Camera Traffic Counts: Modeling

By: John Trelford, Lucas Chiang, Travis Welsh, Brian Pham, and Ritesh Penumatsa
(Group 11)

Dataset Overview

Camera Traffic Counts:

- 15-minute interval traffic by intersection in Austin
- Collected by GRIDSMART optical traffic detectors

Subsetted Data

How?

- Filter to only rows where year = 2019
- 14,717,624 rows in dataset where year = 2019

Why?

- Prevents interference from the COVID-19 pandemic period
- The full Camera Traffic Counts dataset was 82.1 million rows
  - Overwhelms RAM
  - Computationally Intensive

Shape - (Rows, Columns): (14717624, 19)

Predictive Problem 1:
Predicting expected volume of vehicles give day of the week and time

Problem Setup

Input Variables: Day of Week, Hour
Target Variable: Volume

- Real-valued outcome

Justification of Feature Set:

- Data is available for every observation
- No leakage - predictors come before the prediction
- Expected signal - volume follows time-based patterns
- Encoding - all predictors are already numeric

Day of the week was one-hot encoded to properly weight the categorical variable (day of the week).

Choice of Model Class and Justification

Model Class 1: Linear Regression

- Appropriate for temporal data with smoothly changing trends
- Easy to interpret the coefficients
- Low variance
- Low computational cost
- Avoids overfitting (few features yet millions of samples)

Model Class 2: Random Forest

- Able to model complex relationships between variables
- More resistant to outliers in the data

Evaluation and validation procedure

How we split the dataset?:

- Training set: January-October
- Testing set: November-December
- Avoids temporal leakage, so only past data predicts future observations
    - Ensures an unbiased estimate of generalization because it simulates the real world scenario of predicting traffic volume without knowledge of the future.

- k-fold CV is inappropriate because it breaks the time order

Justify metric choice(s) by the decision/prediction context:

- RMSE ensures there are no massive mispredictions
- R-Squared indicates how well the model captures the overall trends

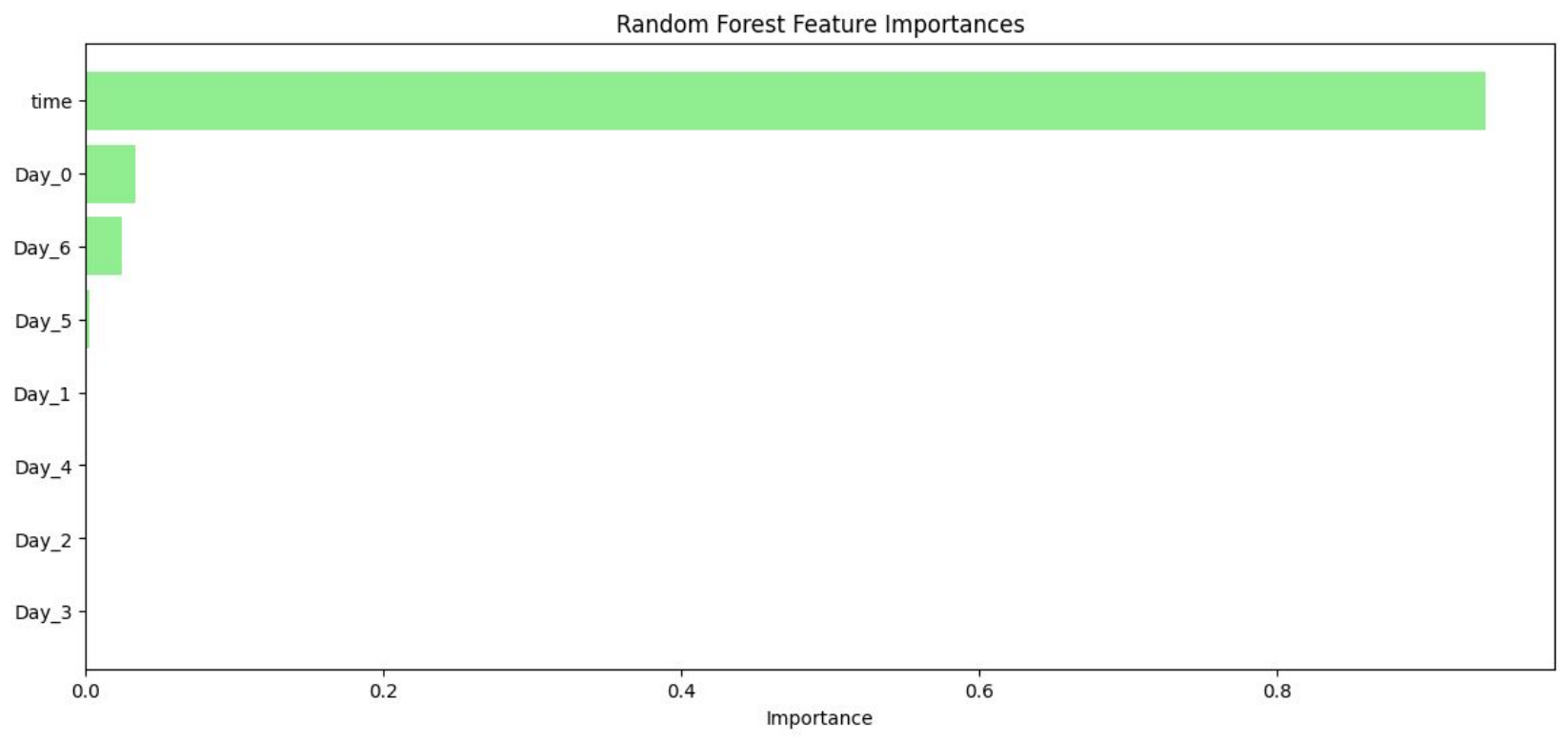
Choice of Hyperparameters and Loss Function

