Applied Machine Learning for Business Analytics

Lecture 1: Introduction

The following lecture slides and notebook will be updated one week before the lecture.

Lecturer: ZHAO Rui

Logistics

- Check course website frequently
  - https://bt5153msba.github.io
- 100% f2f lectures
  - Attendance check would be conducted randomly
- Class hours
  - From 6:30 pm to 8:30 pm

Agenda

1. Course overview
2. What is machine learning
3. From Business Problems to ML Solutions
4. Gap between theory and production
5. How LLMs are changing the game
6. Group Projects

1. Course overview

Goals of this course

- Understand conceptually the mechanism of machine learning and data science algorithms
- Implement the whole pipeline for your ML projects
- Select appropriate machine learning tools/techniques for business applications

Learn and Improve upon the applications of machine learning

Course background and overview

- Basic ML/Data Mining models have been covered in other modules
- In BT5153:
  - “**Advanced**” architecture
  - **Hands-on** Experiences
  - In each lecture, roughly 90% Slides and **10% IPython notebooks**.
  - More **Practical** Assignments/Exams

In practice, be solution-focused, not buzzword-focused.

Models & Systems

- E2E ML System
  - Data
  - Evaluation
  - Deployment
- Representation(Deep) Learning
  - Word Embeddings
  - Transformers
  - BERT
- Large language models
- Agents

Hands-on experience

- Understanding domain, prior knowledge
- Data integration, selection, clearing, pre-processing, etc
- Learning models (little math, more intuitive ideas)
- Compare models
- Model interpretability
- Consolidating and deploying discovered knowledge
- Apply discovered knowledge to practical problems
- Work with LLMs
- Python programming is not the teaching focus

Course assessment

- In-class Quizzes (10%)
- Individual Assignments (50%)
  - Three weekly individual assignments (10% each)
  - One mini-kaggle project (20%)
- Group Project (40%)
  - Project proposal (5%)
  - Final presentation (20%)
  - Final report (15%)

In-class Quiz

- It would be used for attendance check
- Up to 5 times. 2 points each time
- If you are going to miss the following class, please email our TA Dingyu and cc me in advance. Otherwise, you will not get this 2 points if we have quiz in that lecture
  - Dingyu: dingyushi@u.nus.edu

Course Schedule

2025

Date	Topic	Content	Assignment
Fri 01/17	Introduction to Machine Learning and its Production	TBU	N.A.
Fri 01/24	From BoW to Word2Vec	TBU	Huggingface Tutorial
Fri 01/31	From Word2Vec to Transformers	TBU	Form your team & Assignment I Out
Fri 02/07	LLM and its Practices I	TBU	N.A.
Fri 02/14	LLM and its Practices II	TBU	LangChain Tutorial
Fri 02/21	LLM and its Practices III	TBU	Assignment II Out
Sun 03/02	Recess Week	N.A.	Proposal Due
Fri 03/07	Data Preparation	TBU	Assignment III Out
Fri 03/14	ML Model Modelling	TBU	Kaggle Starts
Fri 03/21	ML Model Evaluation	TBU	N.A.
Fri 03/28	NO CLASS (NUS Well-Being Day)	N.A.	N.A.
Fri 04/04	ML Model Deployment	TBU	Kaggle Competition
Fri 04/11	Why do ML Projects Fail in Business	TBU	N.A.
Fri 04/18	No CLASS (Good Friday)	N.A.	Kaggle Report
Sun 04/27	N.A.	N.A.	Presentation and Final Report Due

2026

Date	Topic	Content	Assignment
Fri 01/16	Introduction	Link	N.A.
Fri 01/23	Text Representations: BoW to Word2Vec	Link	N.A.
Fri 01/30	Transformers	Link	Assignment I Out
Fri 02/06	LLM Fundamentals	Link	Assignment I Due
Fri 02/13	Training & Scaling LLMs	Link	Assignment II Out
Fri 02/20	Inference & Reasoning	Link	Assignment II Due
Fri 02/27	Recess Week	N.A.	Proposal Due
Fri 03/06	RAG & Context Management	Link	Assignment III Out
Fri 03/13	Agent Design Patterns	Link	Assignment III Due
Fri 03/20	Agent Production&Security	Link	Kaggle Starts
Fri 03/27	ML Model Evaluation	N.A.	N.A.
Fri 04/03	Good Friday	N.A.	N.A.
Fri 04/10	ML Model Deployment	Link	Kaggle Competition
Fri 04/17	Why ML&LLM Projects Fail	Link	N.A.
Fri 04/24	N.A.	N.A.	Kaggle Report

