Analyzing and Forecasting Consumer Price Index (CPI) in the U.S. Using Data Mining Approach

Nathaniel Getachew, Gan Fang, Jerry Mann, Xiao Luo

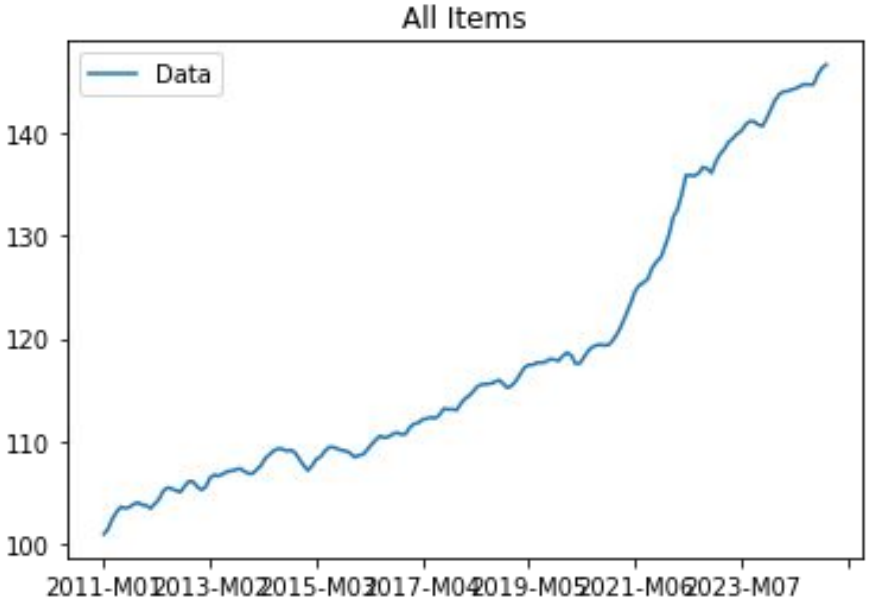
Introduction

- Consumer Price Index(CPI) - a metric for the prices of various goods
- Dataset: International Monetary Fund
- CPI calculated by U.S. Bureau of Labor Statistics

U.S. BUREAU OF LABOR STATISTICS

CPI Calculation & Motivation

- Prices of “baskets”(categories) of goods are used to calculate category CPI
- Category CPI’s are weighted and used to calculate overall CPI per country
  - Weights change every 2 years
  - Data used to calculate category CPIs is not easily available
- Modeling overall CPI as a function of categorical CPI(and vice versa) offers the ability to understand specific trends
- Units: Percent change from the base period

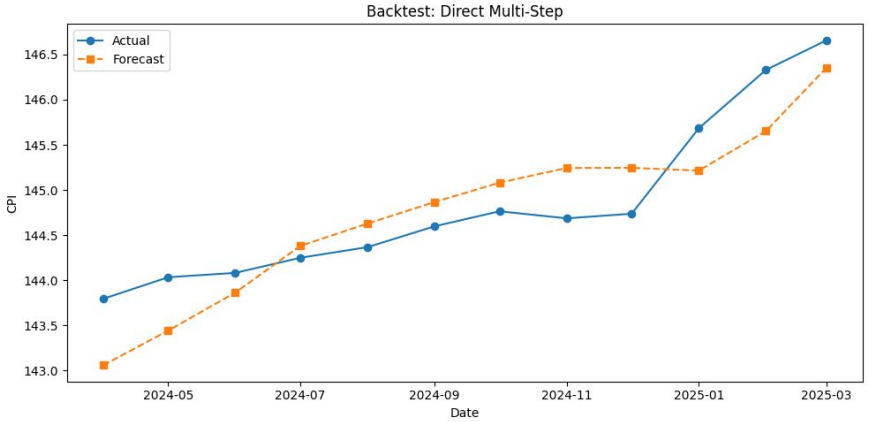


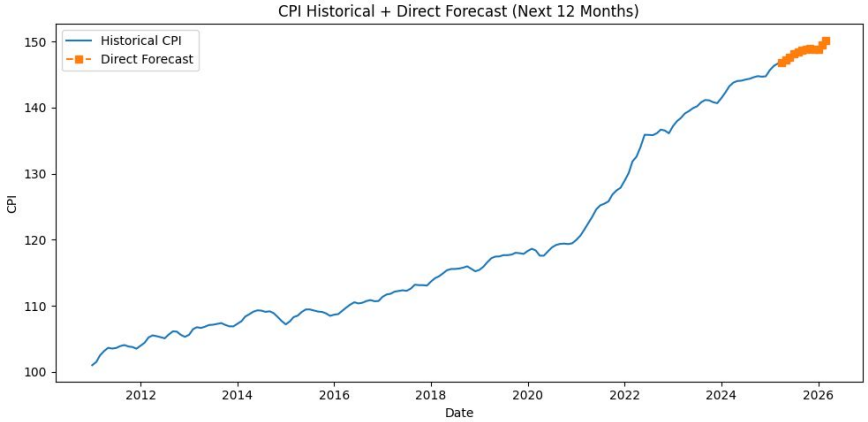
CPI Calculation & Motivation

- Prices of “baskets”(categories) of goods are used to calculate category CPI
- Category CPI’s are weighted and used to calculate overall CPI per country
  - Weights change every 2 years
  - Data used to calculate category CPIs is not easily available
- Modeling overall CPI as a function of categorical CPI(and vice versa) offers the ability to understand specific trends
- Units: Percent change from the base period

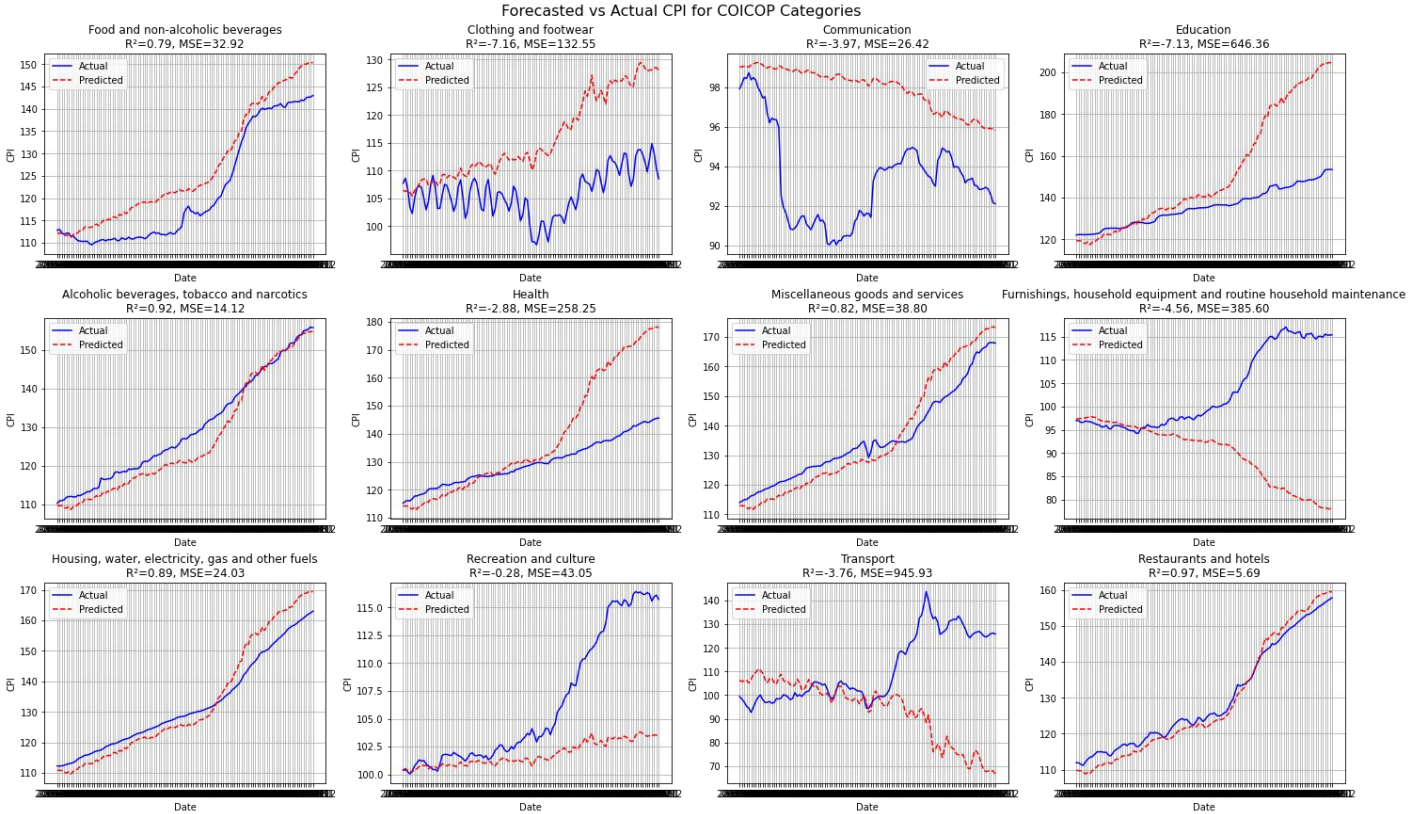
Category
Food and Non-alcoholic Beverages
Clothing and Footwear
Communication
Education
Alcoholic Beverages, Tobacco and Narcotics
Health
Miscellaneous Goods and Services
Furnishings, Household Equipment and Routine Household Maintenance
Housing, Water, Electricity, Gas and Other Fuels
Recreation and culture
Transport
Restaurants and Hotels

Linear Regression





Linear Regression - All Categories vs Overall CPI



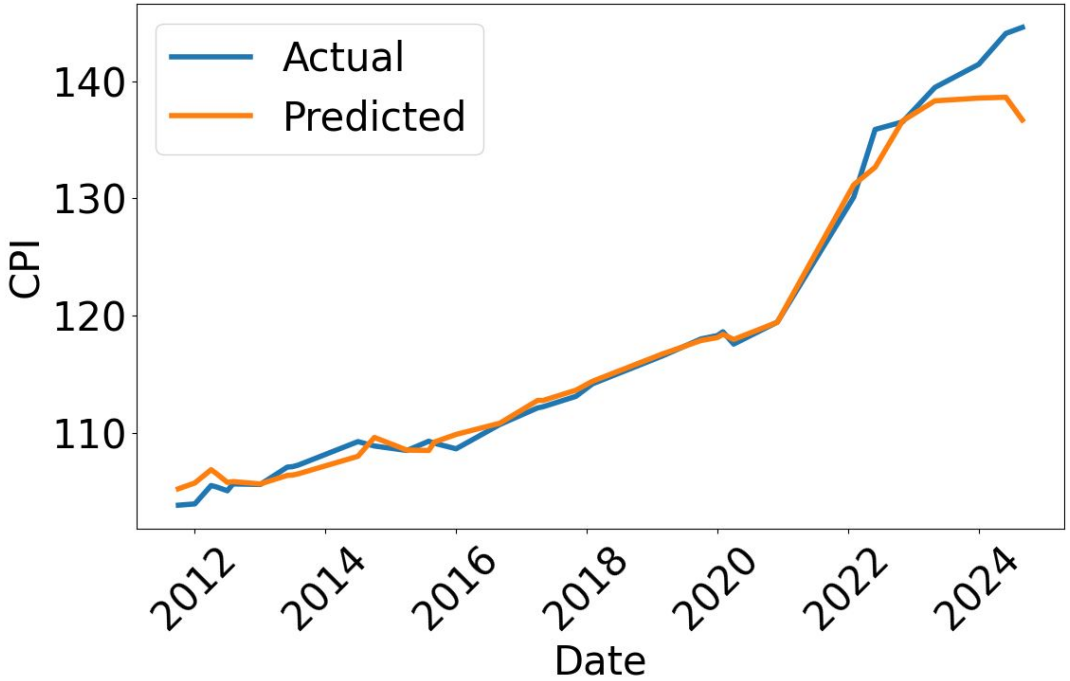
SVM-Based CPI Forecasting: What is SVM?

