AI Tradeoff

Team A

Chenxi Li, Kartik Moudgil, Yuchen Wang, Zhouyi Yang

Which model is better in Primary Scenario

?

Random Forest vs Linear Regression Lasso

Which model is better in Primary Scenario

Goal:

1. Predict the popularity of movies for planning royalty payments and future movie acquisitions.
2. Show the popularity to users.

Requirement:

1. **Accurately** predict the popularity of movies **daily** (accurate enough for making proper acquisition decision and capture user interest) -> Normalized Popularity MSE less than 0.3
2. **Training time** and **Inference time** together must be less than one day (only want a movieId-popularity mapping daily)
3. **Model size** less than 1GB
4. Less need for interpretability

Popularity Prediction

Code for comparison between LR-Lasso and RFR

```
from sklearn.ensemble import RandomForestRegressor
from sklearn.model_selection import cross_val_score
from sklearn.model_selection import GridSearchCV

# Calculate accuracy of linear regression lasso
parameters = {'alpha':np.arange(0.001,100,0.001)}
linear_model = Lasso()
clf = GridSearchCV(linear_model, parameters, cv=5, scoring='neg_mean_squared_error')
clf.fit(X_train, y_train)
print(clf.best_params_, clf.best_score_)
print("The MSE of linear regression is: %f"%(-clf.best_score_))

# Calculate accuracy of random forest regression
parameters = {'max_depth':range(1,10),'n_estimators':range(1,10)}
clf = GridSearchCV(RandomForestRegressor(), parameters, cv=5, scoring='neg_mean_squared_error')
clf.fit(X_train, y_train.ravel())
print(clf.best_params_, clf.best_score_)
print("The MSE of random forest regression is: %f"%(-clf.best_score_))
```