2. What is Machine Learning

Machine Learning is Everywhere

Definition of Machine Learning

- Machine Learning is an approach to **learn** *complex pattern* from existing data and use these patterns to make **predictions** on **unseen data**.
- Therefore, there are following points to determine if a ML solution will fit your problem
    - Learn
    - Complex Pattern
    - Existing Data
    - Predictions
    - Unseen Data

Learn

- The system has the capacity to learn
  - From the data
- To apply Machine Learning, there must be something for it to learn.
  - E.g., database is not the ML System

Complex Pattern

- The patterns are complex
  - Look-up operation vs Object Detection
- What is difficult to humans is different from what is hard to machines

Complex Pattern

- There are patterns to learn
  - Should we predict the next outcome of toto?

  - Should we predict doge price?

Existing Data

- Data is available
- It is possible to collect data
- Exceptions?
  - Zero-shot learning (still trained over data from other domains)
  - Online learning

Predictions

- It is a “predictive” problem
  - We can benefit from a large quantity of cheap but approximate predictions.
- It is not only limited to estimations of values in the future
  - What is the tranx probability of this users in the following 10 days?
  - Is this cash out action a money laundry one?

Unseen Data

- Unseen data shares patterns with the training data
    - Training and unseen data should come from a similar distribution

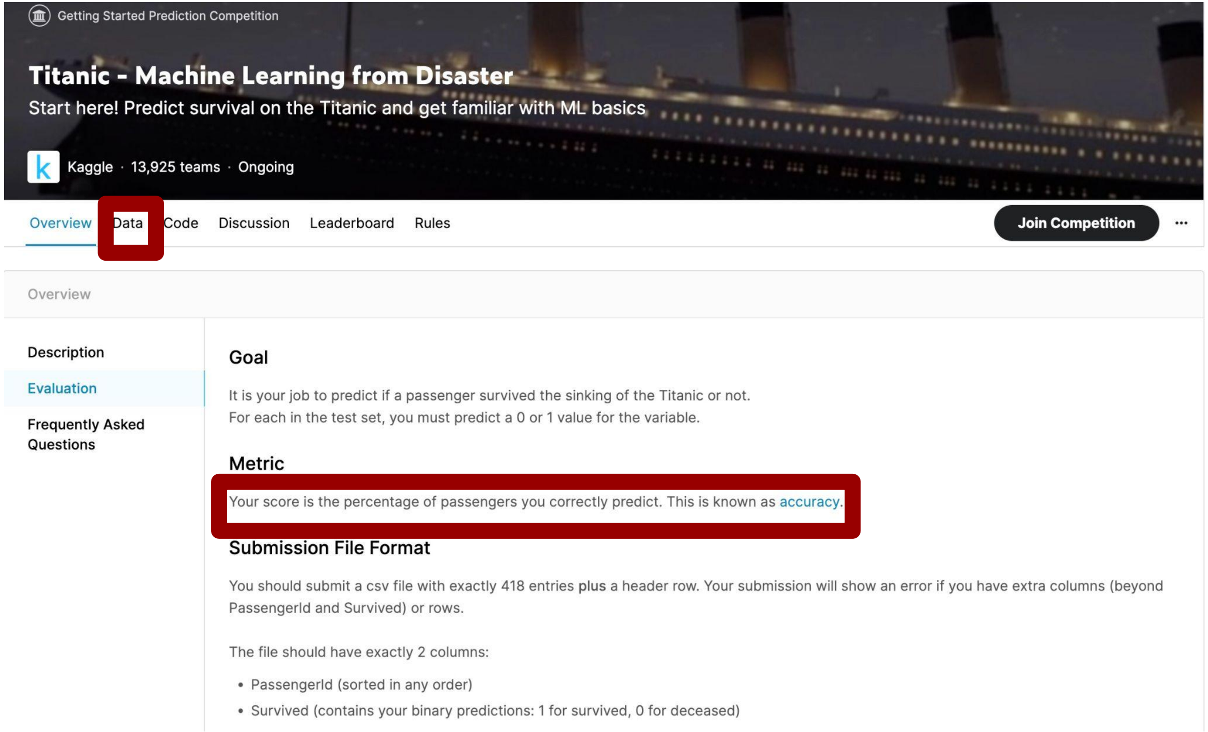
Domain Knowledge -> Solid Assumption

Other Factors to Make ML Solutions Viable

- The task is repetitive
  - New samples keep coming
- The cost of wrong predictions is cheap
  - Recommended wrong movies
- It is at scale
  - ML models are run 24/7
- The patterns are constantly changing
  - Subject matter experts are unable to encode the complete rule-set to solve the problem