- Supervised learning method for both classification and regression
- Regression variant called Support Vector Regression
- Uses kernels (e.g., RBF) to model nonlinear relationships
- Controls complexity to prevent overfitting

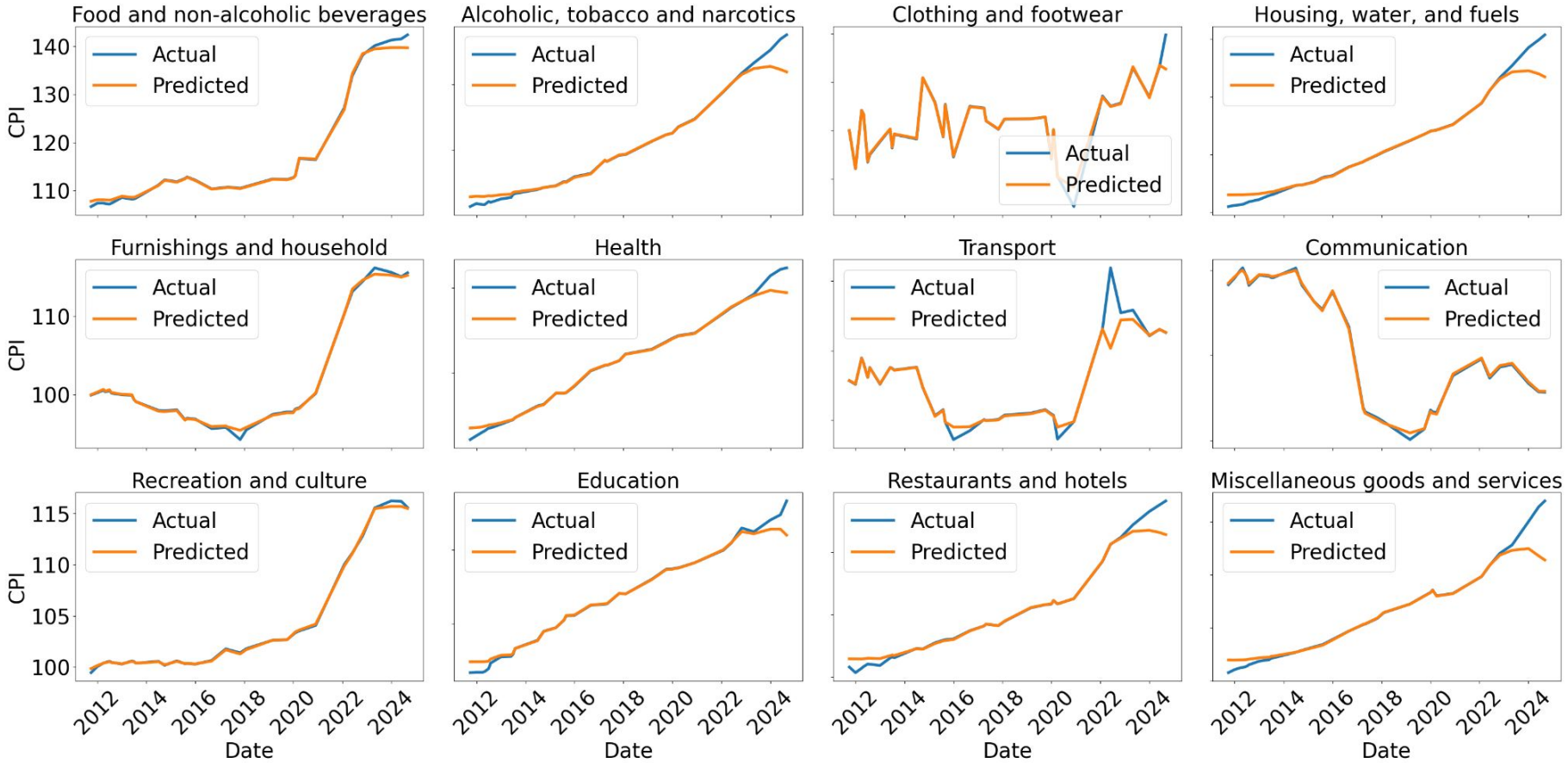
SVM-Based CPI Forecasting: Why SVR?

- Captures complex, nonlinear interactions among economic indicators
- Kernel reveals hidden patterns while avoiding excess complexity
- Tracks smooth trends and sudden jumps in CPI data
- Robust against noisy and correlated input features
- Balances flexibility and stability for reliable forecasts

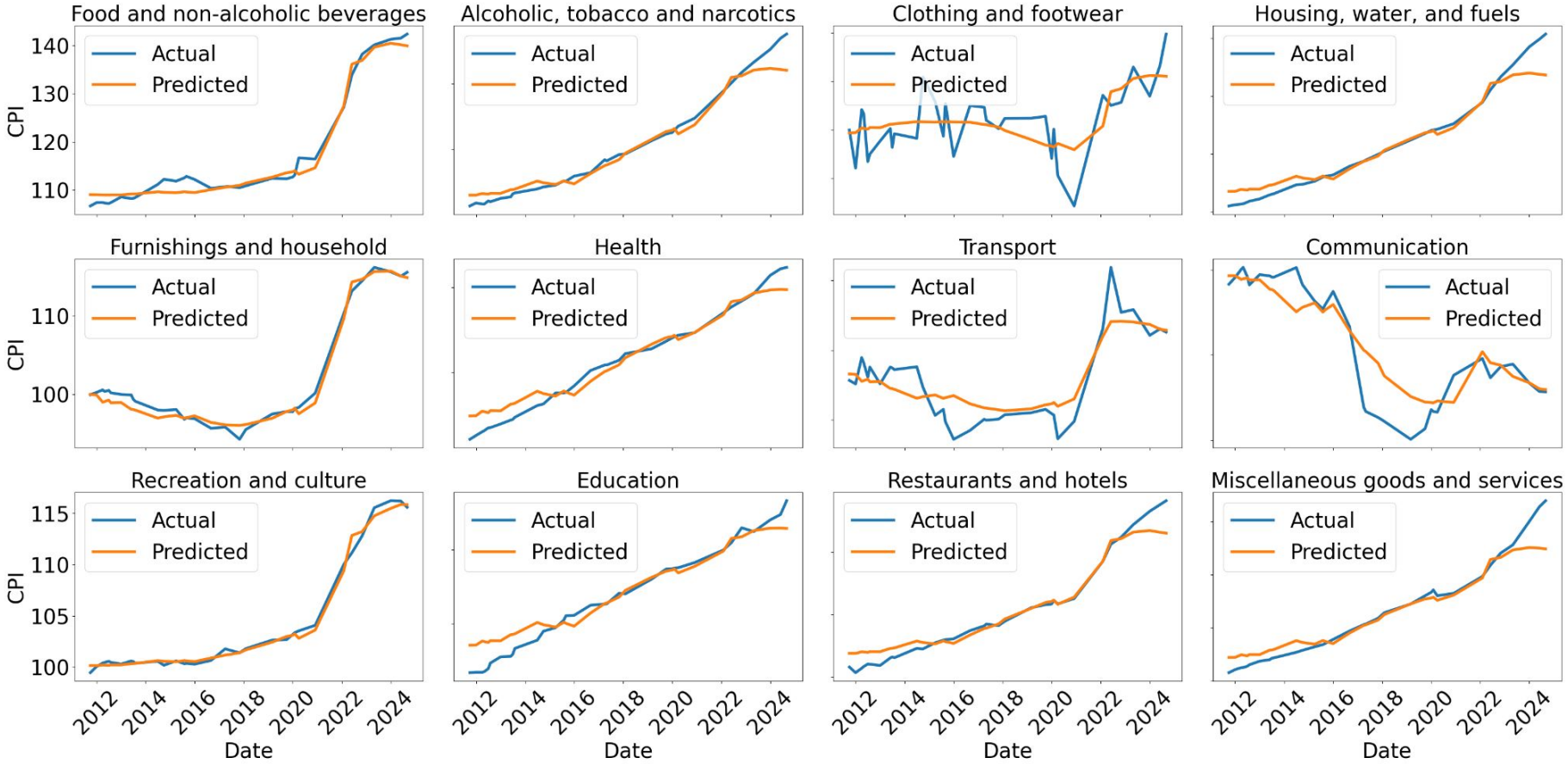
Predict Overall CPI using All Categories



Predict One Category Using Its Data



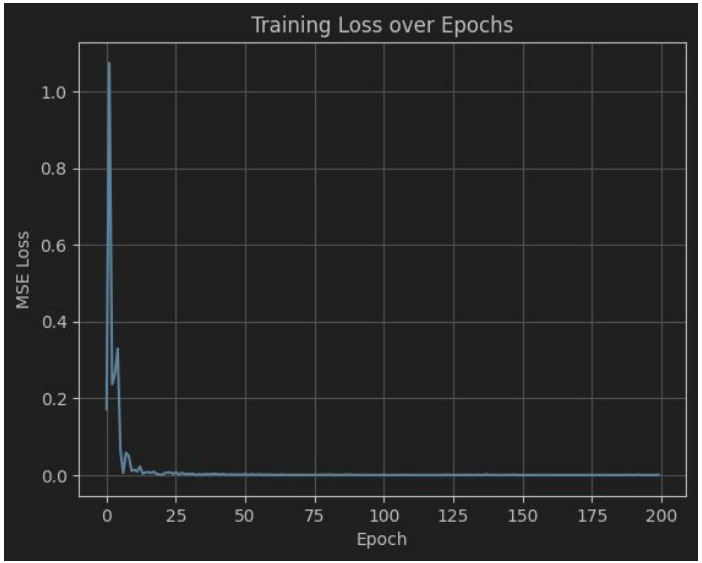
Predict One Category Using Overall CPI

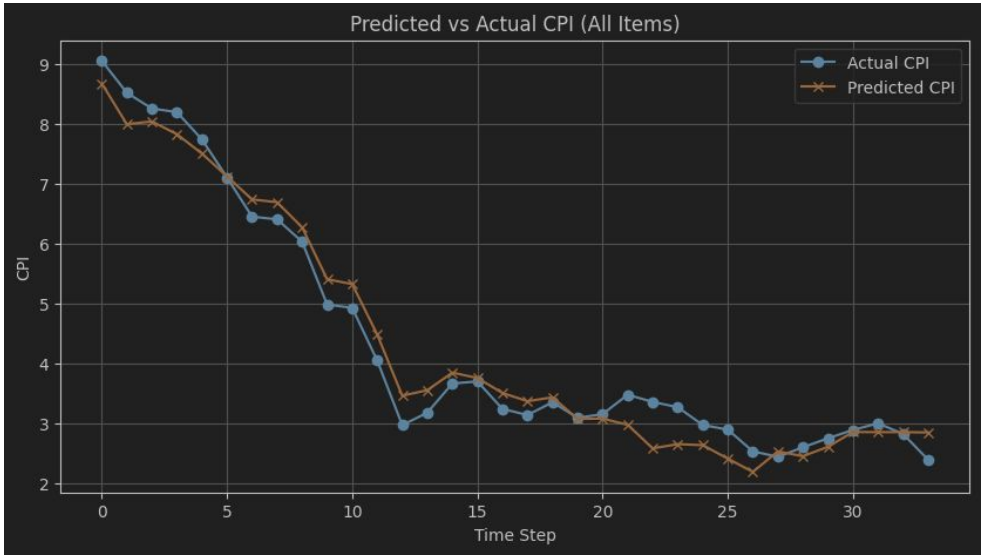


Approach: Neural Networks to Forecast CPI

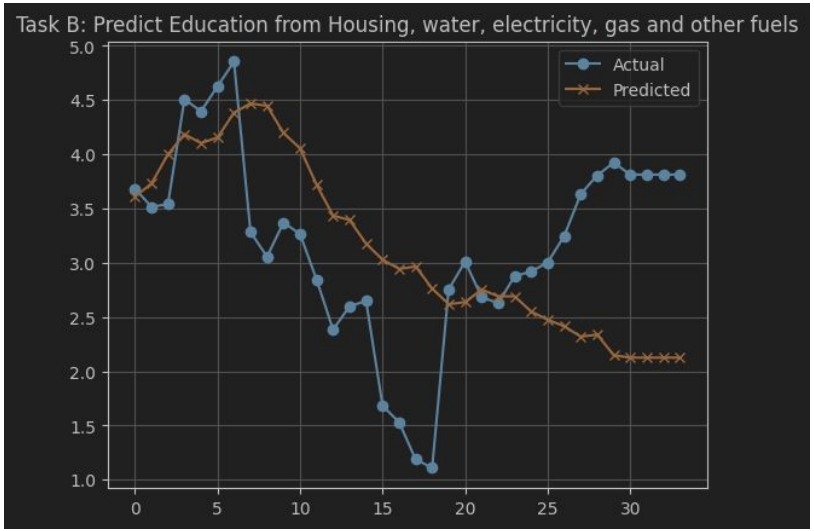
- Predict overall CPI using all other categories
- Predict one category from another
- LSTM: Time series forecasting using past 12 months of overall CPI
- Multivariate LSTM Forecasting