- Linear Regression:
  - No tunable hyperparameters
  - Search method: minimizing MSE
    - Penalizes larger errors more heavily, and we want to protect again large mispredictions

Hyperparameters and Loss Function for Random Forest

- Created a subset of 500,000 randomly selected rows
- Used a random search method to test 20 combinations of optimal values for n_estimators, max_depth, min_samples_leaf, and min_samples_split.
- This is better than a grid search because of the data set's large size.
- n_estimators: 50
- max_depth: 9
- min_samples_leaf: 2
- min_samples_split: 5

Random Forest automatically uses MSE for the loss function. It is a good fit because the target (volume) is a continuous variable.
The MSE was found to be 3377.39.


Random Forest Feature Importances
time
Day_0
Day_6
Day_5
Day_1
Day_4
Day_2
Day_3
0.0
0.2
0.4
0.6
0.8
Importance

Evaluation and conclusion

Linear Regression:

- RMSE: 57.2460
- R-Squared: 0.0637
- Training Time: 9.26 seconds
- Energy consumed: 0.000124 kWh
- CO2 emissions: 0.000059 kg CO2eq

Random Forest:

- RMSE: 58.1373
- R-Squared: 0.0344
- Training Time: 29 minutes, 5.93 seconds
- Energy Consumed: 0.025459 kWh
- CO2 Emissions: 0.012120 kg CO2 eq

Overall, Linear Regression is more precise and less computationally intensive than Random Forest. Linear Regression could be used with city planning.

Predictive Problem 2:

Predicting average speed of vehicles with precipitation type (rain vs. snow) and temperature.

Problem Setup

Input Variables: Average Precipitation (in), Average Snow (in), Average Temperature (°F)

Other Feature Variables: Volume, Month, Day, Year, Day of the Week, or any other numeric variables in the dataset

Decided to drop temporal variables and non-transformed categorical variables because it would not be applicable in this problem.

Target Variable: Average Speed (mph)

Justification of Feature Set: We want to use as many variables as possible to adjust for other factors influencing traffic speed, allowing the model to improve predictions.

Choice of Model Class and Justification

Model Class 1: Random Forest Regression

Model Class 2: Gradient Boosting Regression

Why it is important?:

Random Forest Regression: Robust, averages multiple trees to capture nonlinear relationships and interactions.

Gradient Boosting Regression: Builds trees sequentially to correct errors, capturing subtle patterns for higher accuracy.

Reason for using both: Allows comparison of parallel vs. sequential learning and ensures reliable predictions.

Evaluation and Validation Procedure

Dataset Split:

- 80% training, 20% testing. Split the samples into the first 80% of the data sorted by dates as training and the remaining 20% of the data sorted by dates as testing

Validation Procedure:

- **GridSearchCV:** 5-fold CV used to systematically tune hyperparameters, ensuring the model generalizes well and selects the best parameter combination.
- **Cross-Validation:** Splitting the training data into 5 folds allows each subset to serve as a validation set once, providing a robust estimate of model performance and reducing variance from a single split.
- **Train-Test Split:** Held-out test set evaluates final model performance on unseen data, providing an unbiased estimate of real-world predictive accuracy.

Evaluation Metrics:

- **R^2:** Measures the proportion of variance explained by the model.
- **RMSE:** Quantifies prediction error in miles per hour.
- These metrics together provide a clear and comprehensive assessment of model fit and reliability.