3. From Business Problems to ML Solutions

Kaggle Style ML Projects



ML Projects here start with:

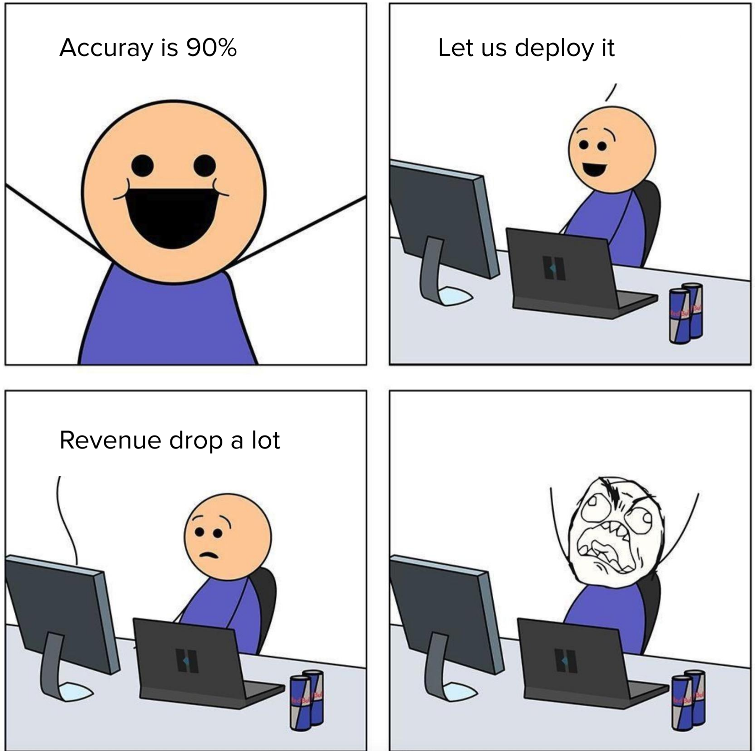
1. Dataset
2. Clearly defined metric

In Real-world

- ML or DS projects start from a business problem instead of a well-defined prediction task.
- Machine learning team is to **formulate** the business problem into the right ML problem and then **solve** it

In Real-world

Building a great ML solution to the wrong business problem is the most frustrating thing for ML/DS org.



How should we translate?

From a business problem to the right data science problem:

- Ask questions
- Explore the data to find high quality insights

A “real” example

- Assume we are working in ML/DS org at Netflix 
- Growth lead come to us with their requests 
- Then, the discussion will start as:

Based on Q1 OKR, we want to increase our users retention rate by 8%. Do you have any better ideas?

Got it. It looks quite impactful and let us work together! Do you have any hypothesis that why our users stop using Netflix?

A “real” example

Based on Q1 OKR, we want to increase our users retention rate by 8% in SEA. We would like to leverage ML solutions to achieve this goal.

Got it. The project looks quite impactful! Do you have any hypothesis that why our users stop using Netflix?

Yeah, we did some market research. Now, amazon prime video is providing lower fees.

Hmm, we also found users browsing time before they watch videos become longer.

Yeah, great sync. We have two business problems here:

- Pricing issues: our competitor is offering lower prices. The solution can be dispatching personalized discount with push notification
- Discoverability issues: our users can not easily find the videos that they are interested. I heard recommendation sys can guess what users will like. Should we also try this solution?

Thanks for the summary. Let us work on ml solutions

Hypothesis Prioritization

From the previous conversion, we are able to formulate hypothesis and create the to-do list by asking questions.