Predict overall CPI using all other categories

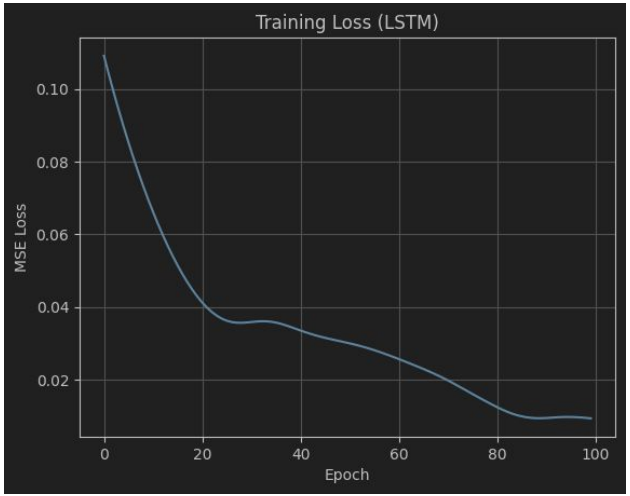


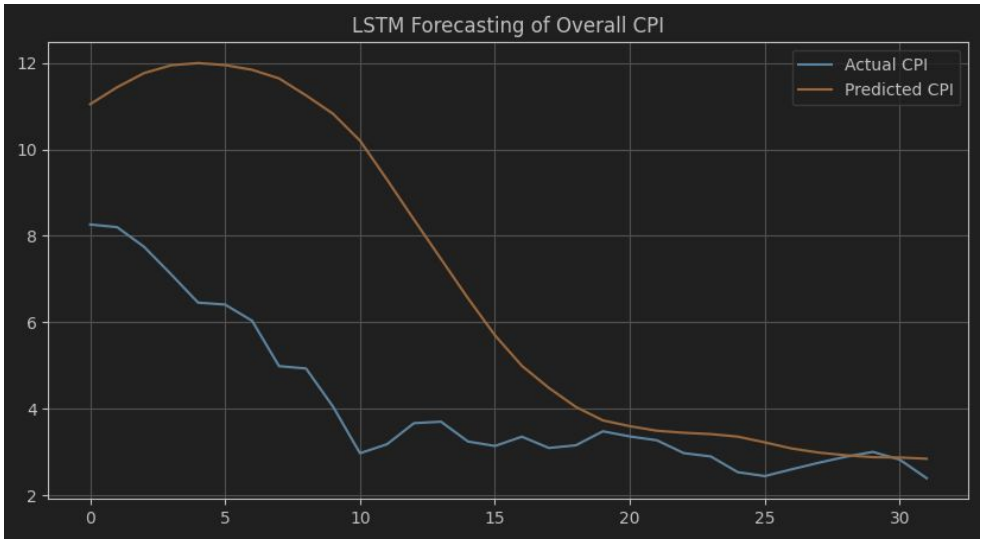


Predict one category from another

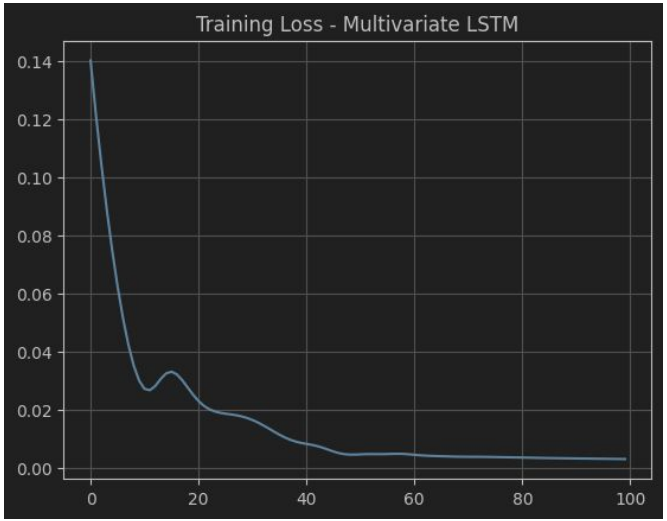


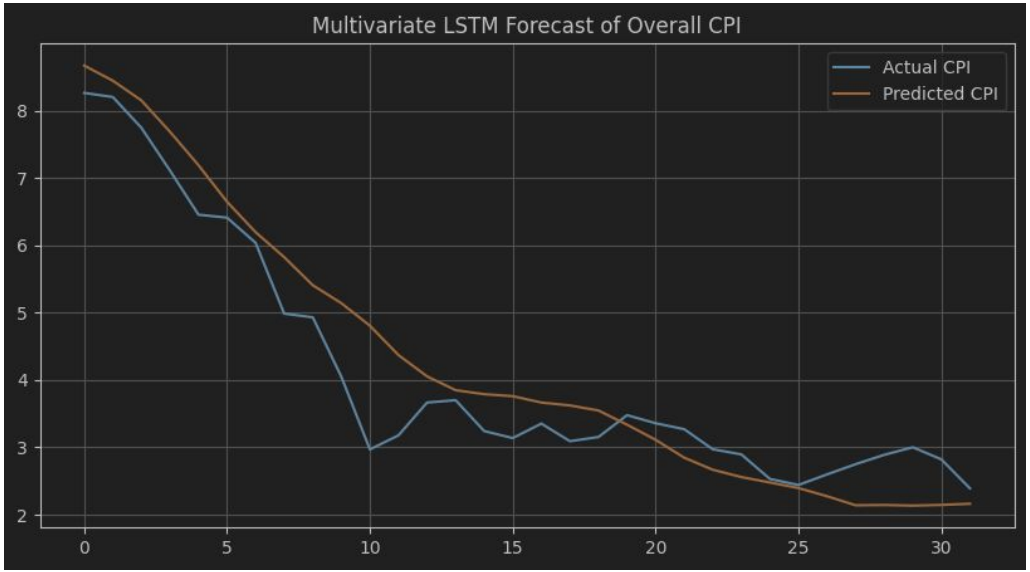
LSTM: Time series forecasting using past 12 months of overall CPI





Multivariate LSTM Forecasting





Conclusion

- SVR shows best performance on both smooth and seasonal CPI
- Accuracy varies highly by category, depending on month to month volatility
- Key drivers of CPI according to Linear Regression: Housing, water, electricity, gas and other fuels, Transport, and Food and non-alcoholic beverages
- Future work: Exploring more autoregressive models, research into models that work better for short term vs long term prediction

A Benchmark for Tabular Data Synthesis

Yuntao Du, Te Guo, Zilin Shen, Hairong Ying

PURDUE
UNIVERSITY

Outline

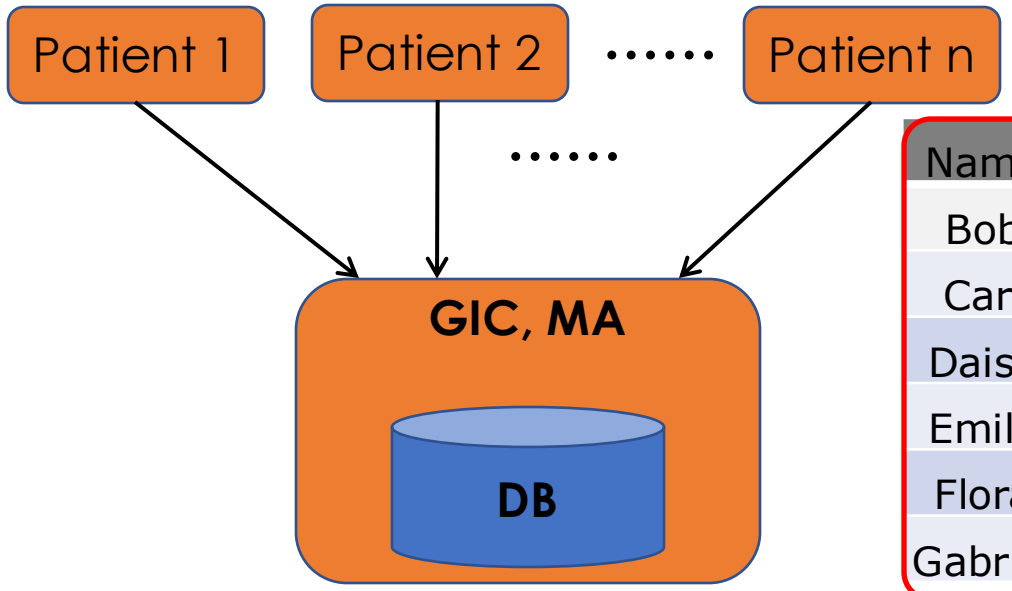
1. Problem Statement
2. Methodology & Evaluation Metrics
3. Experiments
4. Observation & Discussion
5. Future Work

1.Problem Statement

Motivation: Why Synthetic Tabular Data

Group Insurance Commissions (GIC, Massachusetts)

- Collected patient data for ~135,000 state employees.
- Gave to researchers and sold to industry.
- Names are removed.
- Medical record of the state governor is identified.



Privacy breach!

Name	Age	Sex	Zip code	Disease
Bob	69	M	47906	Cancer
Carl	65	M	47907	Cancer
Daisy	52	F	47902	Flu
Emily	43	F	46204	Gastritis
Flora	42	F	46208	Hepatitis
Gabriel	47	F	46203	Bronchitis

Motivation: Why Synthetic Tabular Data

Netflix Movie Rating Data

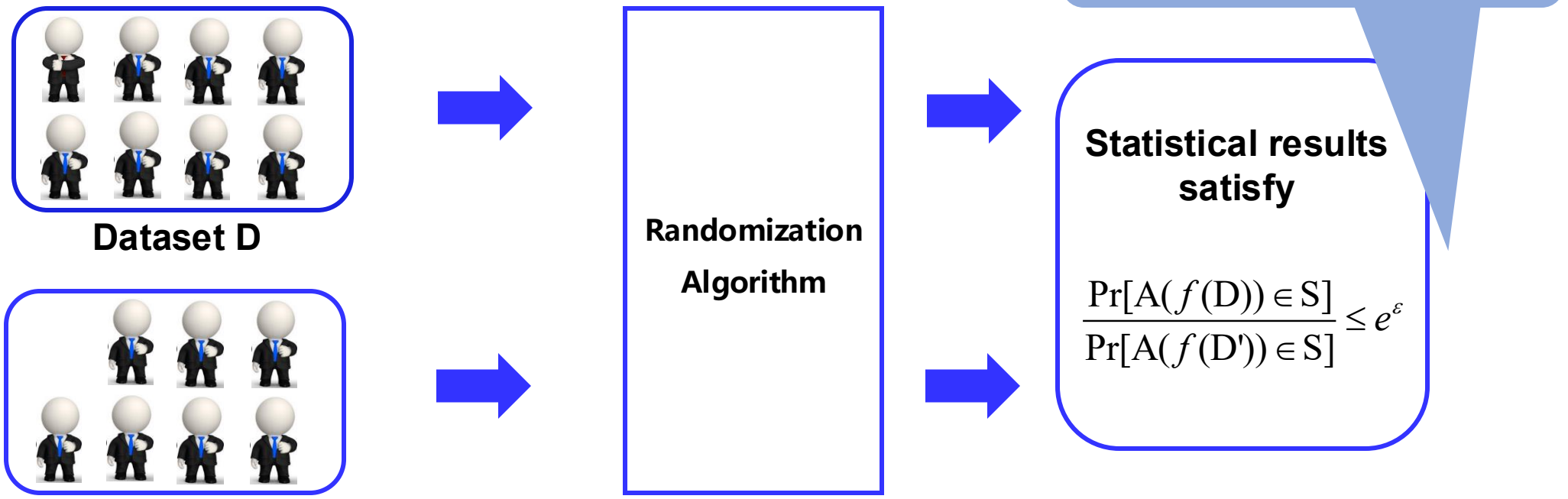
- Netflix released anonymized movie rating data for its Netflix challenge
  - With date and value of movie ratings
- Knowing 6-8 approximate movie ratings and dates is able to uniquely identify a record with over 90% probability
  - Correlating with a set of 50 users from imdb.com yields two records
- Netflix settled a class action lawsuit and canceled second phase of the challenge

Privacy breach!

Motivation: Why Synthetic Tabular Data

Differential Privacy

- Powerful tool for protecting privacy
- Provable privacy guarantee



Not perfect, utility loss!

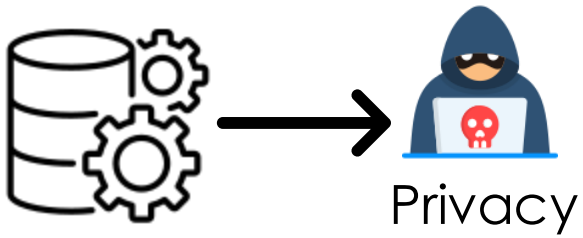
Motivation: Why Synthetic Tabular Data

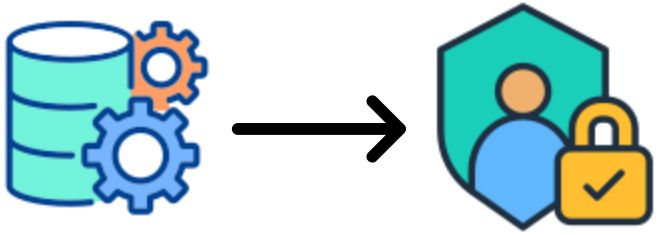
Can we just publish synthetic data?

- Yes, if satisfies differential privacy
- Unfortunately, many synthesis algorithms does not incorporate with differential privacy due to utility loss
  - Empirical privacy evaluation is necessary!

