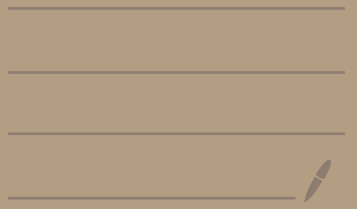
CISC 351



Week 1

- Good data is rare, missing values / noise
- Garbage in, garbage out. 45% Cleaning/loading
- Subjectivity of labeling noise
- Statistical significance vs Magnitude

Key Concepts:

- Good questions: Feasible, Novelty, Well motivated
- Discretization: Equal-Frequency, Equal width
- Normalization / Standardization: min/max. Z score $\frac{x-\mu}{\sigma}$

Missing Data

- MCAR - missing completely at random
- MAR - Missing at random
- MNAR - missing not at random

EDA

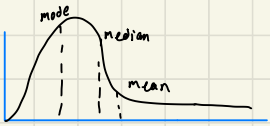
- 5 number summary min, 1q, median, 3q, max
- Correlation ≠ Causation

Hypothesis Testing

- type of test depends on data: distribution, paired / Independent.

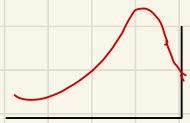
T-test: normally distributed, Mann-Whitney U test not normal.

Positive / Right Skewness



Mean > median > mode

Left / Negative Skew



Mean < median < mode

Week 2 · Regression analysis

1.) Data Exploration and Quality

- initial audit: load data using data.info and describe()
- Handle missingness, descriptive values
- Target variable analysis. if highly skewed → log transformation.

2. Feature Engineering

- feature scaling and creating new features, calculated features
- Encoding data: Nominal to dummy vars
  - ↳ mapping quality ordinal
- group by aggregation

3. Model Building interpretation

- Linear regression models with ridge + Lasso
- RMSE (Root mean square error)
- multicollinearity: Checking for redundant features using variance inflation Factor VIF and dendograms → visual
  - ↳ how much is this feature explained by other features.
- Statistical Signifigance: use OLS to find P values

4. Data Leakage

- test set into leaks to training phase
  - ↳ split data before preparation, only on trainsiting set
- use cross validation, K-fold is good

Survival Analysis

- Predicting time untill an event occurs
- Censoring: When the desired event does not happen, study concludes
- Survival functions: S(t): Probability of individual surviving past time t.
- Hazard Function: h(t): death rate at specific instant given survival up to that point.
  - Kaplan-Meier curve: estimate survival probabilities over time.
  - COX proportional: estimate survival time, using regression analysis

---

Week 3: Neural Networks

Perceptron and Nonlinearity

- Perception capable of binary classification for linearly seperable data
- mapping function used to transform data to be linearly seperable (XOR problem)
- More data = better model

Neural Network Architecture

- MLP: Multi-Layer Perceptron - forward neural net: input / output hidden layers.
- Interconnectivity: fully connected: node $L1 \rightarrow$ node $L2$.
- Tensor Notation: Scaler: 0D, Vectors: 1D, Fed mini batches

Training

- Minimize total cost function, find optimal θ parameters
- Cross entropy loss:
- Back Propagation: Primary methods for adapting weights. Uses chain rule to calculate gradient. update network based on error
- Optimization means global min of cost function
- Activation function introduces non linearity: Sigmoid, ReLU etc.
- Softmax layer: Used to output probabilities.
- Hand Crafted vs. Learnable Kernels: SVMS rely on H.C.K
  - ↳ Deep learning uses learnable Kernels

Week 3- Optimization

Challenges

- gradient Desc is too expensive
  - summing entire datasets, input features, layers and parameters
    - ↳ many iterations

Key Optimization Algorithm

- GD (Grad Desc) - forward-back pass for entire dataset
- Stochastic gradient descent: use subset of data
  - ↳ single sample or mini batch SGD
- momentum: speed/velocity to training
- Nesterov momentum: lookahead gradient step

Adaptive Learning Rate

- Diff Parameters should learn at diff speeds
- Adagrad: large gradient, smaller features
- Adadelta: uses limited history
- RMSProp: exponentially decaying of avg gradients
- Adam: Momentum + RMSProp

Core Mechanisms

- Convolutional Layer: uses filters (Kernels) to extract feature map
- Shared weight: increase efficiency
- Hyperparameters: Stride: step size, Padding: adding rows/cols, dilation: increasing space between kernels.
- Pooling Layer: Max pooling reduces number of hidden units