1. Accuracy
2. Training time & Inference Time
3. Model Size
4. Interpretability
5. Linear / Non-linear

Accuracy

Split the data into train-test split of 70% : 30%

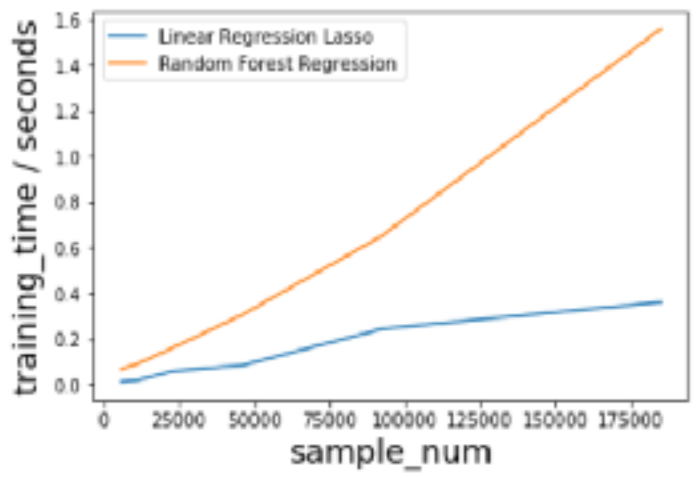
Linear Regression Lasso MSE: 0.496426

Random Forest MSE: 0.283986