Motivation: Why Synthetic Tabular Data

- **Why Synthetic Tabular Data**
  - Enables **privacy-preserving data sharing**
  - Supports **data-driven decision making** across science, industry and policy
  - Avoids legal and ethical issues tied to using real data
- **Key Use Case:**
  - Share representative datasets **without compromising privacy**





Challenges in Evaluation

- **Rise of Tabular Data Synthesizers**
  - **Statistical** synthesizers: based on low-order marginals
  - **Deep Generative** synthesizers: learn and sample from real data distributions
    - GANs, VAEs, diffusion models, LLMs

❗ **But the problem is…**

❌ No **standard framework** for evaluating synthesizers

❌ **Fidelity—privacy—utility trade-offs** are unclear

❌ Evaluation methods are **inconsistent**, making fair comparison difficult

Project Goal

- **What is proposed**
  - A **benchmark** framework for evaluating tabular data synthesizers.
- **Evaluate across three key dimensions**
  - **Fidelity**—Does the synthetic data resemble real data?
  - **Privacy**—Are individuals in the dataset protected?
  - **Utility**—Is the data useful for ML models and queries?
- **Outcome**
  - Uncovers **strengths & weaknesses** of current synthesizers
  - Highlights **privacy risks**, **utility limitations**, and **evaluation gaps**

2.Methodology & Evaluation Metrics

Overview of Evaluation Framework

- Goal of Evaluation
  - A unified benchmarking framework to compare statistical and deep generative models

To assess tabular data synthesizers across three key aspects

Evaluation Dimension	Key Question	Metric
Fidelity	How realistic is the synthetic data?	Wasserstein Distance
Privacy	Does it protect individuals in the training data?	Membership Disclosure Score (MDS)
Utility	Is the data useful for downstream tasks (ML, queries)?	MLA, Query Error

Fidelity Evaluation

- Metric: Wasserstein Distance
  - Measures how close synthetic data distribution is to real data
  - Works on **numerical, categorical, or mixed attributes**
  - Evaluates on **marginal distributions** (1-way, 2-way, etc.)
- Definition:
  - Compute average Wasserstein distance across selected marginals
  - Lower score → **better fidelity**

$$\text{Fidelity(A)} \triangleq \frac{1}{|V|} \sum_{v \in V} \mathcal{W}(f(v, D), f(v, S))$$

- **Why Wasserstein?**
  - Handles mixed data types (numerical & categorical)
  - Captures **structural differences** between real & synthetic distributions

Privacy Evaluation

- Metric: Membership Disclosure Score (MDS)
  - Measures risk of inferring whether a real record was used in training
- Idea
  - For each record, measure how adding/removing it affects the output of the synthesizer
  - Uses shadow models trained on slightly different datasets to estimate this
- **MDS = max change in synthetic data closeness with and without a record**
  - Lower MDS → **better privacy protection**
  - More robust than syntactic comparisons (e.g., simple similarity)

$$\mathrm{MDS}(\mathrm{A}) \triangleq \max_{x \in D} \Big| \underbrace{\mathbb{E}_{H \subset D, S \sim O_{H \cup \{x\}}}[\mathrm{dist}(x, S)]}_{\textit{closeness of } x \textit{ when trained with } x} - \underbrace{\mathbb{E}_{H \subset D \setminus x, S' \sim O_H}[\mathrm{dist}(x, S')]}_{\textit{closeness of } x \textit{ when } \textbf{not} \textit{ trained with } x} \Big|,$$

Utility Evaluation

- Machine Learning Affinity (MLA)
  - Measures **performance drop** of ML models trained on synthetic vs real data
  - Lower MLA = **better utility**
- Query Error
  - Evaluates accuracy for **range and point queries**
  - Combines random categorical value queries and numerical range queries
  - Lower error = **better statistical utility**

$$\mathrm{MLA(A)} \coloneqq \frac{1}{|\mathcal{E}|} \sum_{e \in \mathcal{E}} \left[\frac{acc(e_{D_{train}}, D_{test}) - acc(e_S, D_{test})}{acc(e_{D_{train}}, D_{test})} \right]$$

3. Experiments

Datasets

Used Datasets (6 Real-world Datasets)

Dataset	Type	Task
Adult	Binary Class	Income prediction
Shoppers	Binary Class	Online purchase intent
Phishing	Binary Class	Phishing website detection
Magic	Binary Class	Gamma telescope simulation
Faults	Multiclass	Steel plate fault detection
Bean	Multiclass	Dray bean classification

Split: 80% training, 20% testing; 20% of training for validation

Synthesizers

- Synthesizers Evaluated
  - Heuristically Private (HP):
    - **CTGAN** – GAN-based, widely used
    - **TVAE** – Variational autoencoder with mode-specific normalization
    - **TabDDPM** – Diffusion-based model for tabular data
    - **REaLTabFormer** – LLM-based method using GPT-2
  - Differentially Private (DP):
    - **MST** – DP graphical model, NIST competition winner
    - **PrivSyn** – Non-parametric DP synthesizer with iterative updates
    - **PATE-GAN** – GAN + teacher aggregation for DP
    - **TableDiffusion** – Diffusion model with DP-SGD

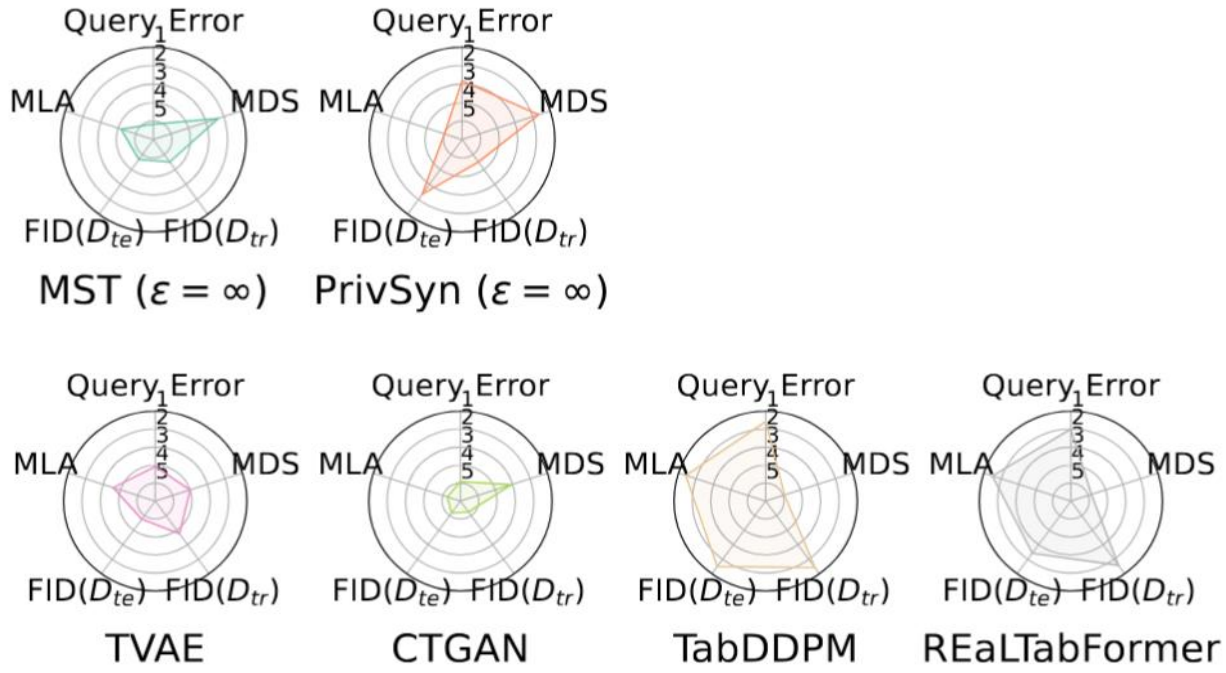
Baseline

We introduce additional baselines to better understand the performance of synthesizers

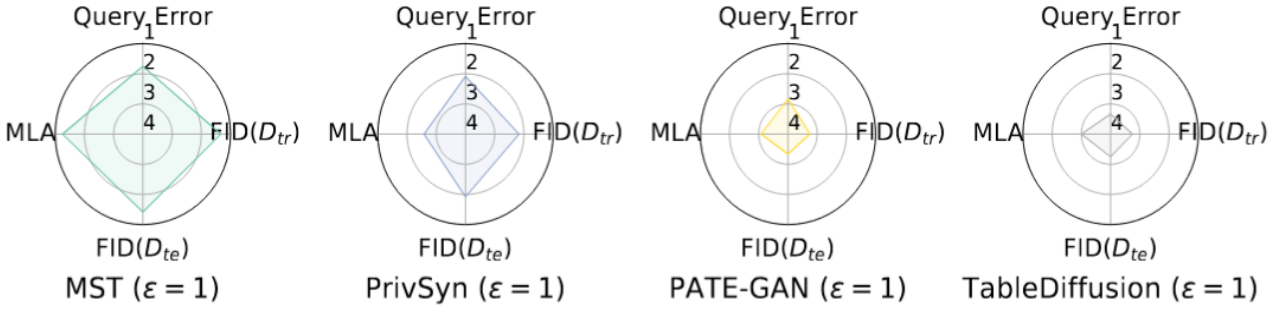
Baseline	Description	Purpose
HALF	Two random splits of real data	Upper bound for **fidelity**
HISTOGRAM	Uses one-way marginals for synthesis	Lower bound for **fidelity**
SELF	Directly uses real data as synthetic data	Lower bound for **privacy**

Fidelity Results

- **Fidelity (Wasserstein Distance)**
  - **Best HP Synthesizers**:
    - TabDDPM and REaLTabFormer achieve near **upper-bound fidelity**
  - **Best DP Synthesizers**:
    - MST and PrivSyn outperform deep models under ε=1



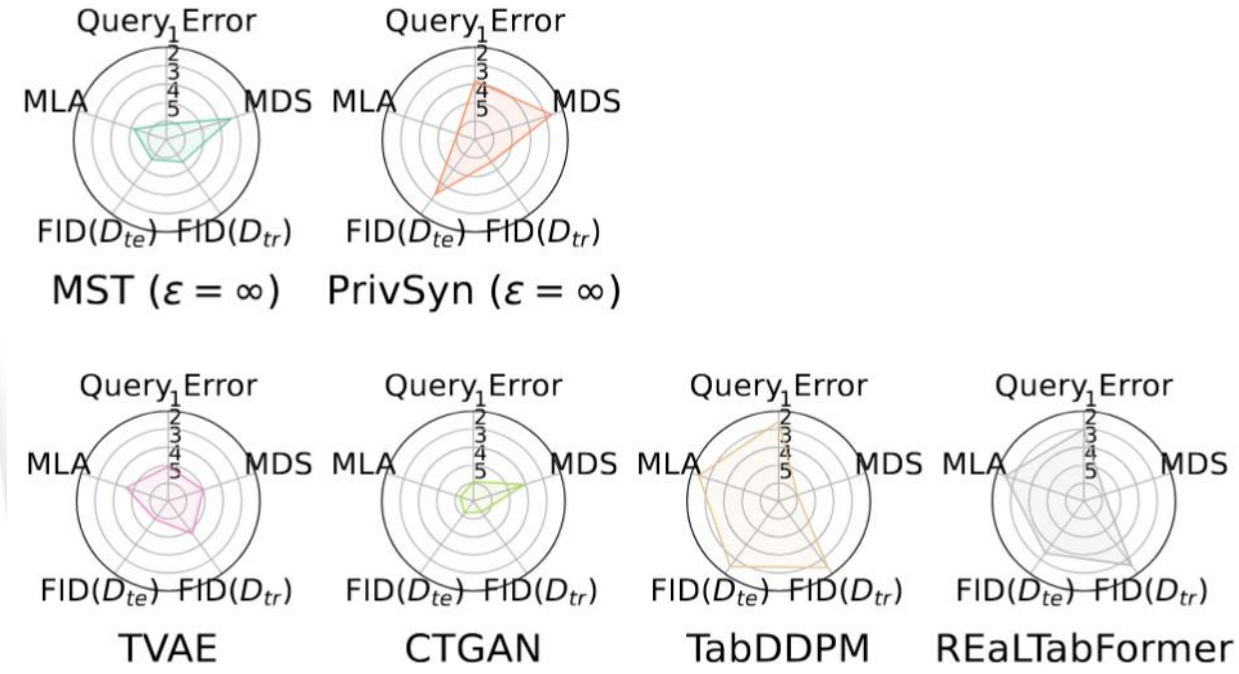
HP synthesizers



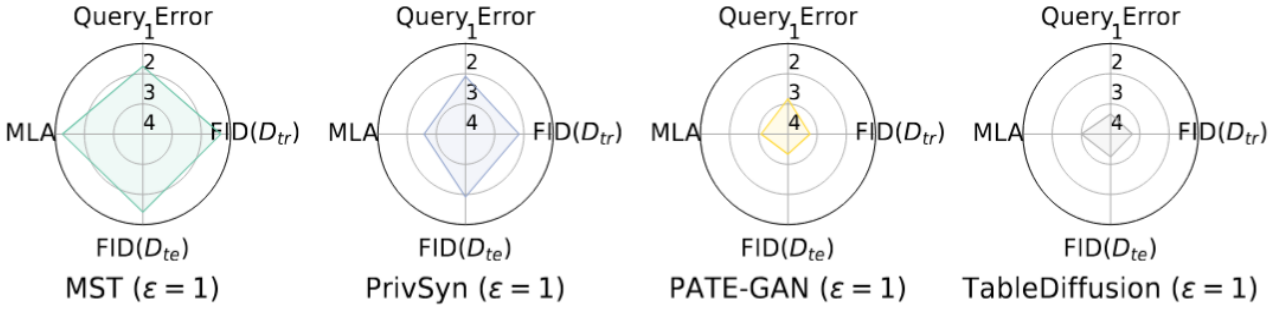
DP synthesizers

Privacy Results

- Privacy (Membership Disclosure Score - MDS)
  - **Statistical methods (MST, PrivSyn)** show strong protection
  - **CTGAN**: low-quality output, but strong privacy by default
  - **TabDDPM**: high utility but **high disclosure risk**