- Pricing Issues
- Discoverability Issues

Pricing Issues

- Business problem: Competitors are offering cheaper prices
- Idea: Send personalized discount with push notification
- ML Problems:
  - Who should we send notifications
  - How much is the voucher?
- ML Solutions:
  - Churn Prediction Model
  - Uplifting Models

Discoverability Issues

- Business problem: Users' conversion rate from homepage visit to video view is low
- Idea: Push personalized content to our users to increase conversion
- ML Problems:
    - Personalized recommendations
- ML Solutions:
    - Collaborative Filtering
    - Deep Learning

Source: https://research.netflix.com/research-area/recommendations

From Business Problems to ML Solutions

- The key skill would be: translating business problems into the correct data science problem
- Ask the right questions, list possible solutions, and explore the data to narrow down the list to one

From Business Problems to ML Solutions

- The key skill would be: translating business problems into the correct data science problem
- Ask the right questions, list possible solutions, and explore the data to narrow down the list to one
- Solve the problems
    - Build a dashboard
        - Build a user retention dashboard under different segments (age, geo, acquisition channels)
    - Data Exploration
        - Visualization, Group comparison (e.g,. Users from one marketing channel have a higher churn rate)
    - Train ML models
        - Should be checked only after trying the first two ideas

- **Junior DS/A are told the problems they need to solve**
- **Senior DS/A define the problems that need to be solved**

Role of ML/DS Org

- Translate abstract data into actionable business insights
- Automate and scale the above process if possible
- Be the interface to bridge biz/product and data
    - Therefore, we usually talk with two departments:
        - Biz departments: product, ops, marketing, growth
        - Engineering departments: data engineers

ML Production is not a few lines

```
import pandas as pd
from sklearn import model
df = pd.read_csv()
X = df[feature]
y = df[label]
model.train(X, y)
model.predict(new_data)
```

Data scientists should know

- SQL
  - Query and extract data
- Python
  - Main programming language
- Presentation and Visualization
  - Talk and present information in an actionable manner
- Machine Learning
  - Automate and improve operations and business decisions
- Cloud services
  - Many companies built infra in the cloud
- Deep learning/LLM libraries
  - Deal with image, video or text data
  - Keras/Pytorch/Huggingface

4. Gap between Research and Production

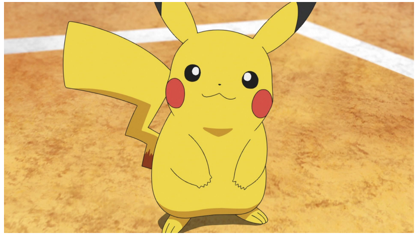
Four phases of ML Projects

- Phase 1: Before ML
- Phase 2: Simplest ML models
  - Start with a simple model that allows visibility: check hypothesis and pipeline
- Phase 3: Further Optimization
  - Different object functions
  - Feature engineering
  - More data
  - Ensembling
- Phase 4: Complex ML models

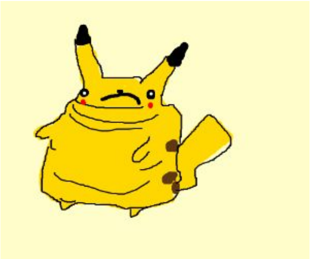
Data

- In real world, data is not perfect:
  - Missing data
  - Scale features
  - Identify outliers
  - Identify highly correlated variables
  - Identify variables with no variances
  - Check for overall hygiene
- Next week, we will discuss more about data preparation for machine learning applications.

Dataset in BT5153



Real Dataset



THE COGNITIVE CODER

By Armand Ruiz, Contributor, InfoWorld | SEP 26, 2017 7:22 AM PDT

The 80/20 data science dilemma

Most data scientists spend only 20 percent of their time on actual data analysis and 80 percent of their time finding, cleaning, and reorganizing huge amounts of data, which is an inefficient data strategy

https://www.quora.com/How-accurate-is-the-80-20-rule-as-a-Data-Scientist

Efficient Coding - Pandas as Example

- In programming, there are often many different ways to do the exact same operation, some of which are more optimized
- It is the same to data science or ML projects
- If your codes are not efficient, it would becomes a bottleneck when the scale and complexity of the problems increase
    - Pandas is the great tool for data manipulation, analysis and visualization.