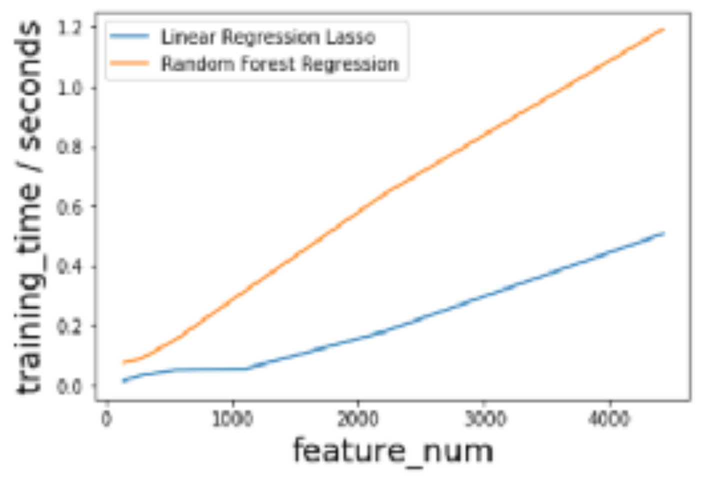
* MSE: Mean Square Error

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2.$$

Training Time

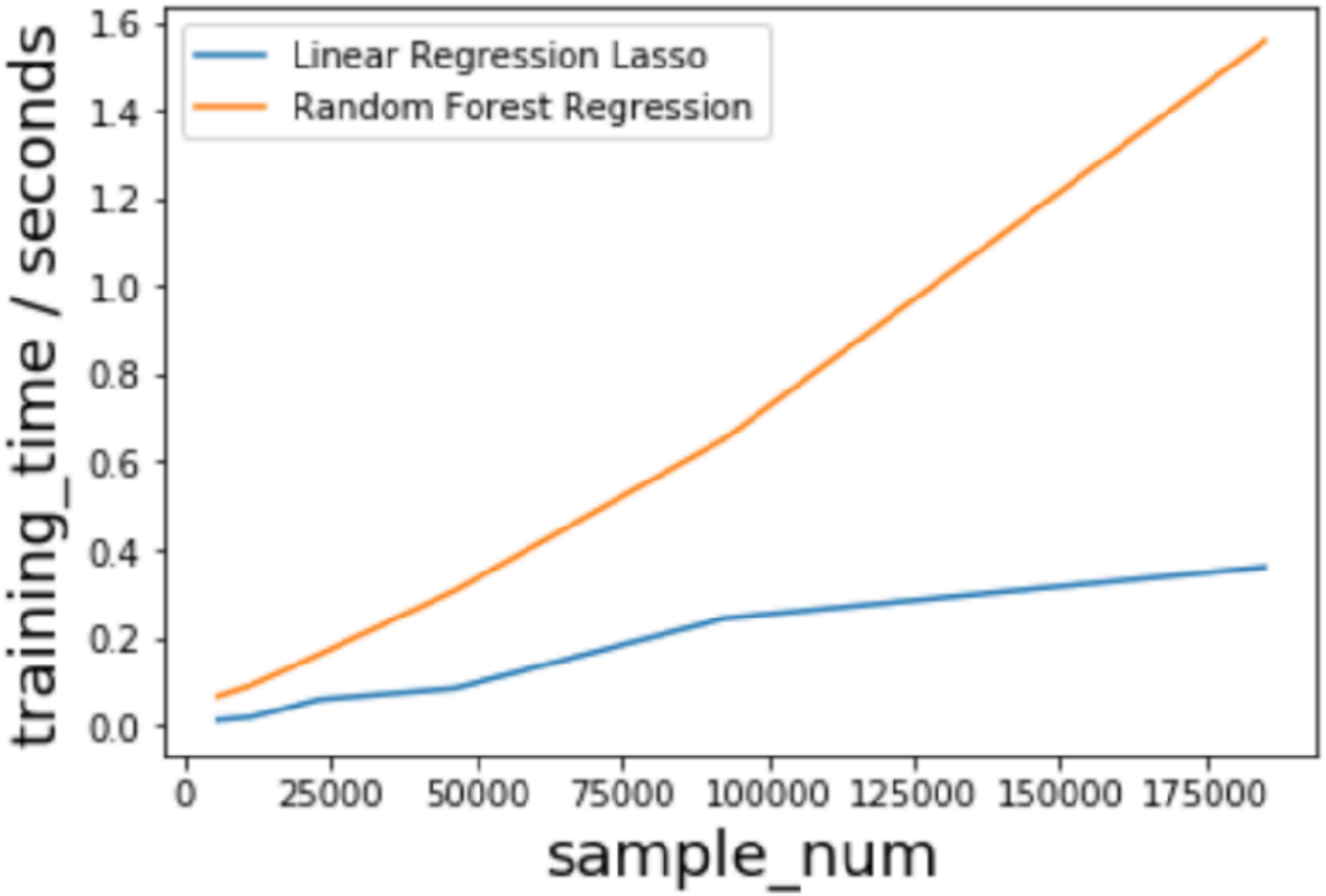


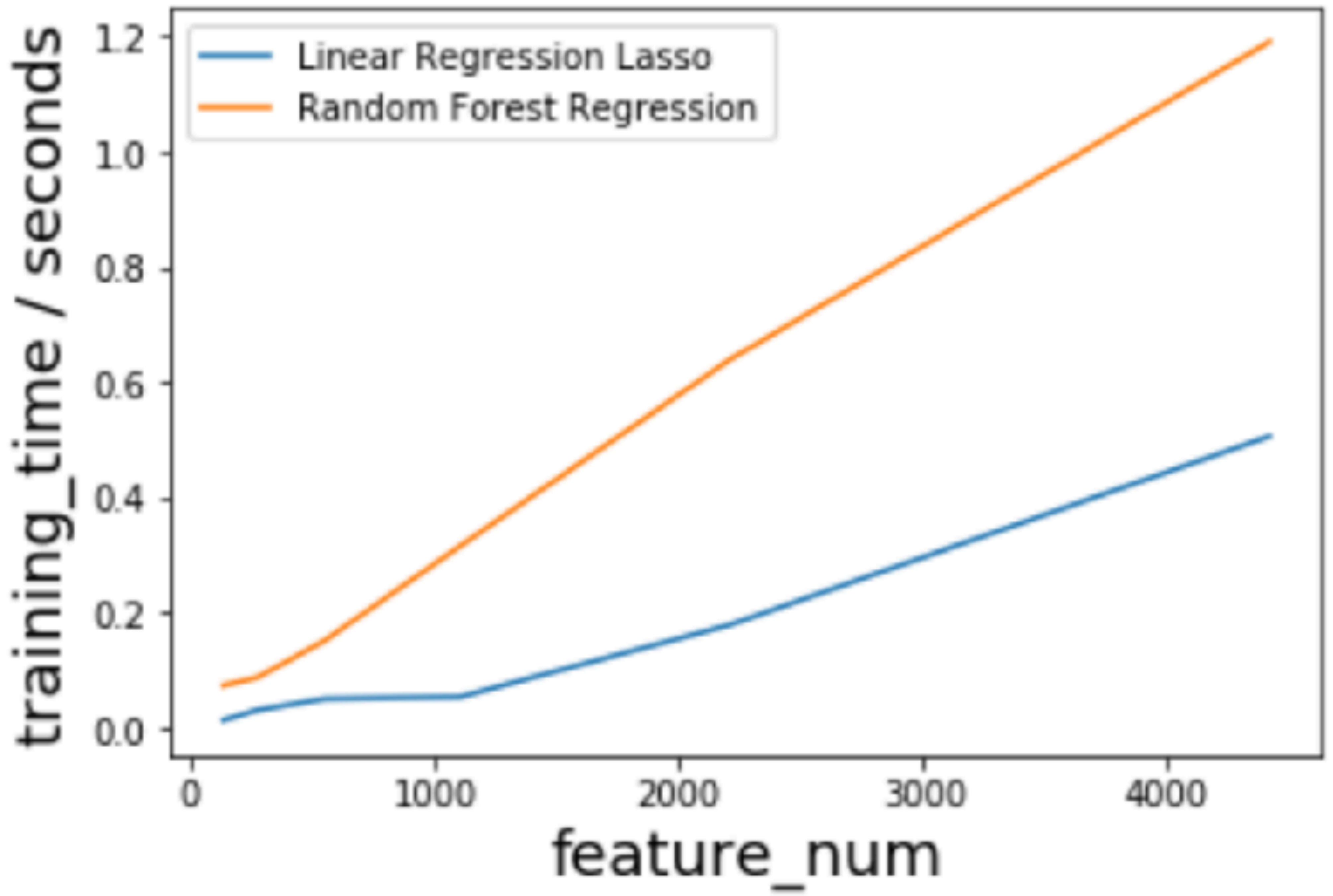
#sample