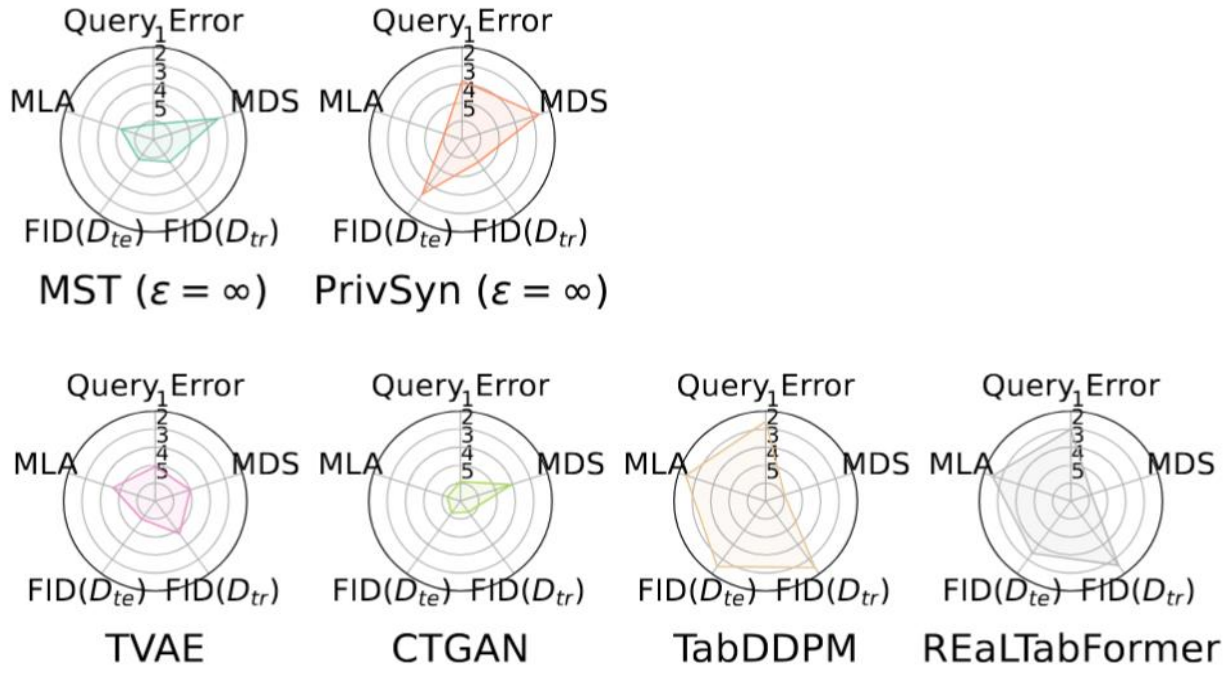
HP synthesizers



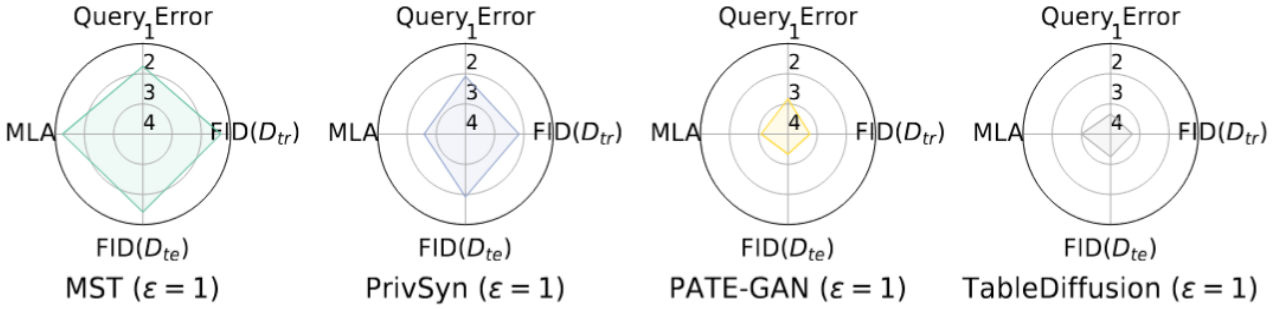
DP synthesizers

Utility Results

- Utility – Machine Learning Affinity (MLA)
  - **Best in HP**: TabDDPM & REaLTabFormer
  - **Best in DP**: MST remains most robust



HP synthesizers

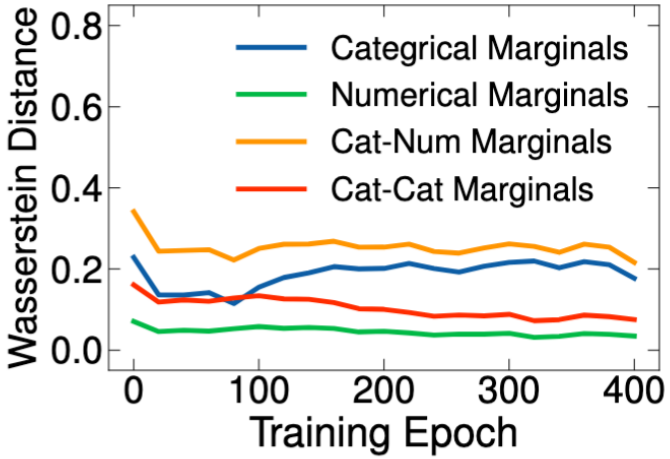


DP synthesizers

4.Observation & Discussion

Observation 1: CTGAN Perform Poorly

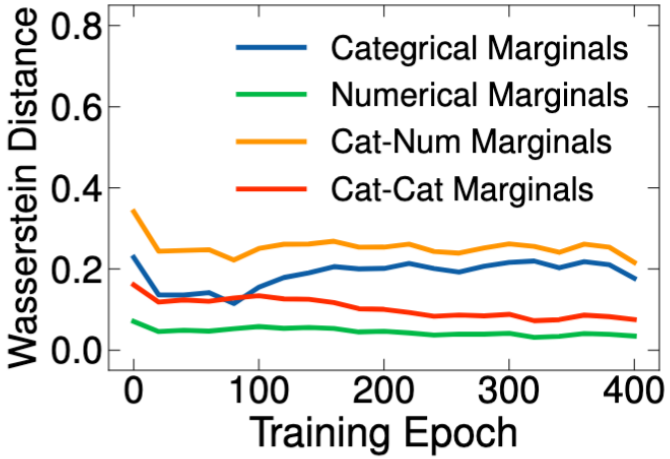
- **Observation:** CTGAN generates the **low-quality** synthetic data among evaluated models.
- **Analysis:**
  - As shown in Figure, fidelity metrics reveal stagnation in both numerical and categorical marginals during training



(a) CTGAN

Observation 1: CTGAN Perform Poorly

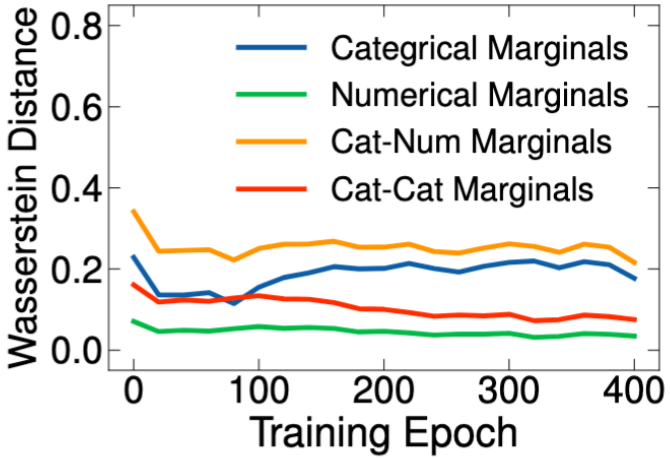
- As shown in Figure, fidelity metrics reveal **stagnation** in both numerical and categorical marginals during training
- **Analysis:**
  - CTGAN uses:
    - Variational Gaussian Mixture Model for numerical features.
    - Conditional sampling for categorical features.



(a) CTGAN

Observation 1: CTGAN Perform Poorly

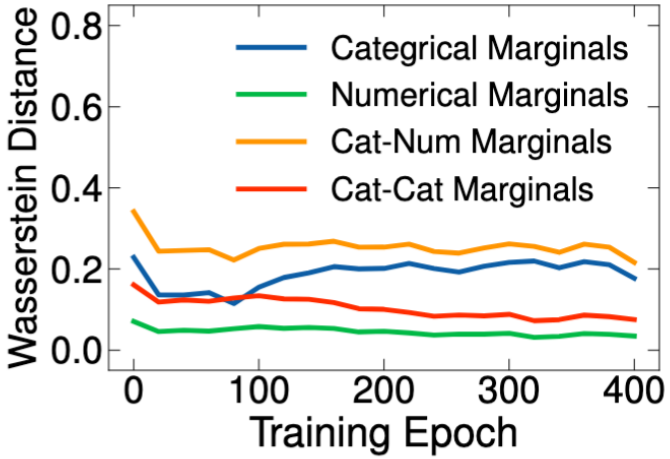
- **Observation:** CTGAN generates the **low-quality** synthetic data among evaluated models.
- **Analysis:**
  - This approach assumes **Gaussian-like distributions**, which doesn't align with the complexity of real-world tabular data.



(a) CTGAN

Observation 1: CTGAN Perform Poorly

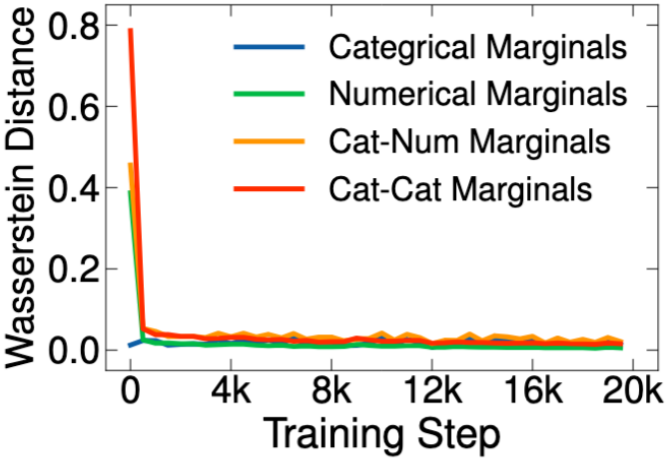
- **Observation:** CTGAN generates the **low-quality** synthetic data among evaluated models.
- **Conclusion:**
  - The performance drop is due to a mismatch between model assumptions and real data characteristics.
  - The poor learning fidelity also explains CTGAN's strong privacy since model fails to capture detailed structure.



(a) CTGAN

Observation 2: TabDDPM Excel in HP Synthesizers

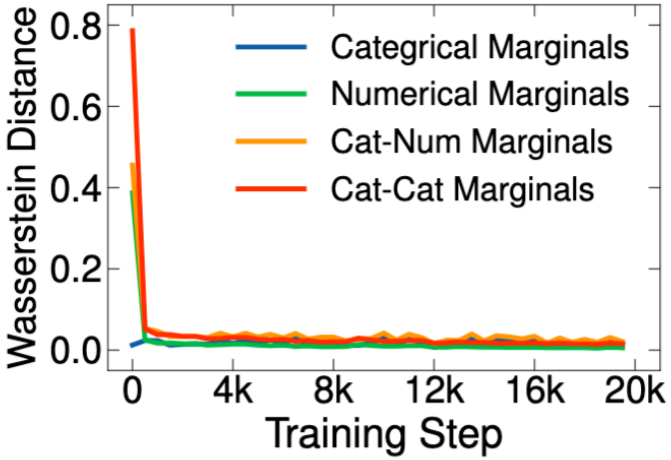
- **Observation:** TabDDPM consistently produces **high-quality synthetic data**.
- **Analysis:**
  - As shown in Figure, Fidelity metrics show **rapid reduction** in Wasserstein distance across all marginal types



(b) TabDDPM

Observation 2: TabDDPM Excel in HP Synthesizers

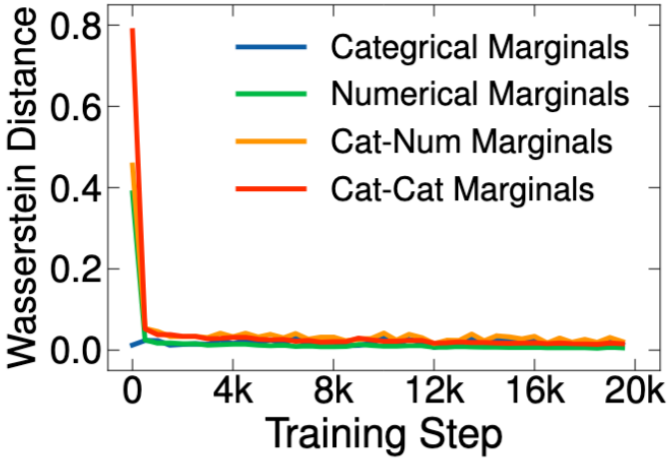
- **Observation:** TabDDPM consistently produces **high-quality synthetic data**.
- **Analysis:**
  - Diffusion models (like TabDDPM) are inherently suited for minimizing Wasserstein distance, unlike other models that minimize KL divergence.