How to loop effectively

- It is quite common to compute a new value from one or multiple columns in the original dataframe.
- Different codes will have different performances
- Tips are shared in this week's lab notebook

```
sum_square = lambda x, y: (x+y) ** 2
print(sum_square(2,3))
```

```
25
```

```
test_data = df_data[['X Coordinate', 'Y Coordinate']].copy()
```

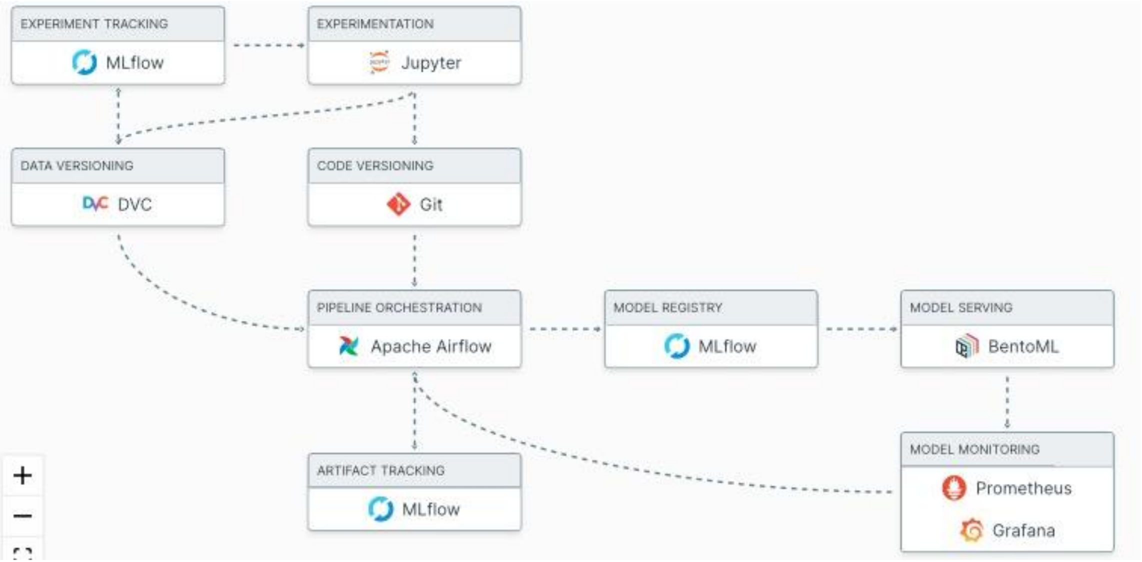
```
%timeit -r5 -n10 test_data.loc[:,'magic'] = [sum_square(value[0], value[1]) for _, value in test_data.iterrows()]
%timeit -r5 -n10 test_data.loc[:,'magic'] = test_data.apply(lambda row: sum_square(row[0], row[1]), axis=1)
%timeit -r5 -n10 test_data.loc[:,'magic']  = test_data.apply(lambda row: sum_square(row[0], row[1]), raw=True, axis=1)
%timeit -r5 -n10 test_data.loc[:,'magic']  = np.vectorize(sum_square)(test_data.iloc[:,0], test_data.iloc[:,1])
%timeit -r5 -n10 test_data.loc[:,'magic']  = np.power(test_data.iloc[:,0]+test_data.iloc[:,1], 2)
#%timeit -r5 -n10 test_data.loc[:,'magic'] = [sum_square(value[0], value[1]) for _, value in test_data.iterrows()]
```

```
470 ms ± 2.26 ms per loop (mean ± std. dev. of 5 runs, 10 loops each)
135 ms ± 3.61 ms per loop (mean ± std. dev. of 5 runs, 10 loops each)
33.4 ms ± 188 µs per loop (mean ± std. dev. of 5 runs, 10 loops each)
4.49 ms ± 62 µs per loop (mean ± std. dev. of 5 runs, 10 loops each)
271 µs ± 44.5 µs per loop (mean ± std. dev. of 5 runs, 10 loops each)
```

1700X speed-up

ML Deployment

- MLOps stack



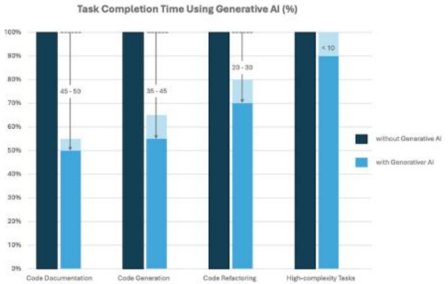
- BT5153 Hands-on notebook
  - Experiment Tracking ✅
  - Experimentation ✅
  - Data Versioning ✅
  - Code Versioning ✅
  - Pipeline Orchestration ✅
  - Runtime Engine ✅
  - Artifact Tracking ✅
  - Model Registry ✅
  - Model Serving ✅
  - Model Monitoring ❌
  - Feature Store ❌

