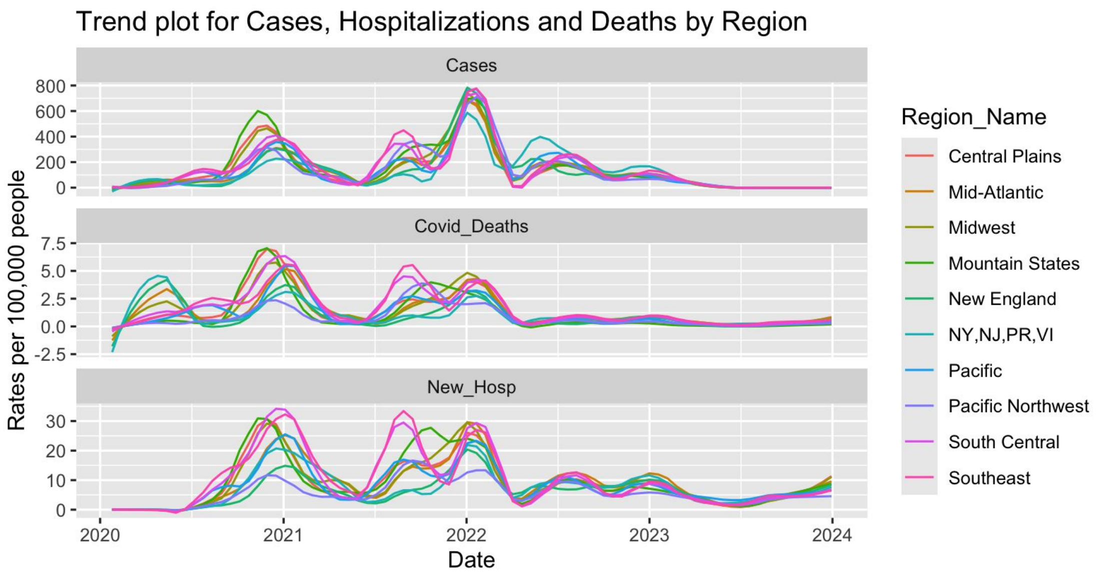
Predicting Covid-19 Death Rates Using Machine Learning

Group 9: Kelly Li & Yana Xu

Introduction

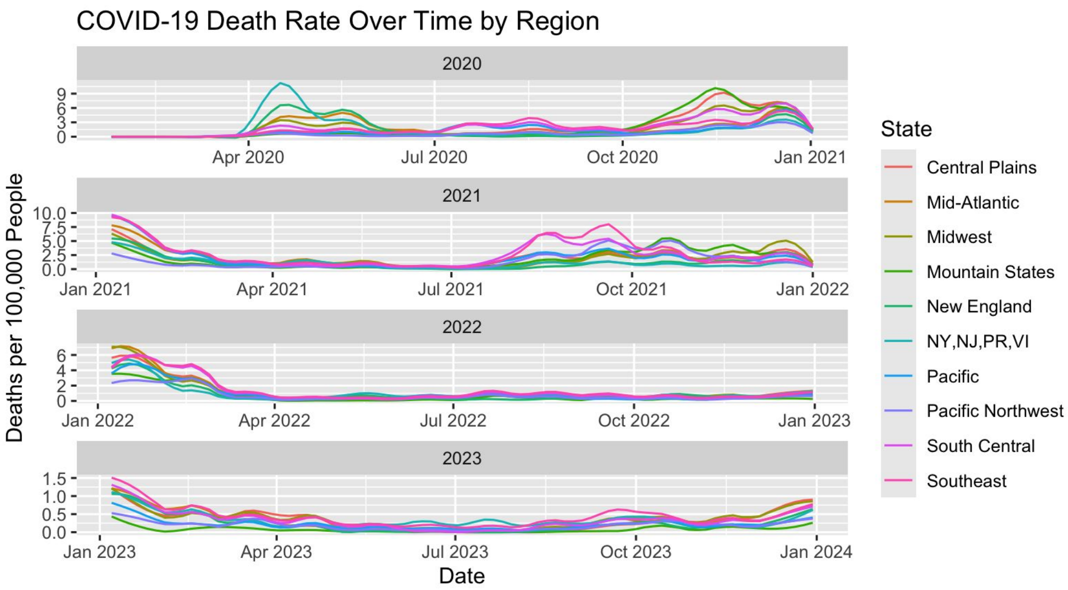
- **Background**
  - Covid-19 has affected millions of people around the world and significantly challenge the public health system
  - Predicting covid death rates can help us prepare for future outbreaks
- **Objective**
  - Use different machine learning models for prediction
  - ***Covid death rates*** *~ Cases + New_Hosp + ICU_Hosp + Series_Complete_Pct + Booster + Bivalent_Booster_Pct*