Evaluation and Procedure cont.

Justification of Metrics:

- R^2 indicates how well the model captures the total variance explained by traffic speed.
- RMSE ensures errors are small in practical terms, which is critical for accurate traffic speed predictions.

Choice of Hyperparameters and Loss Function

Key Hyperparameters Tuned:

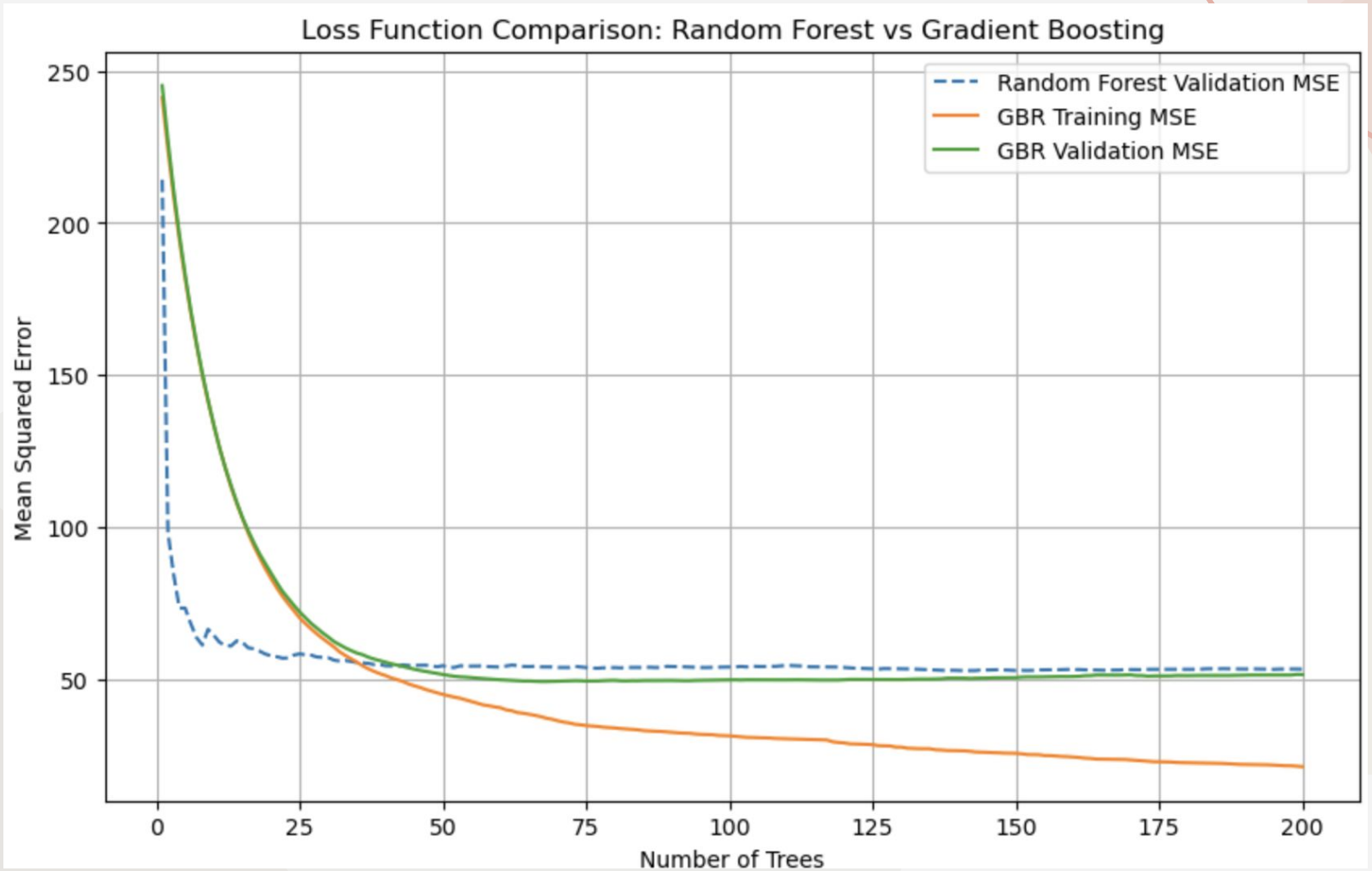
- n_estimators, max_depth, learning_rate, min_samples_split, min_samples_leaf
- **Search Method:** GridSearchCV with 5-fold cross-validation
- **Stopping Criteria:** Number of trees (n_estimators) and early stopping based on cross-validation performance

Loss Function Justification:

- Optimized **mean squared error (MSE)**, suitable for our continuous target variable (average speed in miles per hour). MSE penalizes larger deviations more heavily, encouraging the model to minimize significant prediction errors.

