End to End Framework for Developing Machine Learning Solution

Presenter: Karan Vijay Singh

UNIVERSITY OF WATERLOO
FACULTY OF MATHEMATICS

Introduction to Machine Learning

- It is a discipline where computer programs make predictions or draw insights based on patterns they identify in data and are able to improve those insights with experience — without humans explicitly telling them how to do so.

- At a conceptual level, we're building a machine that given a certain set of inputs will produce a certain desired output by finding patterns in data and learning from it.

- For example, given the area in square feet,furnished, address, parking and number of bedrooms (the input) we're looking to predict a home's sale price (the output).

Why designing a Machine Learning Solution is different from designing a Software solution?

For software product,

- Testing a software system is relatively straight-forward as compared to testing an ML solution as you know the desired outcomes in software.
- Also in Traditional software engineering, using encapsulation and modular design helps us to create maintainable code and it is easy to make isolated changes and improvements.

But in ML,

- Desired outcome depends on kind of problem you are trying to solve.
- You may get the output you are looking for but it's not always the case.
- Desired behavior cannot be effectively implemented in software logic without dependency on external data.

Motivation

- Google Scholar Search -> No papers on RE for ML but lot on ML for RE.
- No authoritative source available that can be consulted when designing a machine learning solution.
- This is an attempt to develop an end to end framework for developing machine learning solutions.

Machine Learning Problem Framework

Presenters:
Bikramjeet Singh(20752928)
Priyansh Narang (20716980)

RE for ML/ ML for RE feat. Google Scholar

- We tried multiple keywords to review past work done in this space: requirement engineering, requirement elicitation, SDLC for ML
- No credible source for RE for ML
- Several papers where authors have used techniques from ML to improve Requirement Engineering:
  - Estimation of effort for tasks
  - Prioritizing requirements
- Few online publishing platforms have articles about the intersection of SE and ML
- This is an attempt at developing and end-to-end framework for systems leveraging Machine Learning

Formal definition

“A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E”

Type of Machine Learning Problems

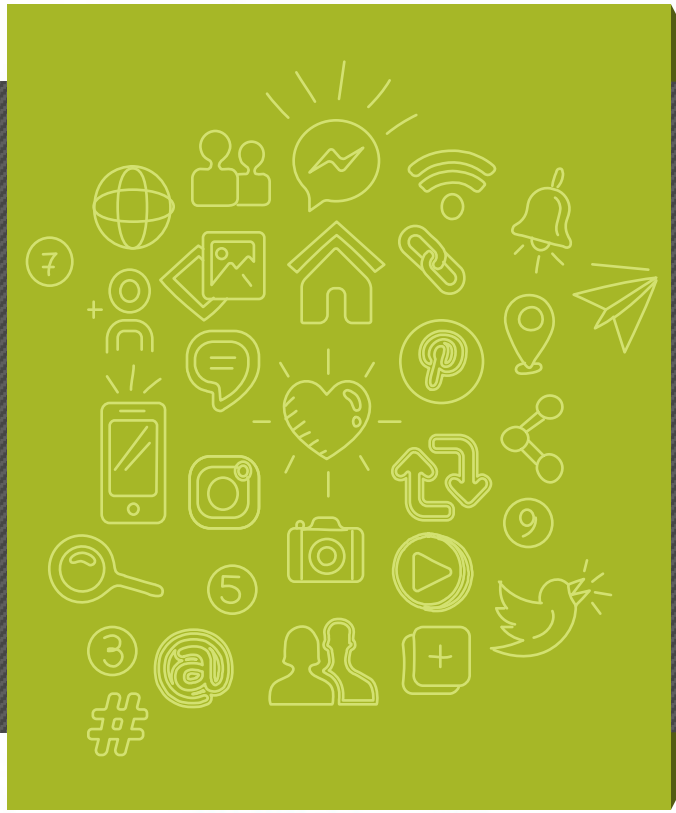
Type of ML Problem	Description	Example
Classification	Pick one of N labels	cat, dog, horse, or bear
Regression	Predict numerical values	click-through rate
Clustering	Group similar examples	most relevant documents (unsupervised)
Association rule learning	Infer likely association patterns in data	If you buy hamburger buns, you're likely to buy hamburgers (unsupervised)
Structured output	Create complex output	natural language parse trees, image recognition bounding boxes

3. What are your success metrics?

- How do you know the system has succeeded? Failed?
- Phrased independently of evaluation metrics
- Tied to the ideal outcome
- Domain/product/team specific
- Are the metrics measurable?
- When are you able to measure them?
- How long will it take for you to know that system is a success or failure?
- Example: Predict the credit limit within 10% range of the manual process
- Example: Reduce the time taken to approve the user for a certain credit limit by 90%

4. What’s the ideal output?

- Write the output you want your models to produce in plain english
- The output must be quantifiable that the machine is capable of producing
- For instance: “User did not enjoy the article” produces much worse results than “User down-voted the article”
- For your ideal output, can you obtain example outputs for training data?