Source: https://mymlops.com/

5. How LLMs or Agents are changing the game

Large Language Models (LLMs)

- Powering conversational and generative AI tasks
- Transforming traditional machine learning and deep learning methodologies
    - Accelerating code generation
        - Code suggestion and snippet generation
        - Automating repetitive tasks
        - Assistance with Debugging and code optimization
    - Data augmentation
        - From data and label
- ML/DL Solution vs Agent Solution



Potential of Generative AI in Software Development (McKinsey 2023)

ML VS Agent Systems

- From model-centric to system-centric
  - ML model does not take actions or make decisions beyond its narrow prediction task
  - An agent perceives its environment, reasons about what to do, take actions, and learns from feedback in a loop, which orchestrates multiple capabilities (maybe including ML models) to achieve goals
- Fraud Analytics Example
  - Traditional ML Approach:
    - Transaction -> Feature Engineering -> Model -> Fraud Score -> Human Reviewer/Rule Engine
  - AI Agent Approach:
    - Run ML models first, see fraud score ->Reasons: “High score, but amount is small and merchant is know” -> Pull customer’s last 30 days of activity -> Finds this is a regular coffee shop -> Notes customer has been traveling (recent transactions in new city) -> concludes: likely legitimate behavior in a new location

Key Paradigm Changes by Andrej Karpathy

- Reinforcement Learning from Verifiable Rewards (RLVR)
- **Jagged Intelligence**
- **LLM Apps-Cursor**
- AI that lives on your computer - Claude Code
- **Vibe Coding**
- Nano Banana

karpathy

Home Blog

2025 LLM Year in Review

20 Dec, 2025

2025 has been a strong and eventful year of progress in LLMs. The following is a list of personally notable and mildly surprising "paradigm changes" - things that altered the landscape and stood out to me conceptually.

source:
https://karpathy.bearblog.dev/year-in-review-2025/

Jagged Intelligence

- A dangerous inconsistency
  - Simultaneously genius polymath and confused grade-schooler
    - Those LLMs are doing gold-medal level work in mathematics, than failing to do simple and low-complexity tasks
  - Spike in capability near verifiable domains
    - Reinforcement Learning from Verifiable Rewards (RLVR) emerged as the new major training stage
  - AGI is not just about Intelligence but also about stability.

July 21, 2025 Research

Advanced version of Gemini with Deep Think officially achieves gold-medal standard at the International Mathematical Olympiad

Thang Luong and Edward Lockhart

Share

Deloitte to refund government, admits using AI in $440k report

Edmund Tadros and Paul Karp

Oct 5, 2025 – 7.41pm

Save Share

Gift this article

Deloitte Australia will issue a partial refund to the federal government after admitting that artificial intelligence had been used in the creation of a $440,000 report littered with errors including three nonexistent academic references and a made-up quote from a Federal Court judgement.

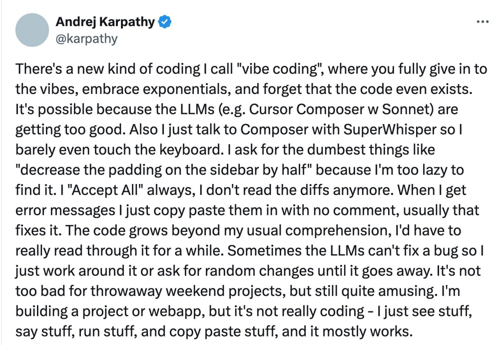
A new version of the report for the Department of Employment and Workplace Relations (DEWR) was quietly uploaded to the department's website on Friday, ahead of a long weekend across much of Australia. It features more than a dozen deletions of nonexistent references and footnotes, a rewritten reference list, and corrections to multiple typographic errors.