Topic 5 - GenAI and Foundational models

- GenAI - AI that creates new content (text, images, audio) can do tasks much faster
- Discriminative models: Predict labels, using labeled data → classify
- Generative models: learn data distribution by predicting next token (word) → create
- GenAI projects nowadays usually reuse models and fine tune rather than pretrain.
- Issues in pre-trained models → missing attributes, security risks
- Foundational models: large scale trained on wide range of downstream tasks.
- Issues in genAI → hallucinations, bias, security risks, costly
- Homogenization → everybody uses same few base models, faster development but less diverse
- Chinchilla scaling law → training dataset size should be 20 times number of parameters,
  ↳ smaller models can perform as good as larger

LLM Architecture and Training

- transformer architecture → 3 training objectives
- Autoencoder: uses masked language to predict missing words in a sentance.
- Autoregressive: Predicts next token in sequence.
- Sequence to sequence: reconstructs spans of text (Input → output transformation)

Decoding - Picking tokens

High temp: more variance / risky response

Low temp: low variance / safer response

- greedy: highest probability
- top-K: pick from top K words (more random)
- top-p: pick smalest set of words covering probability p

Application Strategies

- ICL: in context learning: refining prompts to get better performace without updating parameters, give zero-shot, one-shot, few-shot examples → cost effective, may not work for smaller models
- RAG: uses domain specific knowledge, uses vectorized database.
- Fine Tuning: Full fine tuning - very expensive. PEFT - uses only 1% param updates
- Emergent Abilities: new skills appear as size grows.

Advanced Reasoning and Decoding

COT - Chain of thought - multi-step problem breakdown

Self-Consistency cot - explore multiple paths - uses majority vote

Tot - Tree of thoughts - explore thoughts, allows back-tracking

Trustworthiness

RLHF - reinforcement learning from human feedback : aligns model with human values and reduce harmfull content

Rouge - Sommarization | Bleu - translation | BertScore - semantic relation

In-Practise

- Best results come from: Prompting → RAG → Fine-tuning → Post Processing

Topic 6 - Tabular Mining Data

- real world data is tabular

Handling Data Imbalances

- to balance, we use over sampling (copy minority data) or under sampling (deleting common data)
- use SMOTE - creates new synthetic data points
- these techinaues must be tuned.
- common methods: grid search, bayesian, Evolutionary

Common Table Tasks

- table preprocessing (cleaning): errors, matching rows, filling in missing values
- table understanding: match schema to known formats
- table analysis: table QA, fact verification, text to sql. (deeper dive)

Model Architecture for Tables

- Encoders - analyze, classify, match data. Read table, create embedding, trained on table structure
- Encoders - Decoders: need to produce output text/code. table input → something generated
- Vision encoders - read tables from images
- Decoder LLM_s (GPT) - interact + reasoning

Advanced LLM Strategies

- LLM's are powerful, but can be slow and expensive
- Prompt engineering: using few-shot examples or Cot reasoning for complex table
- task decomposition: breaking big question into sub task, getting right columns
- compression (SpreadsheetLLM): compress and remove empty space using inverted index to store unique values once
- Specalized models: TableGPT and Jellyfish are fine tuned on table tasks

Table RAG

- RAG is used to find right information
- Query Rewriting: turns question into specific cell query, to find exact data needed
- Retrievers: use key words to grab most relevant rows or tables. Spealized retrievers aren't always better than standard text-based.

Retrievals - User query → embed → retrieve relevant rows → LLM → Answer

Topic 7 - Text Data Mining

- how computers attempt to understand, process and generate human language through NLP

Challenge in Language

- human language is ambigous, slang/typos, relies on context/sarcasm (many interpretations)
- data centric challenges such as bias and ethical issues.

Text Preprocessing (cleaning data)

- tokenization: Breaking Sentances into tokens
- Stop-word removal: deleting common filler words: "the" "is" "a"
- Stemming and Lemmatization: Chopping off end of word to find the "root" of word

Traditional Text Models

- Bag of words (Bow): treats document like multi-set of words, ignoring order they appear in
- TF-IDF: weighting system that makes word more important, if appears more often
- Vector space model: turns document and Searches into vectors then uses cosine similarity to find best users query

Word Meaning (word Embeddings)

- Taxonomy (word Net): database that groups synonyms to show relation
- word embeddings (word2vec): learns next word based on neighboring words/Sentences.
- Evolution: models have become more context-sensitive, where meaning changes based on context.

Raw text → Tokenization → Embeddings → Transformer → Probabilities → Sampling → Output