RE for Data Cleaning with Machine Learning
CS 846

Presenter: Ishank Jain

UNIVERSITY OF WATERLOO
FACULTY OF MATHEMATICS

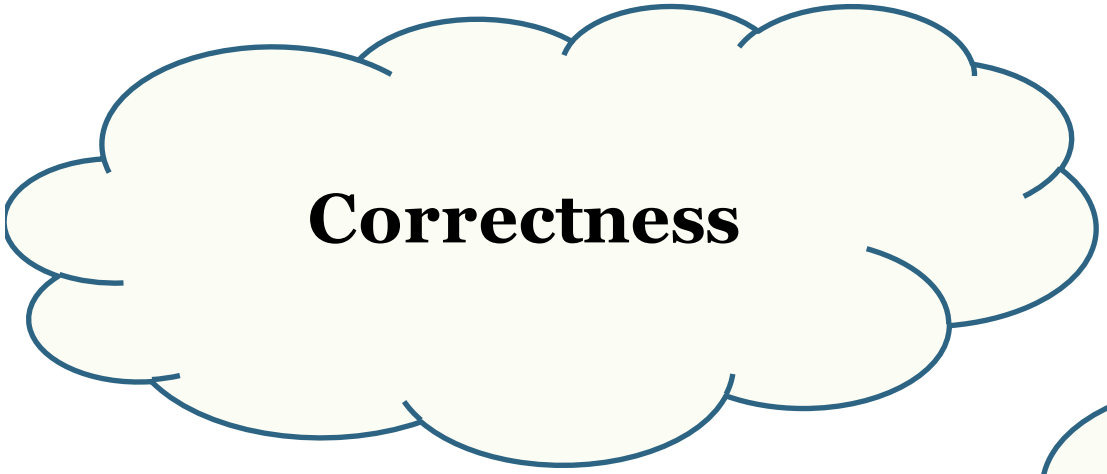
REQUIRED CHARACTERISTICS

Scripting languages that are appropriate for skilled and unskilled programmers.

Systems will need to have automated algorithms with human help only when necessary.

New data sources must be integrated incrementally as they are uncovered.

CHALLENGES



REQUIREMENTS

- Datasets:
  - Training data
  - Clean data
  - Test data
- Rules and constraints to detect dirty cells.
- Machine learning architecture: this may include
  - Clustering algorithm to detect outliers, dirty cells. For instance ActiveClean, Tamr.
  - Neural network based algorithm which is trained on a feature graph model to generate potential domain, for instance, HoloClean.
  - Classification and boosting algorithm (SVM, Naïve Bais etc.) to assign the correct class label from the domain based on a loss minimization function or to detect duplicates, for instance, BoostClean and Tamr.

REQUIREMENT ENGINEERING FOR DATABASES AND DATA CLEANING

PRESENTED TO: PROF. DAN BERRY
PRESENTED BY: MARIAN BOKTOR
COURSE CODE: CS846

UNIVERSITY OF WATERLOO
FACULTY OF MATHEMATICS
David R. Cheriton School
of Computer Science

How to know if a database is any good?

- **Helps support and ensure the accuracy and integrity of information**
- Divides information into subject-based tables to reduce redundant data
- Provides Access with the information it requires to join the information in the tables together as needed
- Accommodates your data processing and reporting needs
- It is easy to modify and maintain without affecting other fields or tables in the database
- Information is easy to retrieve, and user applications are easy to develop and build

How to know if a database is any good?

- The database is scalable, meaning that it can be expanded to meet the changing needs of an organization
- Normalized, to minimize data redundancy, I/O redesign transaction sizes and to enforce referencing integrity
- Semantically same data have same representations in heterogeneous sources
- We have to have a single view and access to accurate and consistent data
- Quality Design: data is stored in a single logical unit instead of multiple files, maintaining data integrity and security, and finally increase the performance of the database

How to know if database software is any good?

- Easy to use
- Customizable
- Mobile-optimized
- Real-time insights
- Cloud-based
- Tailored for any company
- Multiple database formats
- Define constraints to maintain data integrity
- e.g. TeamDesk, Knack, Oracle Database Cloud Service, Microsoft Azure Cloud

Why we need a good database?

- Centralized systems
- Better management of human resource (HR) matters
- Managing customer data and relationships
- Efficient inventory tracking
- Planning for growth

High Quality Data | Criteria

- Validity
  e.g. Mandatory, Data-type, Range, Foreign-key and Unique constraints
- Accuracy
- Completeness
- Consistency
- Uniformity
- Traceability
- Timeliness