Cursor/New Layer of LLM Apps

- Cursor revealed a new “LLM app” layer
    - People started talking about “Cursor for X”
- LLM apps bundle and orchestrate LLM calls for downstream applications
    - Context engineering
    - Multiple LLM calls in complex Direct Acyclic Graphs (DAGs)
    - Application-specific GUI for human in the loop
    - “Autonomy sider” for control

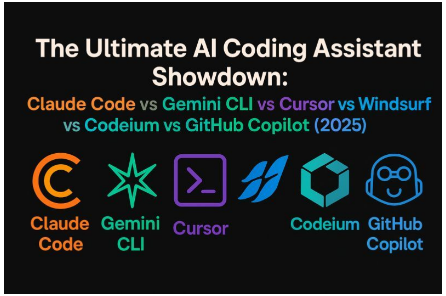
Vibe Coding

- AI Crossed Capability threshold to build programs via English
  - Forgetting that the code even exists
- Democratizes programming
  - Not reserved for highly trained professionals
- Empowers professionals to write more software
  - Code is suddenly free and discardable
- Will terraform software and alter job descriptions



Andrej Karpathy
@karpathy

There's a new kind of coding I call "vibe coding", where you fully give in to the vibes, embrace exponentials, and forget that the code even exists. It's possible because the LLMs (e.g. Cursor Composer w Sonnet) are getting too good. Also I just talk to Composer with SuperWhisper so I barely even touch the keyboard. I ask for the dumbest things like "decrease the padding on the sidebar by half" because I'm too lazy to find it. I "Accept All" always, I don't read the diffs anymore. When I get error messages I just copy paste them in with no comment, usually that fixes it. The code grows beyond my usual comprehension, I'd have to really read through it for a while. Sometimes the LLMs can't fix a bug so I just work around it or ask for random changes until it goes away. It's not too bad for throwaway weekend projects, but still quite amusing. I'm building a project or webapp, but it's not really coding - I just see stuff, say stuff, run stuff, and copy paste stuff, and it mostly works.



https://www.linkedin.com/posts/thamizhelango_ai-developers-coding assistants-activity-7359267481573601282-OVys/

How do you use LLM those days?

How Zapier' CEO is using AI

- Three examples:
  - Use AI to reverse engineer company culture from all meeting transcripts
  - Use agent to give structured feedback on every candidate and also the interviewers
  - Use grok to source talent
- <u>"Three Interns" Mindset</u>:
  - Avoid the replacement trap: a major conceptual error is only using AI to automate tasks that a person is already performing
  - You should ask: if I had three interns and infinite time, what else would I do
    - **Targeting Unsolved Problems**: many high value tasks simply do not happen today because they are too "economically expensive" or tedious for a human to follow through on consistently
    - **Unlocking Creativity**: by assuming you have infinite capacity, you can for example following up on every sales call - rather than just the three things they currently have the bandwidth for

6. Group Projects

Group project

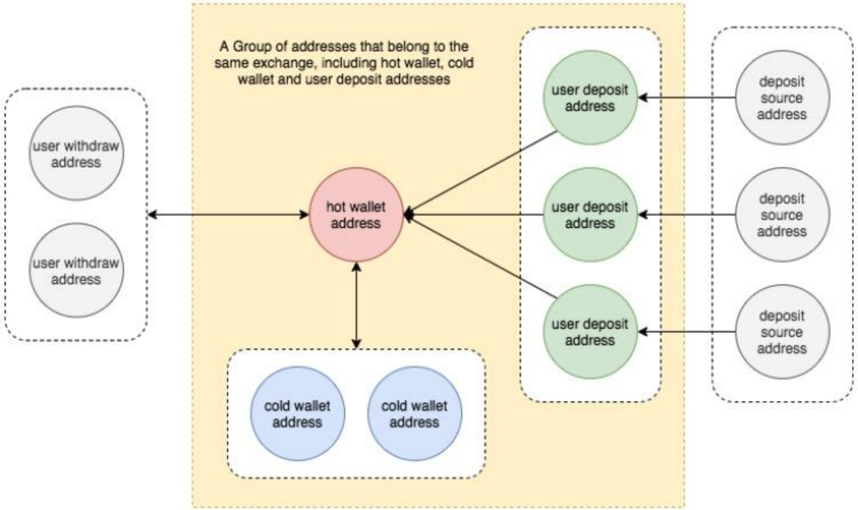
- Build an ML/DS application
- Must work in groups of four or five
- One-pager proposal + Presentation + Report
- Detailed guidelines could be found here

Paper analysis using NLP

- We collected and published all papers that were submitted from 2019 to 2025 (7 years !). Those papers discussed various kinds of applications of machine learning.