Exploratory Data Analysis (EDA)



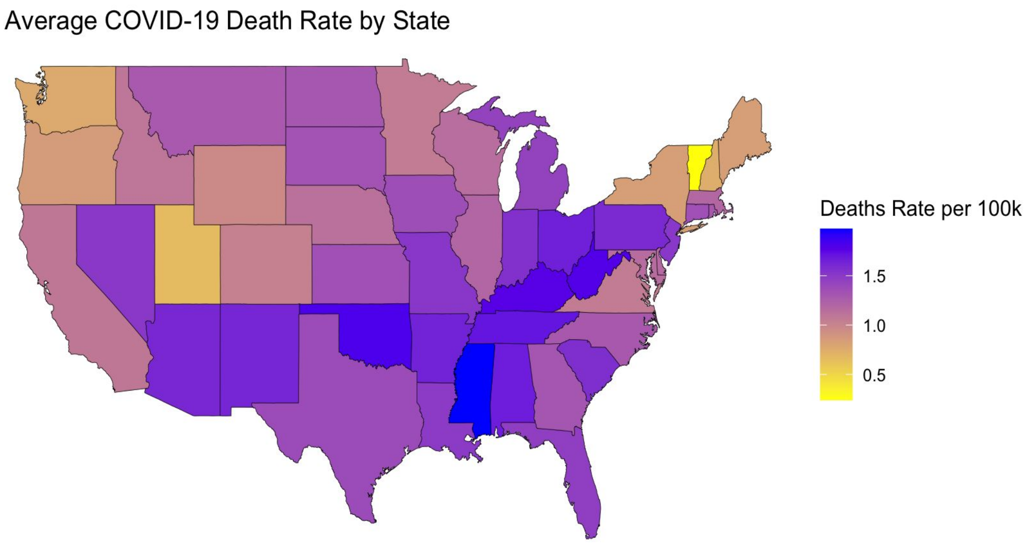
- South Central and Southeast showed higher peaks
- Regional differences narrowed over time
- All follow similar wave pattern over time
- Predictive potential

Exploratory Data Analysis (EDA)



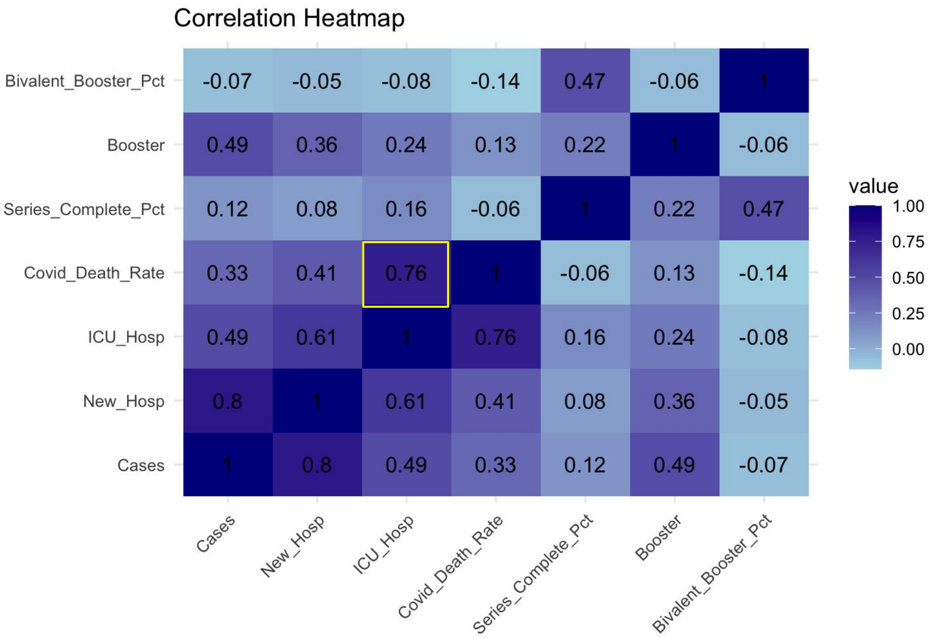
- South Central showed higher peaks after 2020
- Death rates declined significantly after March 2022