Link Between Surrogate Loss and Decision Metrics:

- The model learned by minimizing MSE, but we reported R^2 and RMSE since they better describe how well predictions match real traffic speeds. RMSE turns the training loss into a more intuitive measure of prediction accuracy, helping connect the optimization goal to real-world performance.


Loss Function Comparison: Random Forest vs Gradient Boosting
Random Forest Validation MSE
GBR Training MSE
GBR Validation MSE
Mean Squared Error
Number of Trees
250
200
150
100
50
0
25
50
75
100
125
150
175
200

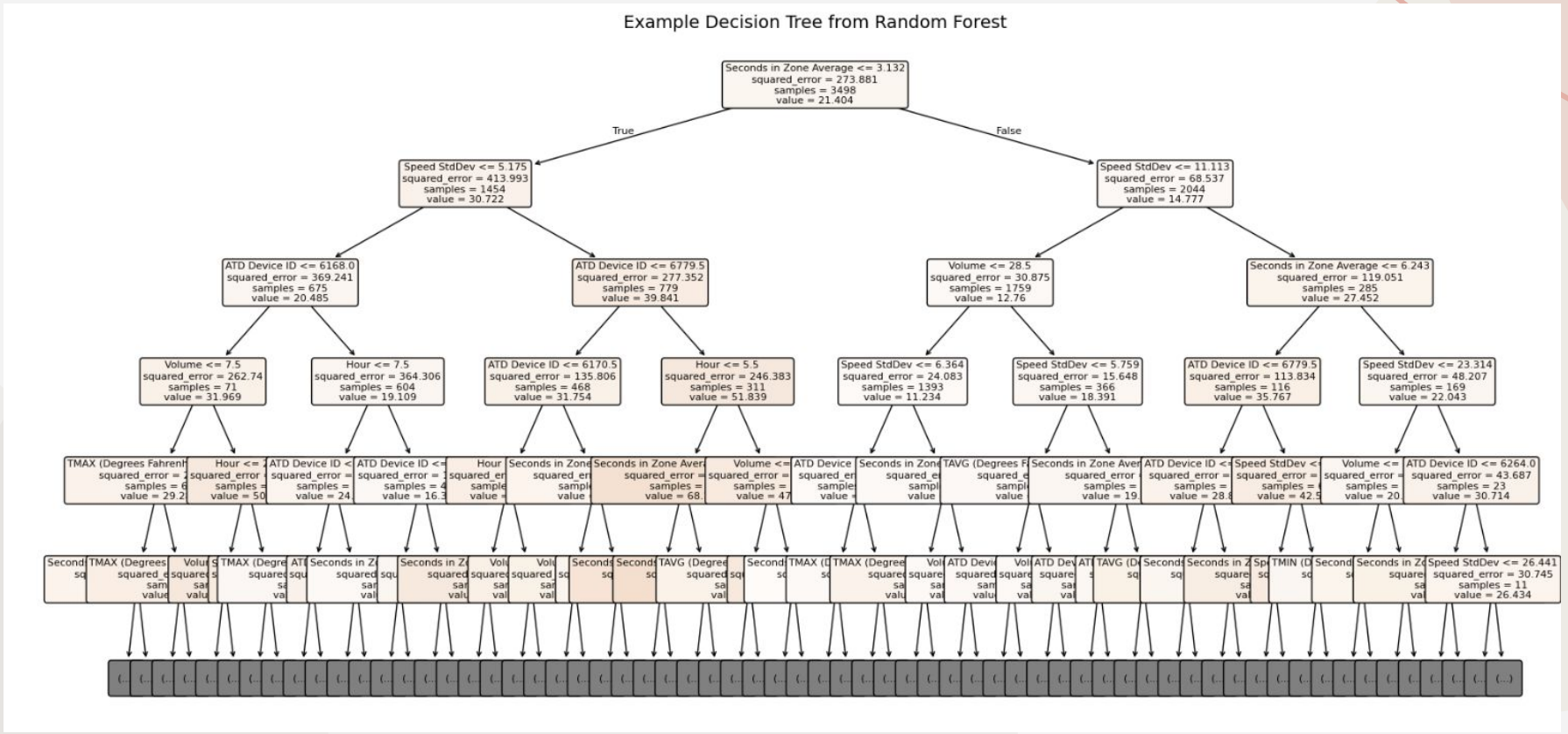
Evaluation

Random Forest Regressor (RFR):

- **Best Parameters:** {'max_depth': None, 'min_samples_leaf': 2, 'min_samples_split': 2, 'n_estimators': 200}
- **Best CV R^2:** ≈0.7518 (across 5 folds)
- **Test Performance:** R^2 ≈ 0.809, RMSE ≈ 0.194
- **Interpretation:** Strong predictive performance; robust to nonlinear relationships and feature interactions.