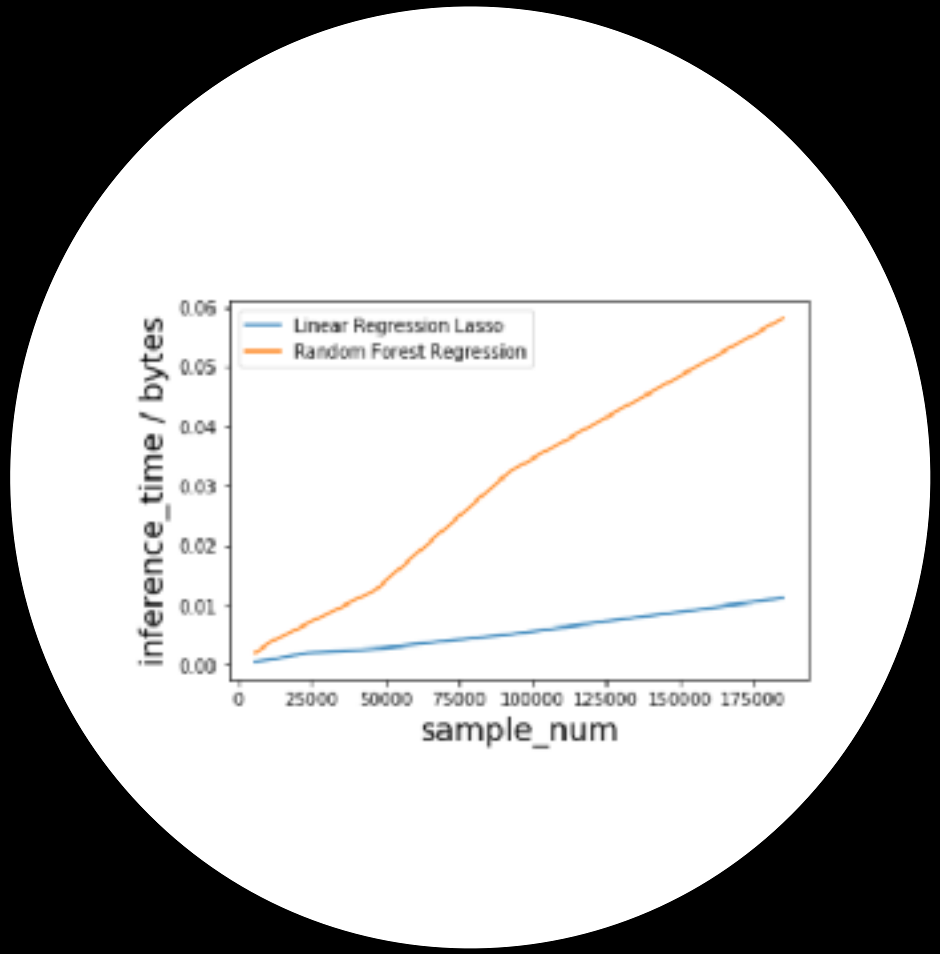
#feature

Lower: Linear Regression

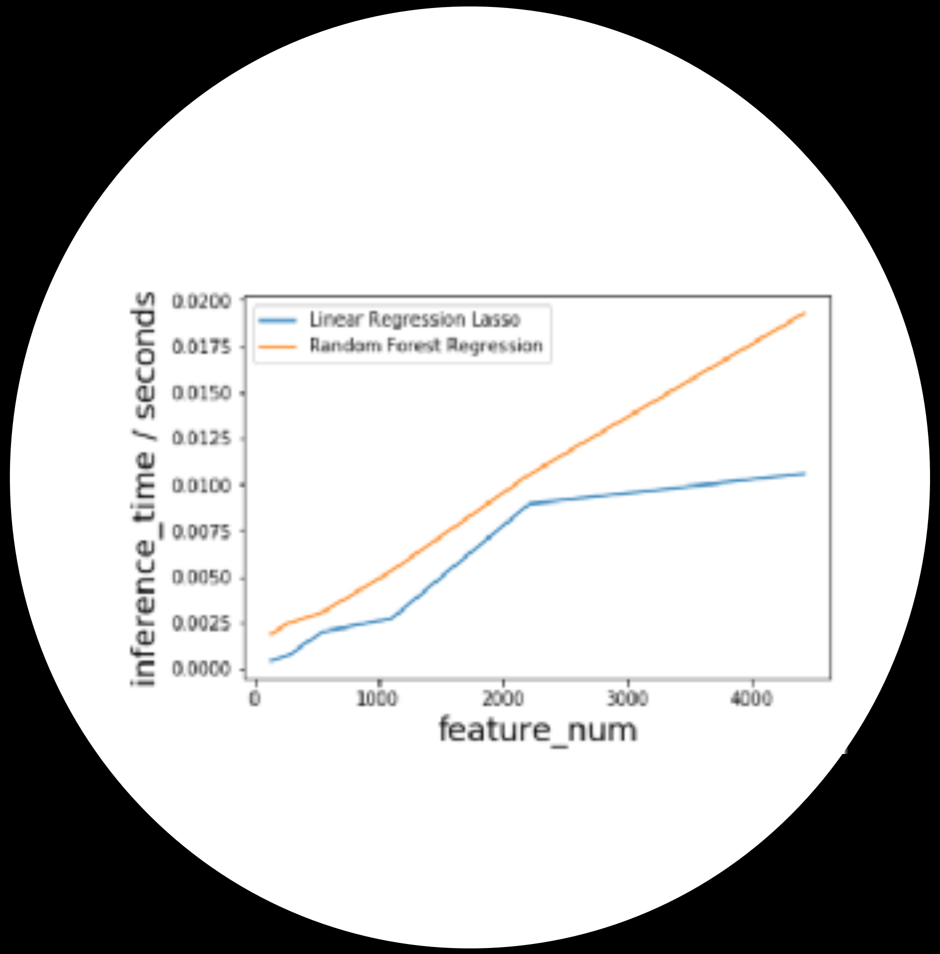




Inference Time