Exploratory Data Analysis (EDA)



- **Highest**
  - East south central
  - MS, OK, KY
- **Lowest**
  - New england
  - VT, UT, WA
- Matches trends

Exploratory Data Analysis (EDA)



Methods: Data

- Cases
- Population (2020-2024)
- Hospitalization
  - Total hosp
  - ICU occupancy
- Deaths
  - Total deaths
  - Covid deaths
- Vaccination
  - Series complete
  - Booster
  - Bivalent booster

- **Data wrangling:** Joining tables
- MMWR_Year, MMWR_Month, MMWR_Week
- Convert NAs to 0
- Compute covid death rate per 100,000 people
- **Final data**: 10712 rows, 21 columns

Methods: Machine Learning Method Overflow

2021 Data → Train Models → Predict 2022 → Compare Actual

We took state-level monthly and weekly averages of cases, hospitalizations, vaccination and booster rates, plus ICU load

Monthly: group_by(State, year, month) → mean of all features + target

Weekly: group_by(State, year, week) → same

Train: all 2021 rows

Test: all 2022 rows

NA handling: na.omit()

- **Linear Regression**
- **K-Nearest Neighbors:** 5-fold CV, preProcess = c("center","scale"), tuneLength = 7
- **Random Forest:** ntree = 50
- **XGBoost**: objective = "reg:squarederror", nrounds = 50

Metrics:

RMSE, MAE, MAPE, and R^2.

Results: How We Measured "Good"

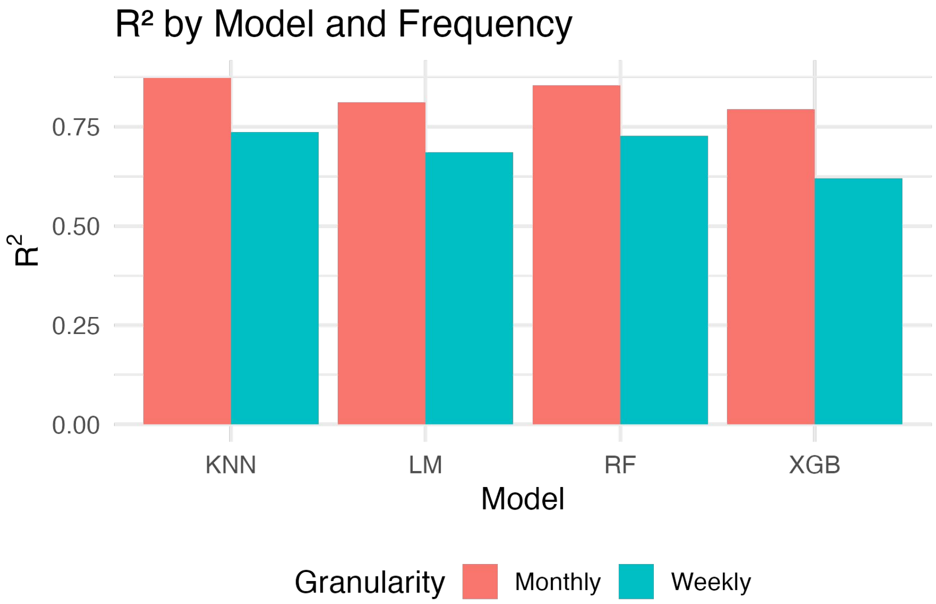
- **R^2** – How much variation the model explains (higher = better)
- **RMSE** – Typical size of a prediction error (lower = better)
- **MAE** – Average absolute error (lower = better)