Gradient Boosting Regressor (GBR):

- **Best Parameters:** {'learning_rate': 0.05, 'max_depth': 7, 'min_samples_leaf': 4, 'min_samples_split': 10, 'n_estimators': 100}
- **Best CV R^2:** ≈0.7498 (across 5 folds)
- **Test Performance:** R^2 ≈ 0.813, RMSE ≈ 0.436
- **Interpretation:** Comparable predictive accuracy; sequential boosting captures subtle patterns but slightly higher RMSE than RFR.


Example Decision Tree from Random Forest
Seconds in Zone Average <= 3.132
squared_error = 273.881
samples = 3498
value = 21.404
True
False
Speed StdDev <= 5.175
squared_error = 413.993
samples = 1454
value = 30.722
Speed StdDev <= 11.113
squared_error = 68.537
samples = 2044
value = 14.777
ATD Device ID <= 6168.0
squared_error = 369.241
samples = 675
value = 20.485
ATD Device ID <= 6779.5
squared_error = 277.352
samples = 779
value = 39.841
Volume <= 28.5
squared_error = 30.875
samples = 1759
value = 12.76
Seconds in Zone Average <= 6.243
squared_error = 119.051
samples = 285
value = 27.452
Volume <= 7.5
squared_error = 262.74
samples = 71
value = 31.969
Hour <= 7.5
squared_error = 364.306
samples = 604
value = 19.109
ATD Device ID <= 6170.5
squared_error = 135.806
samples = 468
value = 31.754
Hour <= 5.5
squared_error = 246.383
samples = 311
value = 51.839
Speed StdDev <= 6.364
squared_error = 24.083
samples = 1393
value = 11.234
Speed StdDev <= 5.759
squared_error = 15.648
samples = 366
value = 18.391
ATD Device ID <= 6779.5
squared_error = 113.834
samples = 116
value = 35.767
Speed StdDev <= 23.314
squared_error = 48.207
samples = 169
value = 22.043
ATD Device ID <= 6264.0
squared_error = 43.687
samples = 23
value = 30.714
Speed StdDev <= 26.441
squared_error = 30.745
samples = 11
value = 26.434

Compute / Energy Considerations

- **Total Model Fits:**
  The Gradient Boosting Regressor (GBR) required significantly more compute, involving 1,215 total fits (243 candidates × 5 folds). The Random Forest Regressor (RFR) was substantially more efficient, requiring only 120 total fits (24 candidates × 5 folds).
- **Training Time per CV Fit:**
  Individual training times varied, with each cross-validation fold for the GBR taking approximately 1.7–6.9 seconds, depending on the complexity of the hyperparameters.
- **Energy Optimization:**
  To manage computational resources and limit energy usage, we employed a moderate dataset size and restricted the hyperparameter search space (param_grid). Specifically for GBR, we utilized pruning to get smaller n_estimators and shallower trees to reduce training complexity and energy consumption.
- **Normalized Prediction Cost:**
  Both optimized models are highly efficient post-training. The cost of a single prediction is fast (milliseconds), confirming their suitability for real-time applications such as traffic speed estimation.

Compute / Energy Considerations

Model	Total Training Fits	Avg. Training Time per Fit (s)	Energy Consumed (kWh)	CO_2 Emitted (kg CO_2 eq)
Random Forest Regressor (RFR)	120	1.2	**0.031 kWh**	**0.013 kg CO_2 eq**
Gradient Boosting Regressor (GBR)	1,215	3.9	**0.142 kWh**	**0.061 kg CO_2 eq**

The GBR consumed ~4.6× more energy and emitted nearly 5× more CO_2 than the RFR.
RFR seems to be the better cost-efficient model.

Which Model Performed Better?

- Both the Gradient Boosting Regressor and the Random Forest Regressor had a similar $R^2 \approx 0.81$.
- The RFR has a lower RMSE, suggesting marginally more precise predictions.
- GBR may offer greater insight into complex non-linear feature interactions through its feature importance scores.
- RFR had a faster performance because of fewer candidates per fit.

Final Choice: The Random Forest Regressor is the preferred model because of its lower RMSE, faster training time, and lower sensitivity to hyperparameter tuning than GBR.

Thank You!
Any Questions?