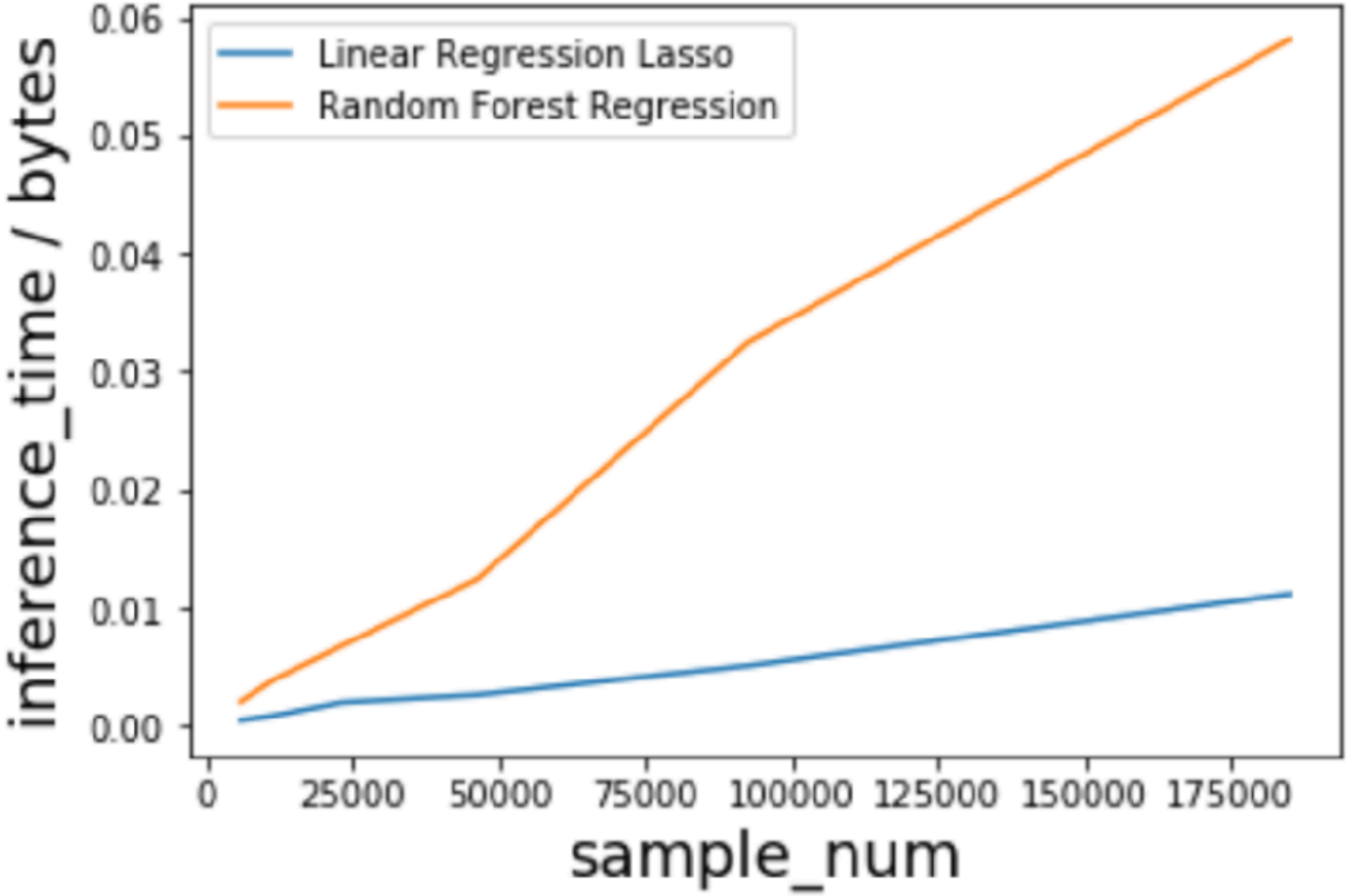


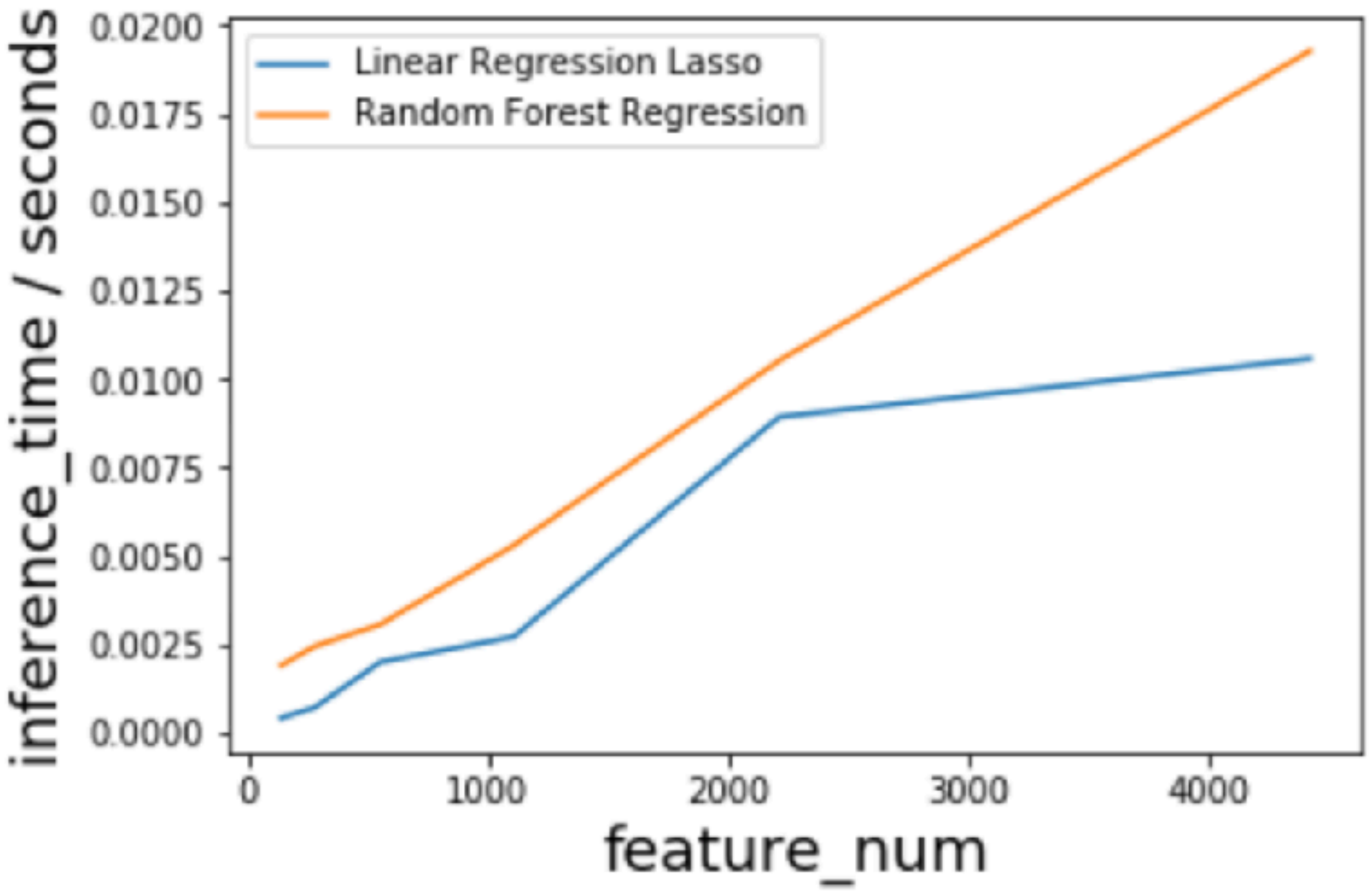
#sample



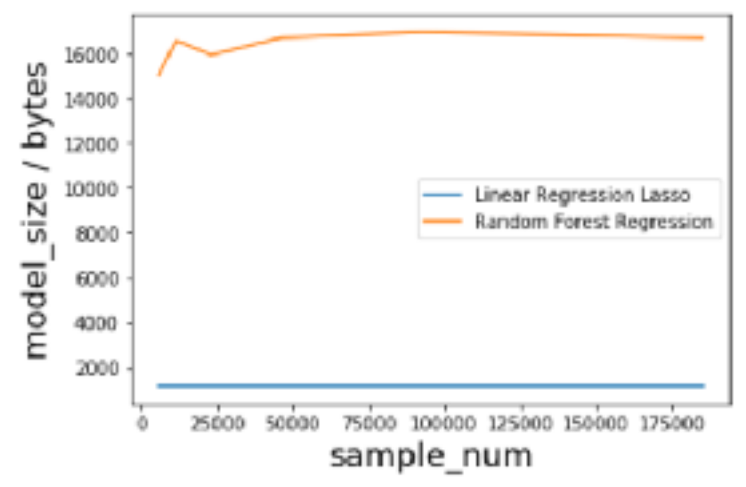
