requirement are often referred to as *frequent patterns*, or *frequent item sets*.

Data Clustering

Definition (Data Clustering)

Given a data matrix *D* (database *D*), partition its rows (records) into sets $C_1 \cdots C_k$, such that the rows (records) in each cluster are “similar” to one another.

Example applications:

- *Customer segmentation*
- *Data summarization*
- *Application to other data mining problems*

Outlier Detection

An outlier is a data point that is significantly different from the remaining data.

- a.k.a *abnormalities*, *discordants*, *deviants*, or *anomalies* in the data mining and statistics literature.

Definition (Outlier Detection)
Given a data matrix *D*, determine the rows of the data matrix that are very different from the remaining rows in the matrix.

Example applications:

- *Intrusion-detection systems*
- *Credit card fraud*
- *Interesting sensor events*
- *Law enforcement*

Data Classification

Characteristics:

- A particular feature of interest in the data is referred to as the *class label*.
- *Supervised problem* (the relationships of the remaining features in the data with respect to this special feature are *learned)*.

The data used to learn these relationships is referred to as the *training data*.

Training model constructed:

- Predict class labels.
- Estimate class labels for records, where the label is missing.

Data Classification

Definition (Data Classification)

Given an $n \times d$ training data matrix *D* (database *D*), and a class label value in $\{1 \cdots k\}$ associated with each of the *n* rows in *D* (records in *D*), create a training model *M*, which can be used to predict the class label of a *d*-dimensional record $\bar{Y} \notin D$.

Data Classification

Relationship between the clustering and the classification:

- Clustering: the data partitioned into *k* groups on the basis of similarity
- Classification: a (test) record is categorized into one of *k* groups, except that this is achieved by learning a model from a training database *D*, rather than based of similarity.

From a learning perspective:

- Clustering is often referred to as *unsupervised learning.*
- Classification problem is referred to as *supervised learning*.

Scalability Issues and the Streaming Scenario

Important scenarios for scalability:

The data is stored on one or more machines, but it is *too large* to process efficiently.

- Need algorithms that minimizes the random access to the disk.
- For very large data sets, big data frameworks: MapReduce or similar.

Scalability Issues and the Streaming Scenario

Two important scenarios for scalability:

- Data generated *continuously* over time in high volume, impractical to store it entirely.

- *Data streams* have to be processed with a real-time algorithm approach. The major challenges:

- *One-pass constraint:* The algorithm needs to process the entire data set in one pass.

- *Concept drift:* Data distribution can change over time.

Exercise: Exploratory Data Analysis

- The datasets provided *nyt1.csv*, *nyt2.csv* represent two (simulated) day's worth of ads shown and clicks recorded on the *New York Times* homepage. Each row represents a single user. There are 5 columns: age, gender (0=female, 1=male), number impressions, number clicks, and logged-in.

Use R to handle this data. Perform some exploratory data analysis:

- Create a new variable, *age group*, that categorizes users as "<18", "18-24", "25-34", "35-44", "45-54", "55-64", and "65+".

For a single day:

- Plot the distribution of number of impressions and click-through-rate (CTR = #clicks / #impressions) for these six age categories

- Define a new variable to segment or categorize users based on their click behavior.

- Explore the data and make visual and quantitative comparisons across user segments/demographics (<18-year-old males versus <18-year-old females or logged-in versus not, for example).

Exercise: Exploratory Data Analysis

For a single day:

- Create metrics/measurements/statistics that summarize the data. Examples of potential metrics include CTR, quantiles, mean, median, variance, and max, and these can be calculated across the various user segments. Be selective. Think about what will be important to track over time – what will compress the data, but still capture user behavior.

- Extend your analysis across days. Visualize some metrics and distributions over time.

- Describe and interpret any patterns you find.

Exercise: RealDirect

Explore realdirect.com thinking about how buyers and sellers would navigate, and how the website is organized. Use rollingsales_Brooklyn.xls dataset provided.

Do the following:

- Load in and clean up the data. Next, conduct exploratory data analysis in order to find out where there are outliers or missing values, decide how you will treat them, make sure the dates are formatted correctly, make sure values you think are numerical are being treated as such, etc.
- Once the data is in good shape, conduct exploratory data analysis to visualize and make comparisons:
- Across neighborhoods
- Across time
- If you have time, start looking for meaningful patterns in this dataset
- Summarize findings

Repeat the exercise for rollingsales_manhattan.xls

1870

STEVENS
INSTITUTE of TECHNOLOGY
School of Business

Thank You

Dragos Bozdog

For academic use only.