> print(results)

	RMSE	MAE	MAPE	R2	Model	Granularity
1	0.6967264	0.4920104	Inf	0.8106763	LM	Monthly
2	0.5706481	0.4216568	Inf	0.8729961	KNN	Monthly
3	0.6130146	0.4598164	Inf	0.8534378	RF	Monthly
4	0.7273703	0.4611908	Inf	0.7936561	XGB	Monthly
5	0.9550125	0.6219931	Inf	0.6848797	LM	Weekly
6	0.8729090	0.5499949	Inf	0.7367332	KNN	Weekly
7	0.8898849	0.5446665	Inf	0.7263938	RF	Weekly
8	1.0496304	0.6403851	Inf	0.6193455	XGB	Weekly

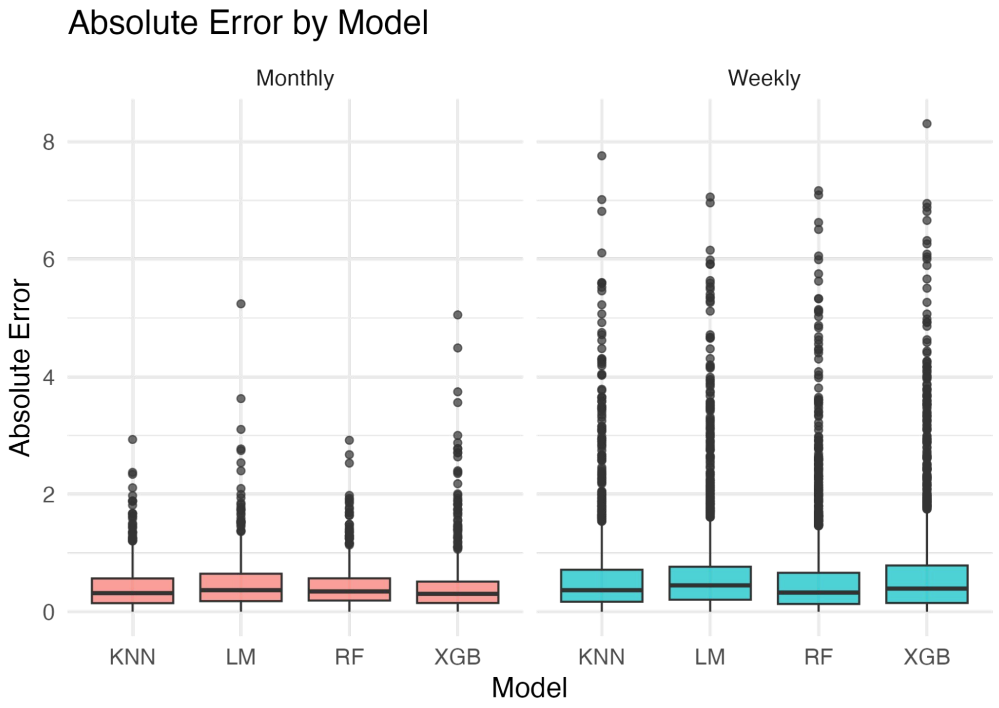
Results: R^2 Results

- Monthly aggregation consistently outperforms weekly data for all models
- KNN on monthly is best ($R^2 \approx 0.87$).