(b) TabDDPM

Observation 2: TabDDPM Excel in HP Synthesizers

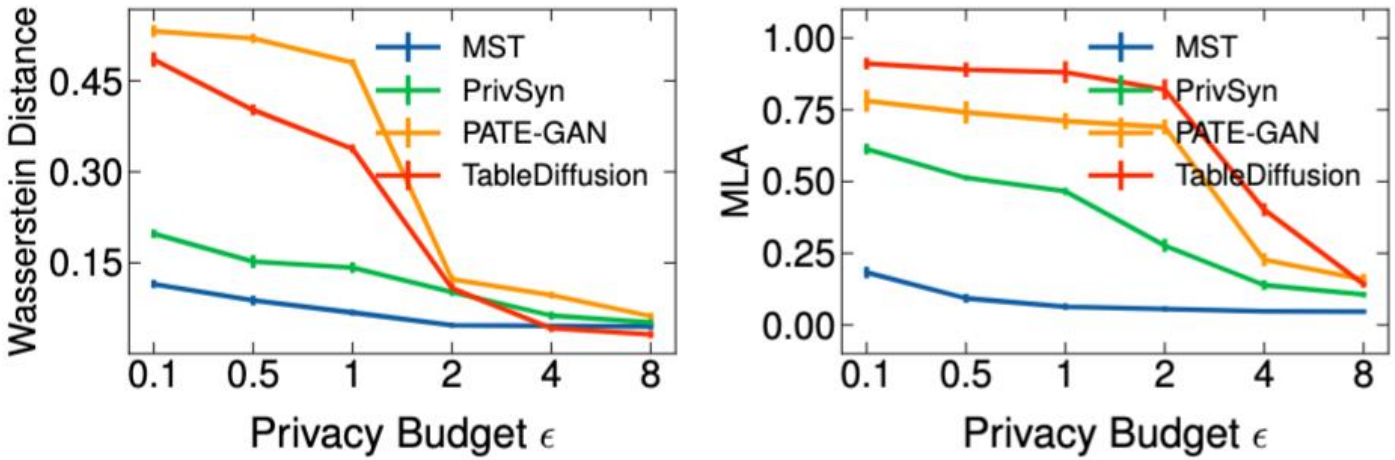
- **Observation:** TabDDPM consistently produces **high-quality synthetic data**.
- **Conclusion**:
  - TabDDPM's architectural advantage makes it more effective at capturing complex tabular distributions.
  - Diffusion models show great promise for tabular data synthesis.



(b) TabDDPM

Observation 3: The Impact of Privacy Budget

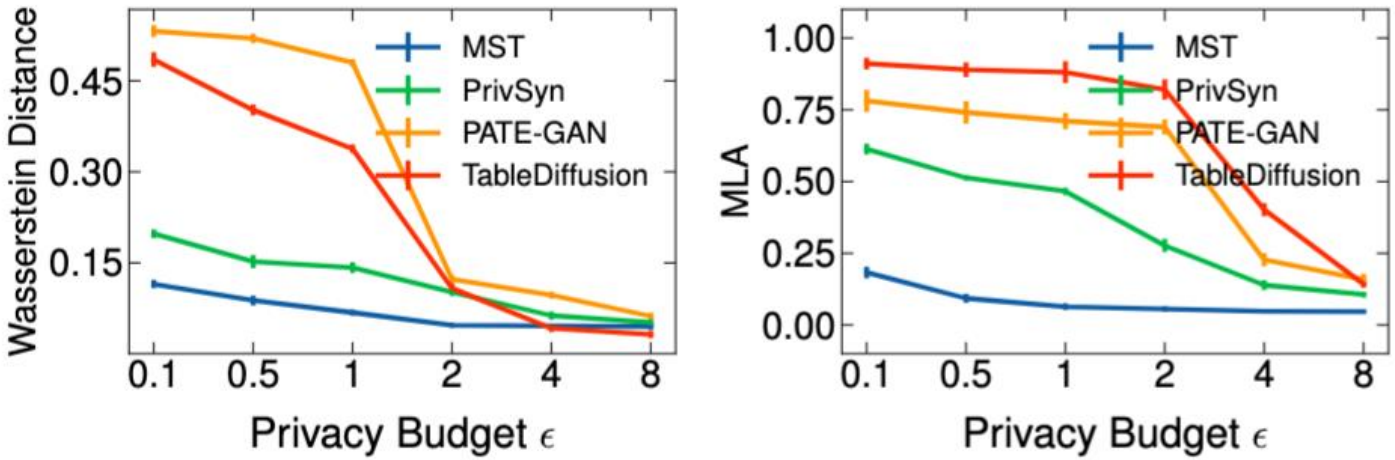
- **Findings**:
  - Statistical methods (e.g., MST) remain robust even at low privacy budgets (e.g., epsilon=0.5).
  - Deep generative models require significantly higher privacy budgets (e.g., epsilon=8) for comparable quality.



Lower score indicates higher fidelity/utility.

Observation 3: The Impact of Privacy Budget

- **Conclusion**:
  - Statistical methods are more privacy-resilient due to their reliance on a limited set of marginals.
  - Deep models suffer under tight privacy constraints.



Lower score indicates higher fidelity/utility.

5.Future Work

Future Work

- **Potential Future Work:**
  - Design privacy-preserving architectures that **retain fidelity under tight budgets**.
  - Explore **hybrid models** that blend statistical robustness with the expressiveness of deep learning.
  - Investigate **adaptive preprocessing** that aligns better with generative assumptions.
  - ……

Thanks,
Q&A

Music Genre Classification with GTZAN Dataset

- CS573 Final Project Presentation -

Hyunwoo Chung
Nicholas Frederick
Rohan Garg
Yaoxu Song

Apr 22, 2025

Outline

GTZAN Dataset Overview

- We use the GTZAN music genre classification dataset available on Kaggle[1].
- The dataset consists of three complementary formats:
  (1) 1,000 thirty-second audio clips across ten genres (100 clips per genre: blues, classical, country, disco, hip-hop, jazz, metal, pop, reggae, rock)
  (2) corresponding mel-spectrogram images
  (3) a CSV file containing 57 extracted audio features for each clip.
- Below is a brief overview of selected audio features:
  - **MFCCs**: Capture the short-term power spectrum on a perceptual mel scale, encoding timbral texture.
  - **RMS Energy**: Root-mean-square of the waveform amplitude per frame, a measure of loudness.
  - **Spectral Centroid**: The "center of mass" of the spectrum, indicating brightness.
  - **Spectral Rolloff**: The frequency below which a fixed percentage (e.g. 85%) of spectral energy lies, summarizing spectral shape.
  - **Tempo**: Estimated beats-per-minute via onset detection and autocorrelation.

[1] `https://www.kaggle.com/datasets/andradaolteanu/gtzan-dataset-music-genre-classification`

Evaluation

- We use accuracy and macro-F1 as our evaluation metrics.
- Accuracy provides a straightforward measure of overall correctness but does not distinguish between false positives and false negatives.
- In contrast, the F1 score–the harmonic mean of precision and recall–balances these error types, offering a more nuanced view of performance than accuracy. Macro-F1 is the average of each genre's one-vs-all F1.
- Together, these metrics capture both the exact-match rate (accuracy) and the balance of per-genre performance (macro-F1).
- $\uparrow$ accuracy, $\downarrow$ macro-F1 $\rightarrow$ overall good performance, ignorance of certain classes.
- $\downarrow$ accuracy, $\uparrow$ macro-F1 $\rightarrow$ strong one-vs-all separation but difficulty in selecting the single best genre among ten.

Experimental Protocol

- Each model is evaluated according to the following steps:

1. Randomly split the entire dataset into 70% training and 30% test sets.
2. Train the model on the training set.
3. Evaluate on the test set, computing test accuracy and test macro-F1.
4. Repeat Steps 1–3 for $R = 100$ replications. Let M_i denote the value of a metric in the ith replication. We compute the mean $\bar{M} = \sum_{i=1}^{R} M_i / R$ and the standard error

$$\text{S.E.}(\bar{M}) = \frac{1}{\sqrt{R}} \sqrt{\frac{1}{R-1} \sum_{i=1}^{R} (M_i - \bar{M})^2}$$

for each metric.

Exploratory Data Analysis: Approaches

1. Assess genre separability $\rightarrow$ t-SNE
2. Investigate multicolinearity $\rightarrow$ Correlation plot
3. Explore dimensionality reduction $\rightarrow$ PCA

t-SNE Visualization

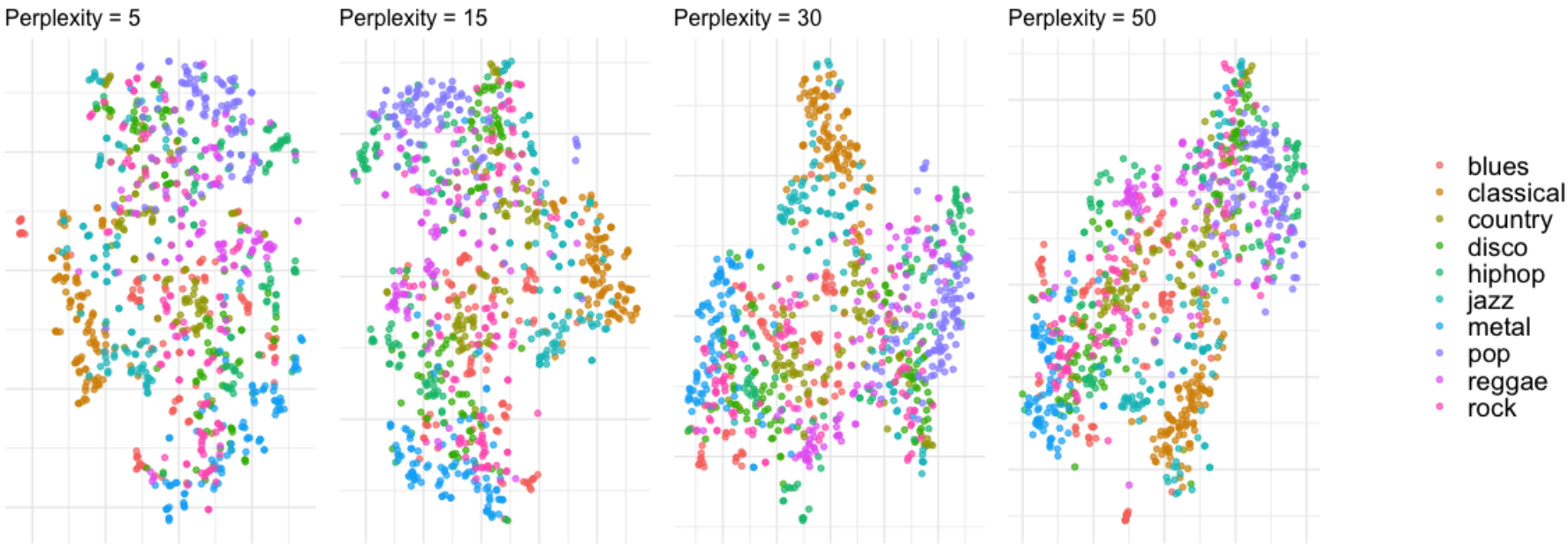


Figure 1: t-SNE of GTZAN audio features for different perplexity values (5, 15, 30, 50).

- We project GTZAN audio features into 2D using t-SNE with four different perplexities (analogous to k in k-NN), to identify distinctive clusters.
- For example, we can observe a clear contrast between classical (yellow) and metal (blue).
- Overall, most genres form distinct clusters, with closely related styles (e.g., disco/pop/reggae) sharing regions.

Correlation Plot

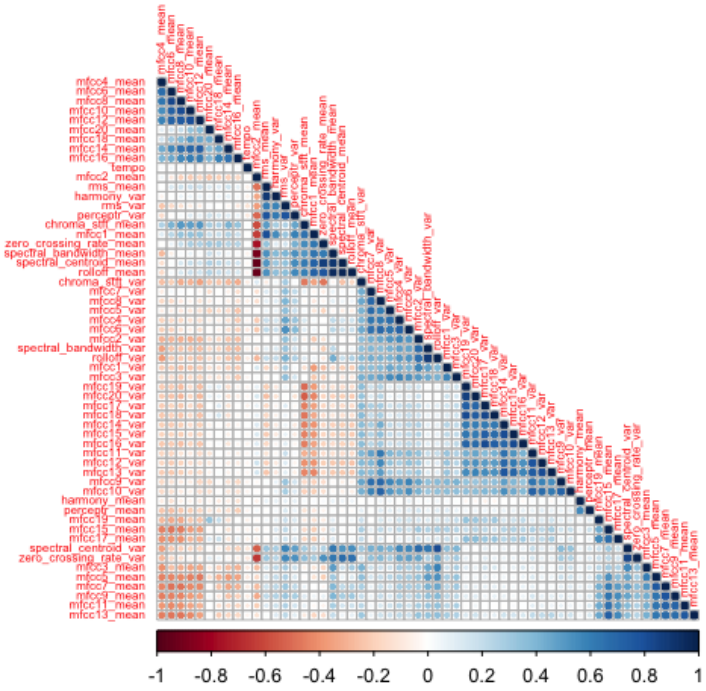


Figure 2: Correlation heatmap of GTZAN audio features.