Project Hint 1

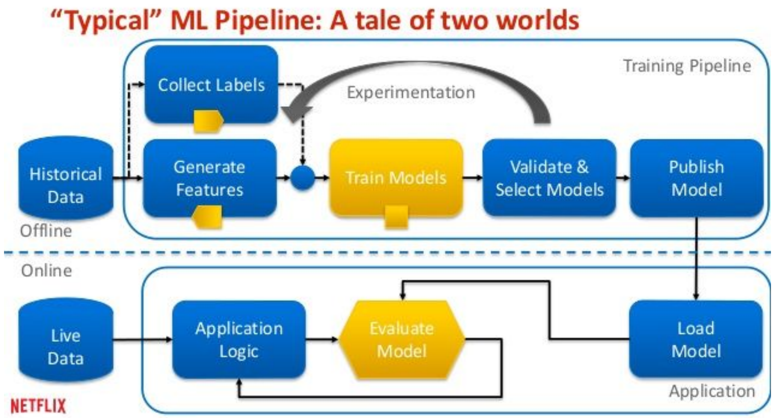
- Find a new business problem which can be solved by ML/LLM solutions
  - For example, assigning attribution labels to cryptocurrency addresses using blockchain data

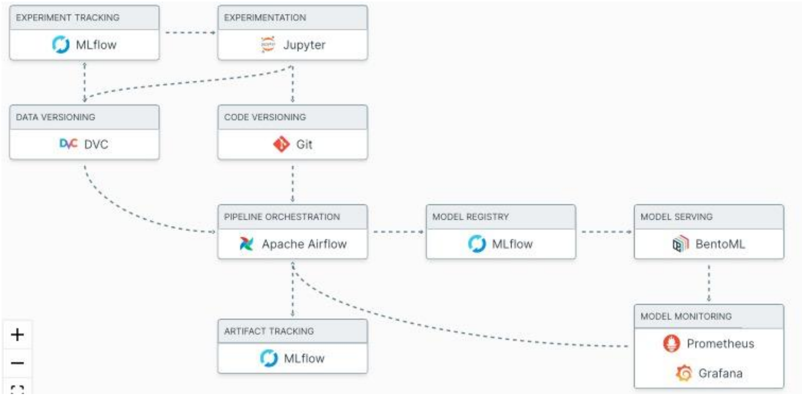


Source: https://arxiv.org/pdf/2003.13399.pdf

Project Hint 2

- Build a end-to-end ML/LLM solutions





Project Hint 3

- In-depth analysis of machine learning algorithms on one specific application
- **Try to explain the findings**

Model	MR	SST-1	SST-2	Subj	TREC	CR	MPQA
CNN-rand	76.1	45.0	82.7	89.6	91.2	79.8	83.4
CNN-static	81.0	45.5	86.8	93.0	92.8	84.7	**89.6**
CNN-non-static	**81.5**	48.0	87.2	93.4	93.6	84.3	89.5
CNN-multichannel	81.1	47.4	**88.1**	93.2	92.2	**85.0**	89.4
RAE (Socher et al., 2011)	77.7	43.2	82.4	–	–	–	86.4
MV-RNN (Socher et al., 2012)	79.0	44.4	82.9	–	–	–	–
RNTN (Socher et al., 2013)	–	45.7	85.4	–	–	–	–
DCNN (Kalchbrenner et al., 2014)	–	48.5	86.8	–	93.0	–	–
Paragraph-Vec (Le and Mikolov, 2014)	–	**48.7**	87.8	–	–	–	–
CCAE (Hermann and Blunsom, 2013)	77.8	–	–	–	–	–	87.2
Sent-Parser (Dong et al., 2014)	79.5	–	–	–	–	–	86.3
NBSVM (Wang and Manning, 2012)	79.4	–	–	93.2	–	81.8	86.3
MNB (Wang and Manning, 2012)	79.0	–	–	**93.6**	–	80.0	86.3
G-Dropout (Wang and Manning, 2013)	79.0	–	–	93.4	–	82.1	86.1
F-Dropout (Wang and Manning, 2013)	79.1	–	–	**93.6**	–	81.9	86.3
Tree-CRF (Nakagawa et al., 2010)	77.3	–	–	–	–	81.4	86.1
CRF-PR (Yang and Cardie, 2014)	–	–	–	–	–	82.7	–
SVM_S (Silva et al., 2011)	–	–	–	–	**95.0**	–	–

Source: https://arxiv.org/abs/1408.5882

Form your group

- Find your group members
- Sign-up in Canvas

Next Class: From BoW to Word2Vec

Must-read:

https://jalammar.github.io/visual-interactive-guide-basics-neural-networks/

if you are not familiar with basic neural network