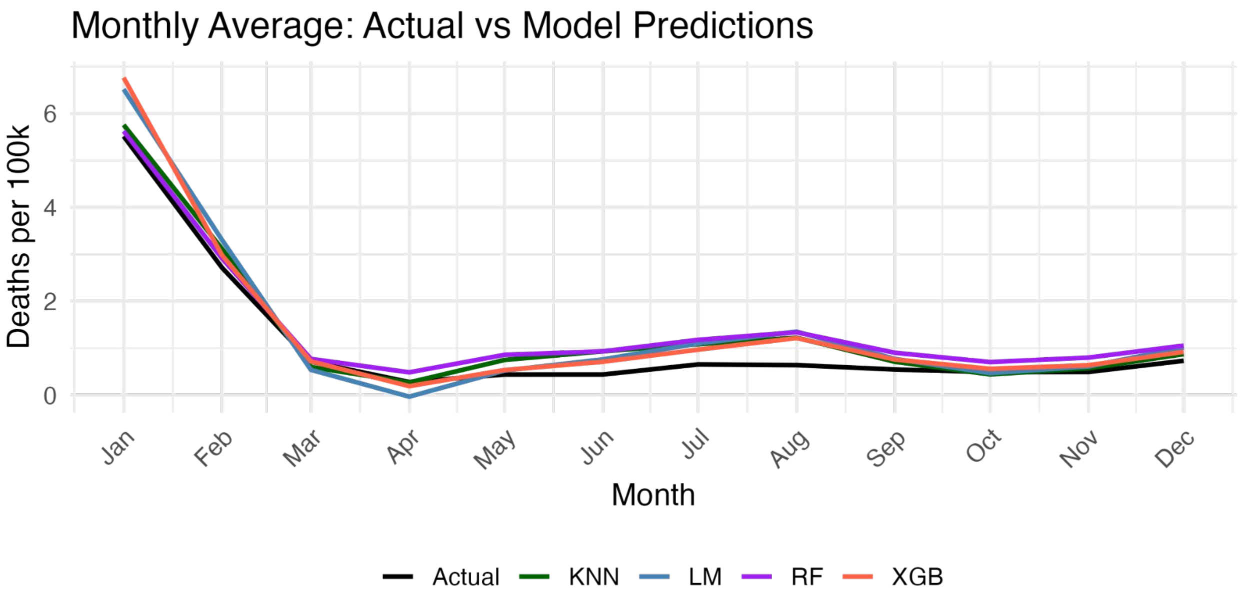


Results: Error Spread

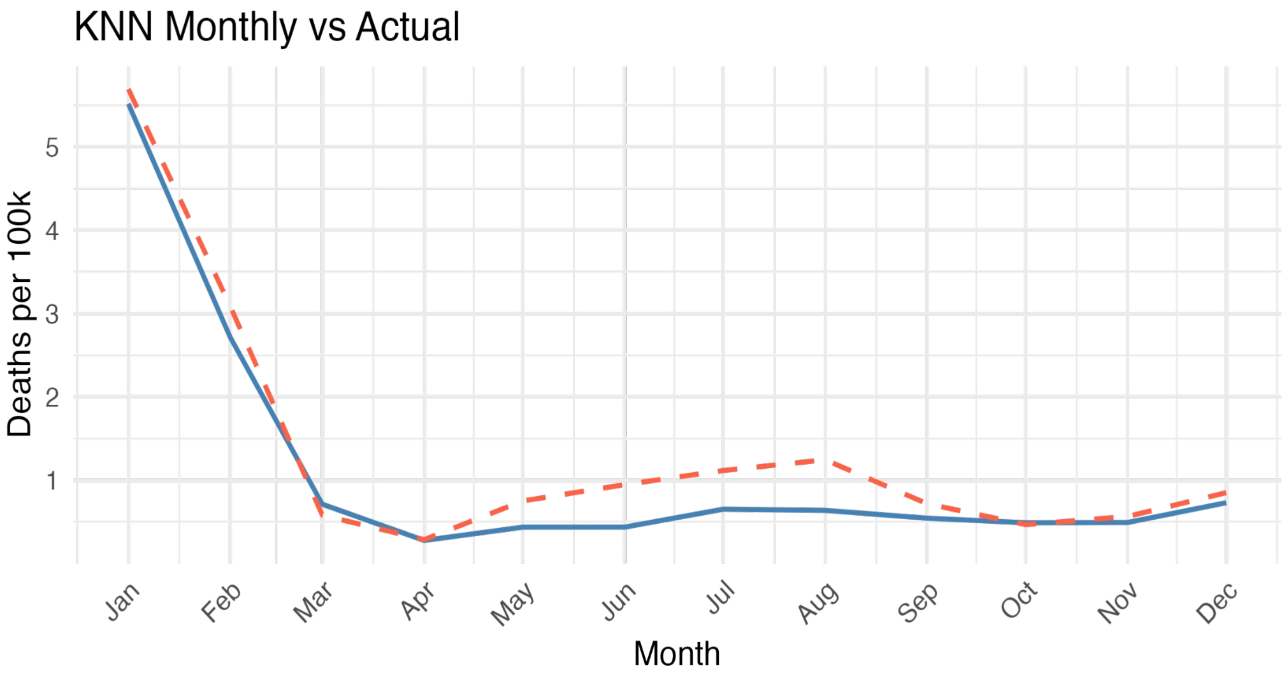
KNN grouped by monthly has the smallest errors and RF is close.