- We prepare a correlation plot to investigate feature redundancy.
- Some features exhibit near-perfect correlations, and such redundancy suggests dropping or combining features via dimensionality reduction (e.g., PCA).

PCA

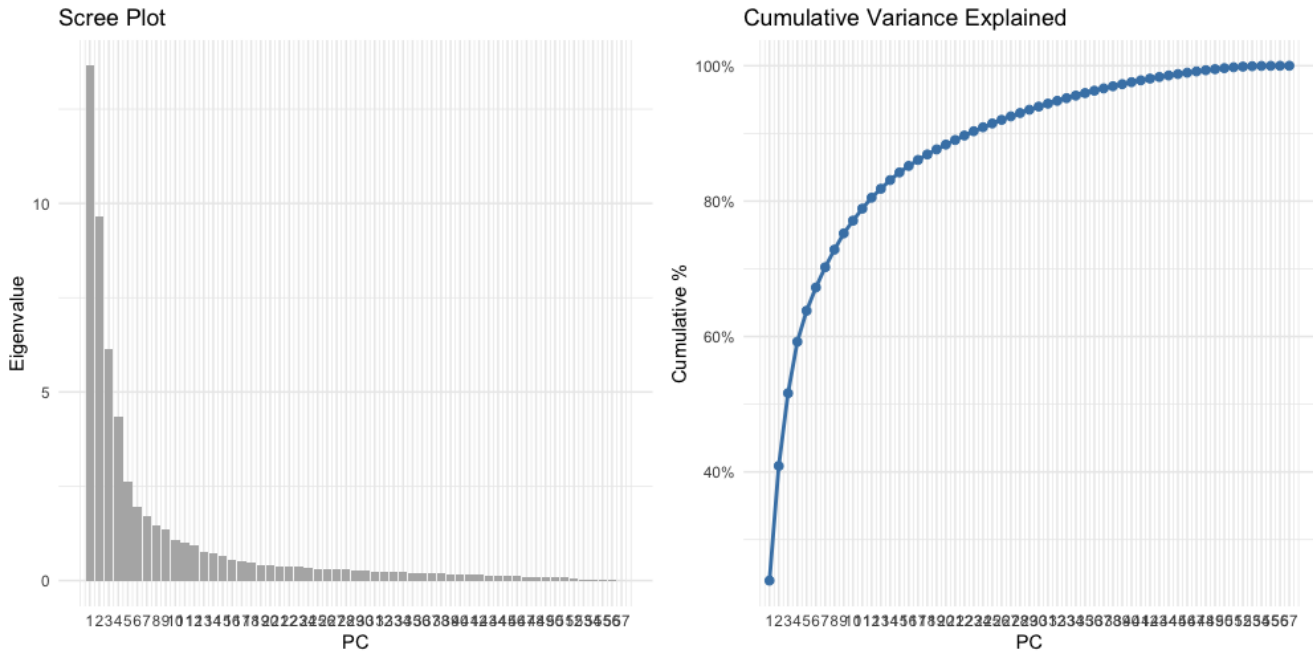


Figure 3: PCA scree and cumulative variance plots for GTZAN audio features.

- We perform PCA to identify the data's intrinsic dimensionality.
- The scree plot (left) shows each PC's eigenvalue (=variance), which falls sharply from PC1 to PC4 and drops below the Kaiser threshold around PC10.
- The cumulative variance plot (right) also suggests retaining 4–10 PCs for downstream modeling, since they together explain 57–78% of total variance.

Multinomial Logistic Regression

- Recall from our t-SNE that many genres form roughly linearly separable clusters.
- Thus, we begin with multinomial logistic regression as a natural baseline.
- It models the class probabilities via a softmax over linear combination of predictors:

$$\mathbb{P}(Y = c|\mathbf{X} = \mathbf{x}) = \frac{\exp(\beta_{0,c} + \mathbf{x}^\top \boldsymbol{\beta}_c)}{1 + \sum_{k=1}^{C-1} \exp(\boldsymbol{\beta}_{0,k} + x^\top \boldsymbol{\beta}_k)}, \quad c = 1, \ldots, C-1$$

and for the reference class C,

$$\mathbb{P}(Y = C|\mathbf{X} = \mathbf{x}) = \frac{1}{1 + \sum_{k=1}^{C-1} \exp(\beta_{0,k} + x^\top \boldsymbol{\beta}_k)}.$$

Challenges & Strategies

1. Unstable convergence: Only 70 training samples per class versus $9 \times (57 + 1) = 522$ parameters, leading to high overfitting risk.
2. Strong multicollinearity: Clear redundancy among features (e.g. adjacent MFCCs, `rolloff_mean` vs. `mfcc2_mean`) in the correlation plot.
3. Excessive feature dimension: PCA shows the first 4–10 PCs capture most of the variability in our data, motivating projection to a lower-dim subspace.

$\Rightarrow$ Need for dimensionality reduction!

- Thus, the following three different approaches considered:
  1. Select a sparse subset (AIC-based forward selection)
  2. Regularize to zero out weak signals (Regularization with LASSO penalty)
  3. Project the original input onto a low-dim subspace (PCA-based)

Results

Model	Accuracy (Mean±S.E.)	Macro-F1 (Mean±S.E.)
Forward Selection	0.614 ± 0.003	$0.957 \pm < 0.001$
Regularization	$\mathbf{0.709 \pm 0.002}$	$\mathbf{0.968 \pm < 0.001}$
PCA-based	0.578 ± 0.003	$0.953 \pm < 0.001$

Table 1: Performance of the three multinomial logistic regression based classification models over 100 randomized 70/30 splits.

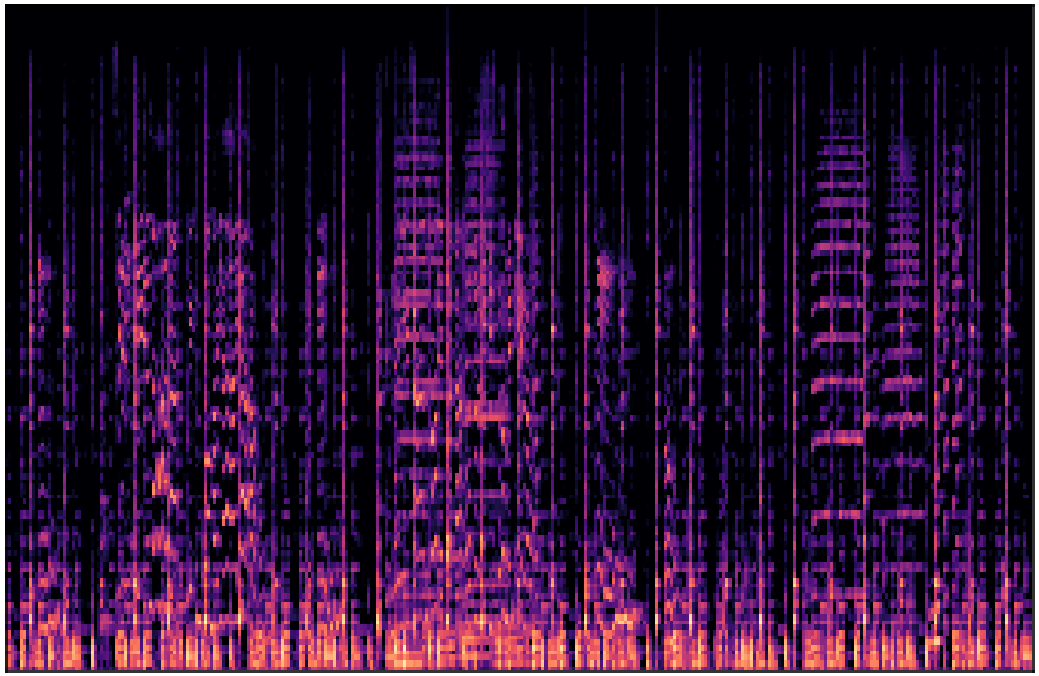
Interpretation and Insights

- Regularization > Forward selection > PCA-based approach.
- Why not forward selection? → relies on a greedy search that may overlook useful combinations of features, limiting its accuracy.
- Why not PCA-based approach? → can reduce dimensionality, but optimizes for total variance rather than class separability. Some components with high variance may carry little genre-discriminative signal.
- All three models share the pattern of moderate accuracy (0.58-0.71) alongside very high macro-F1 (0.95-0.97).
- That is, our models do an excellent job of distinguishing each genre in isolation (high per-class precision and recall), but they are less reliable at the harder multiclass decision task, hence resulting in lower accuracy.

Interpretation and Insights (cont'd)

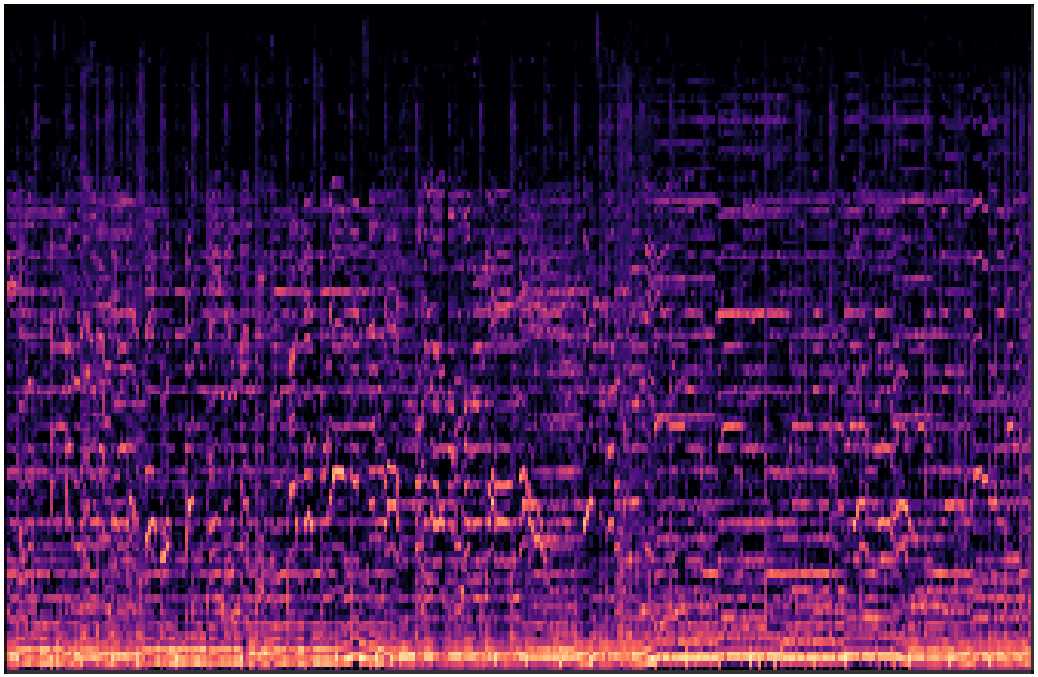
- LASSO consistently retained features tied to four key auditory cues:
  - **Harmonic content**: `harmony_mean`, `perceptr_mean` → tonal strength and perceptual sharpness (high in classical, jazz; low in noise-driven genres).
  - **Spectral shape**: `chroma_stft_mean/var`, `spectral_centroid_var`, `rolloff_mean` → energy distribution (pitch classes, brightness, high-frequency cutoff) distinguishing "bright" vs. "dark" genres.
  - **Dynamics & rhythm**: `rms_var`, `tempo` → loudness fluctuations and beat rate separate punchy styles (hip-hop, rock) from smoother ones (country, pop).
  - **Timbral texture**: select MFCCs (e.g. `mfcc3_mean`, `mfcc12_mean`, `mfcc17_mean`) → capture mid-/high-frequency spectral details that refine distinctions among similar genres.
- Forward selection yields an even sparser model–its greedy, conservative search may have skipped informative feature combinations.
- PCA-based approach is interpretable only at the component level (no direct feature meaning).

Mel Spectrogram Representation: Blues



- A lot of black to represent little to no music, and darker colors for light and soft parts of the song at great height
- Not a lot of bright colors to represent loud and intense music.

Mel Spectrogram Representation: Country



- Bright colors appear more frequently and at greater heights for more intense and ample music.
- Not a lot of black or dark purple colors to represent the soft music we saw in the blues example.

Types of Neural Networks

- We use a few neural networks to try to get the best performance out of our models and to produce the most accurate data testing:
  - FNN(Feedforward Neural Network): Good at one dimensional and simple datasets, fully connected networks, not great on complex data. Goes from Inputs to Hidden Layers to Outputs in terms of layering.
  - CNN(Convolutional Neural Network): Works best with image recognition, better with complex datasets, analyzes for features (patterns, edges, etc.) and variables to use for convolutions. Requires a large dataset to perform well.
  - MLP(Multi-Layer Perceptron)-Fully connected neural network, has a similiar layer structure as FNN, relies on activation functions like ReLu, good with structured data but bad with images
- We want to cover all of our bases when trying to produce the most accurate results. While CNN is the best, FNN or MLP may find something that it missed and thus it can't hurt to see if they have more accurate labeling.

Interpretation and Insights

- FNN $\approx$ CNN $\approx$ MLP
- Linear models presents poor results with low accuracy (0.161-0.300) and Macro F1 scores (0.130-0.295)
- We apply changes with the epochs, models themselves, test and training data sizes, etc. and no noticeable differences.
- Largely inconsistent, with the highest result getting around .500 and the lowest getting around .120, a .380 difference when comparing the two.
- Neural networks are a surprisingly poor method to use when trying to predict the music genre from a Mel Spectrogram Representation.

The Support Vector Machine

- Support Vector Machine (SVM) is a supervised machine learning algorithm used for classification and regression tasks.
- Core concept: Find an optimal hyperplane that maximally separates different classes in the feature space
- Key components of SVM:
  - **Hyperplane**: The decision boundary that separates data points of different classes.
  - **Support Vectors**: The data points closest to the hyperplane that influence its position and orientation.
  - **Margin**: The distance between the hyperplane and the closest data points (support vectors).

The Support Vector Machine (cont'd)

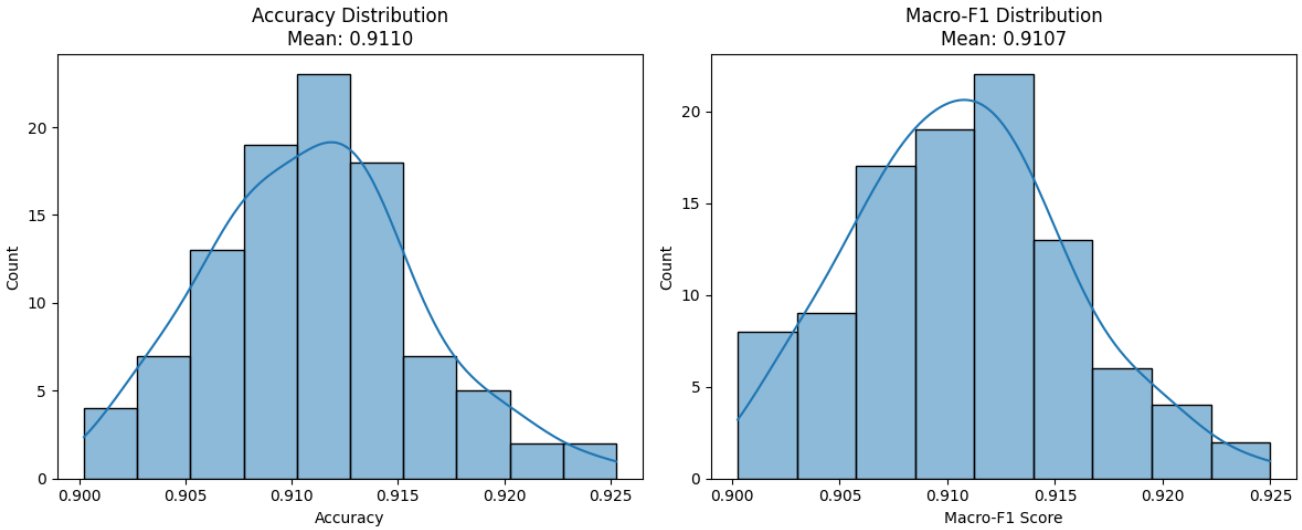
Advantages of SVM:

- Effective in high-dimensional spaces
- Memory efficient as it uses only a subset of training points (support vectors)
- Versatile through different kernel functions
- Robust against overfitting, especially in high-dimensional spaces

Limitations of SVM:

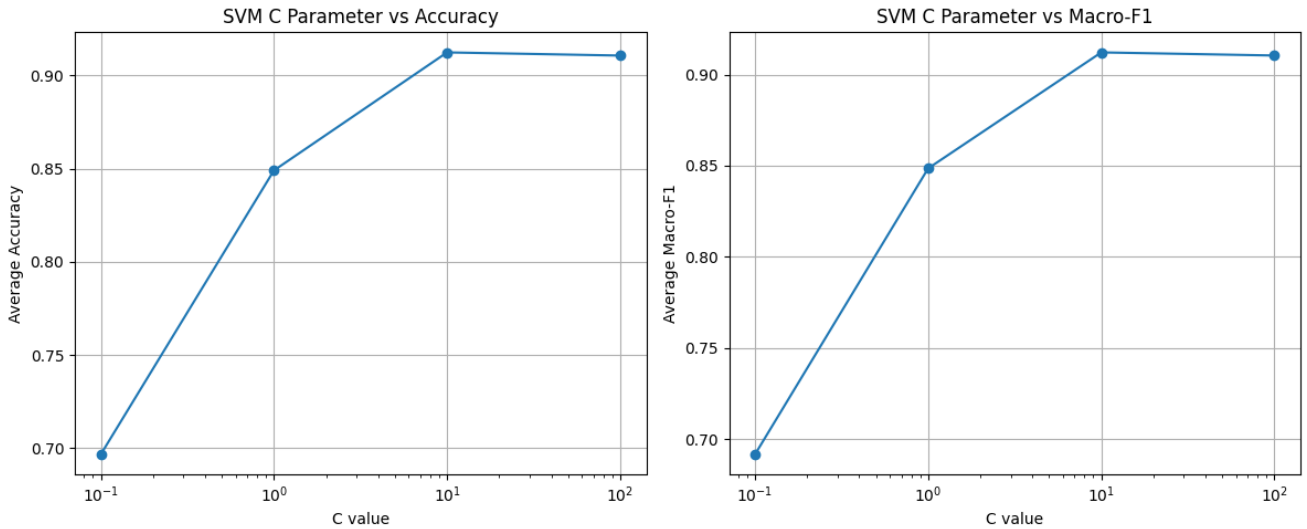
- Computationally intensive for large datasets
- Selecting an appropriate kernel and parameters requires tuning
- Less interpretable than simpler models

SVM: Accuracy and Macro-F1



- We see an average accuracy of 91.1% and an average Macro-F1 score of 91.07%.

SVM: C-Parameter Tuning



- The C parameter[2] plays a crucial role in determining the balance between achieving a low training error and allowing for misclassifications.
- Large Value of parameter $C \rightarrow$ small margin, Small Value of parameter $C \rightarrow$ large margin.
- We see that the best C value is $C = 10$. This gives the best average F1 score of 0.912.

[2] Controls the trade-off between margin width and misclassification penalty:

$$\min_{w,b,\{\xi_i\}} \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n}\xi_i \quad \text{s.t.} \quad y_i\left(w^\top x_i + b\right) \geq 1 - \xi_i,\ \xi_i \geq 0.$$

Leveraging Diverse Audio Features

- Music is a complex signal with rich information encoded in its spectral content, harmonic structure, and temporal evolution. To effectively classify genres, we focus on three complementary feature representations: Mel-spectrograms, Chroma, and Temporal features.
- They provide different perspectives on the same audio signal, offering potentially uncorrelated information that can be leveraged by an ensemble model.

Feature	**Characteristics**
Mel-spectrograms	Spatial patterns (frequency relationships), Temporal evolution (changes over time)
Chroma Features	Spatial patterns (pitch class distribution), Temporal evolution (harmonic changes)
Temporal Features	Sequential data representing dynamic evolution over time (ZCR, Centroid, RMS)

Table 2: Feature Characteristics and Connection to Model Architecture

Ensemble Model: Architecture

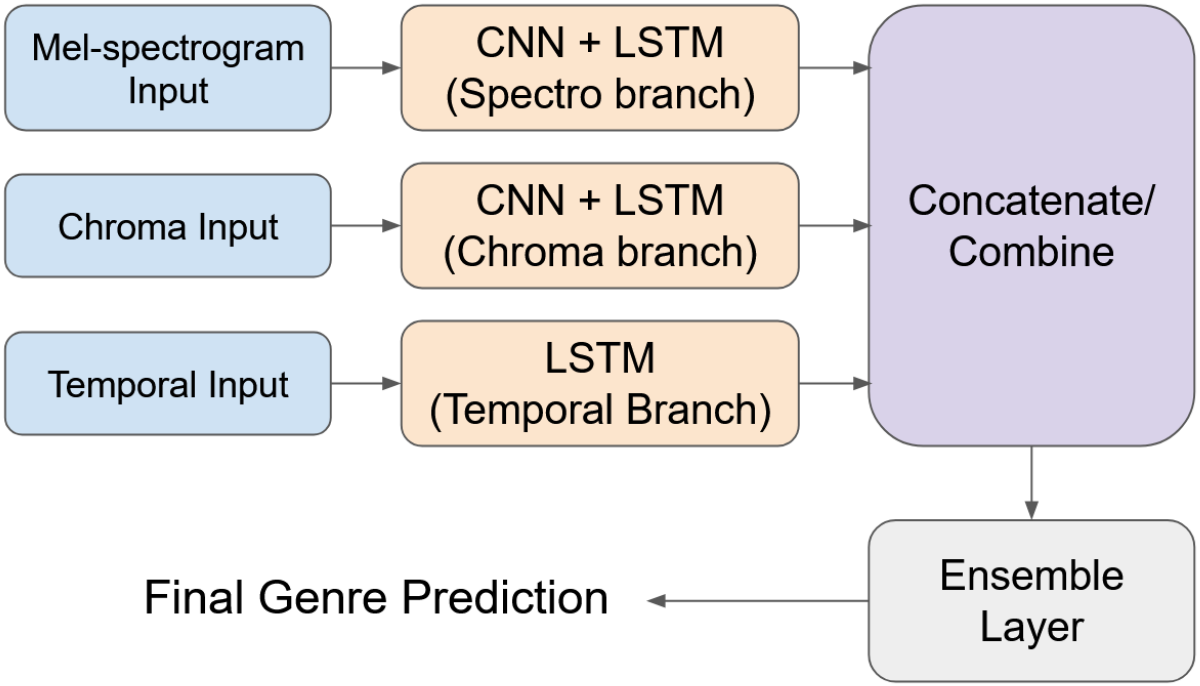


Figure 4: Combining Specialized Deep Learning Models

Ensemble Model: Accuracy and Macro-F1

Model	Accuracy (Mean±S.E.)	Macro-F1 (Mean±S.E.)
Mel-spectrogram	0.878 ± 0.007	0.878 ± 0.007
Chroma	0.619 ± 0.010	0.610 ± 0.011
Temporal	0.595 ± 0.004	0.592 ± 0.004
Ensemble	**0.886 ± 0.013**	**0.886 ± 0.012**

Table 3: Performance of the three feature-specific models and the ensemble model over 100 randomized 70/30 splits.

Ensemble Model: Performance Comparison

- **Compare with Linear Models:** Linear models struggle to capture the complex, non-linear relationships that evolve over time and across frequencies. The ensemble model can learn these intricate patterns more effectively.
- **Compare with Basic Neural Networks:** Shallow NNs may not simultaneously capture both the spatial hierarchies within the spectrogram and the long-range temporal dependencies. The ensemble addresses this by employing CNN+LSTM to model temporal evolution within those representations.
- **Compare with SVM:** While the SVM demonstrates the effectiveness of well-engineered aggregated features and non-linear decision boundaries, our ensemble model achieves a competitive accuracy, leveraging the automatic feature learning capabilities of DNNs directly on richer data representations, offering potential advantages in capturing more nuanced, data-driven features.

Final Results

Model	Accuracy (Mean±S.E.)	Macro-F1 (Mean±S.E.)
Forward Selection	0.614 ± 0.003	$0.957 \pm < 0.001$
Regularization	0.709 ± 0.002	$0.968 \pm < 0.001$
PCA-based	0.578 ± 0.003	$0.953 \pm < 0.001$
CNN	0.300 ± 0.019	0.295 ± 0.003
FNN	0.161 ± 0.003	0.130 ± 0.002
MLP	0.214 ± 0.004	0.151 ± 0.005
SVM	$\mathbf{0.9110 \pm 0.005}$	$\mathbf{0.9107 \pm 0.005}$
Mel-spectrogram	0.878 ± 0.007	0.878 ± 0.007
Chroma	0.619 ± 0.010	0.610 ± 0.011
Temporal	0.595 ± 0.004	0.592 ± 0.004
Ensemble	0.886 ± 0.013	0.886 ± 0.012

Discussion: Future Ideas

- Advance the models to include artists, labels, sub-genres, etc.
- Using large unlabeled audio to help with the learning process before tuning.
- Compare with other datasets, can adaptation methods help make the results more similar?
- How can we make these models more efficient for apps or live settings?
- Have people give feedback on the models and see if they're relevant.
- How to continuously update the models without starting over as advancements come in
- Incorporating data augmentation techniques to improve the robustness and generalization of our models.

Thank You!