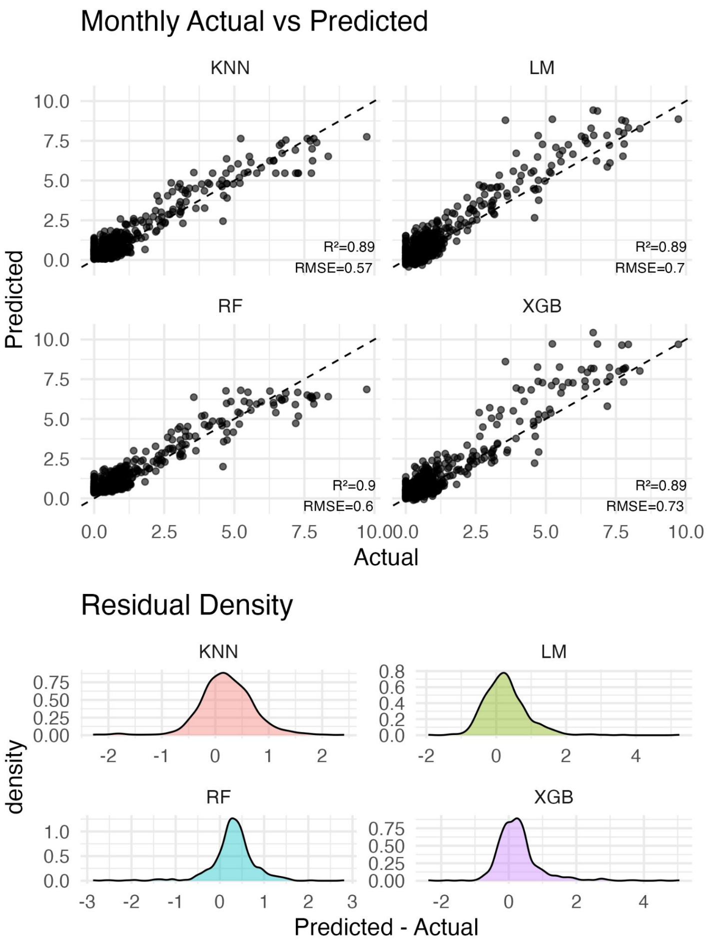
Results: Tracking the Trend



Results: KNN(Best Model) time series plot



Results: Detailed Fit & Residuals



Results: Key Findings

- Monthly aggregation outperforms weekly data for mortality prediction
- KNN provides best overall performance (RMSE: 0.57, R^2: 0.87)
- All models struggle with extreme values

Discussion

- **Monthly forecasting recommended for pandemic planning**
- **Model Insights**

LM & XGB (worse)

- Miss key non-linear trends
- XGB under-tuned for limited data

KNN & RF (better)

- Learn local patterns
- Robust to noise and spikes

Discussion: Why does it Matter?

- Pick the right tool in future outbreaks
- Quick, data-driven early warnings
- Helps public health plan resources