#feature

Lower: Linear Regression

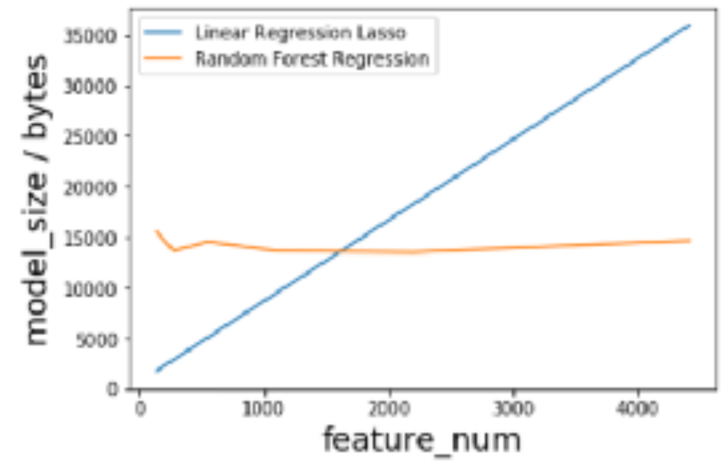




Model Size

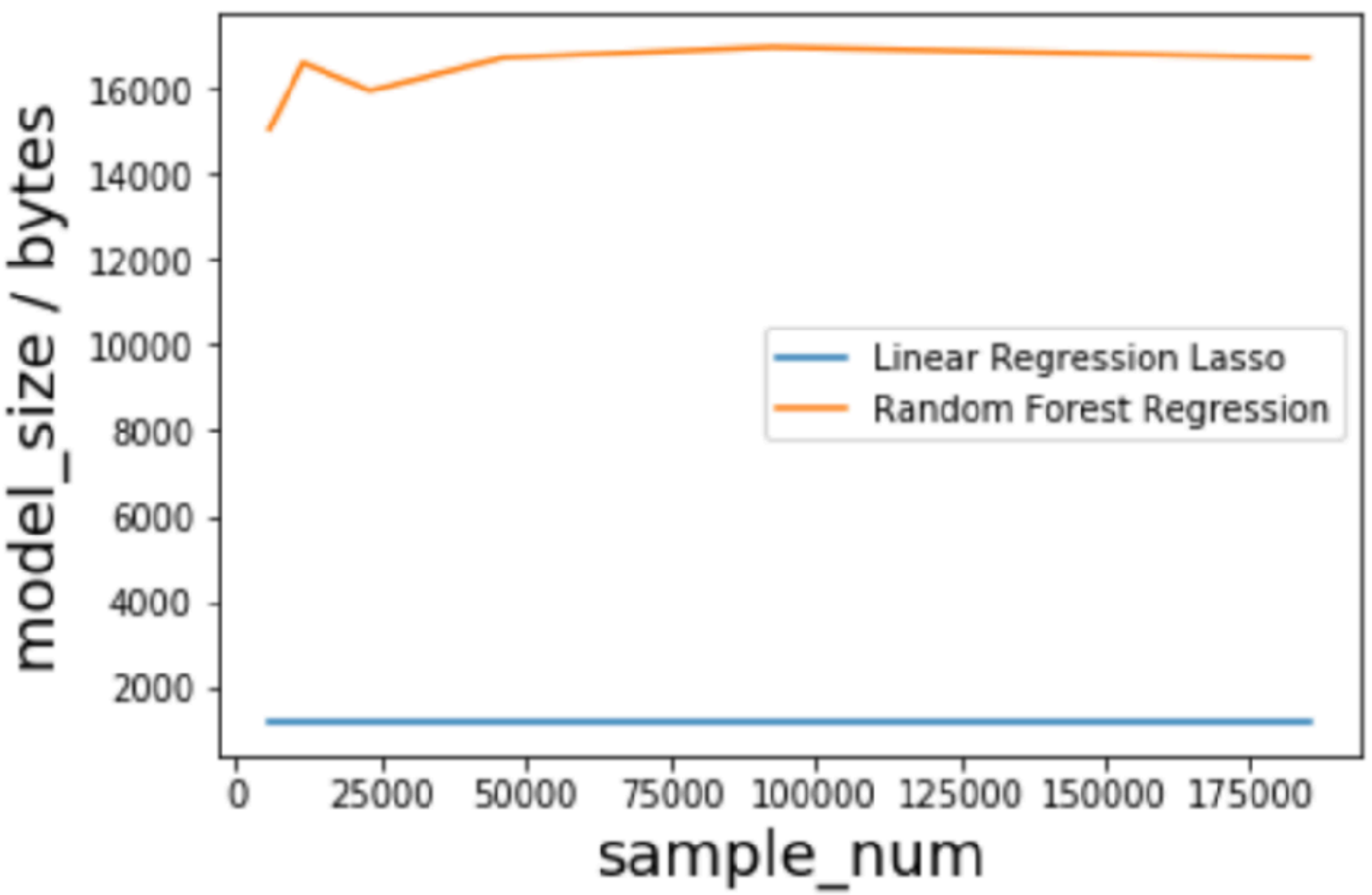


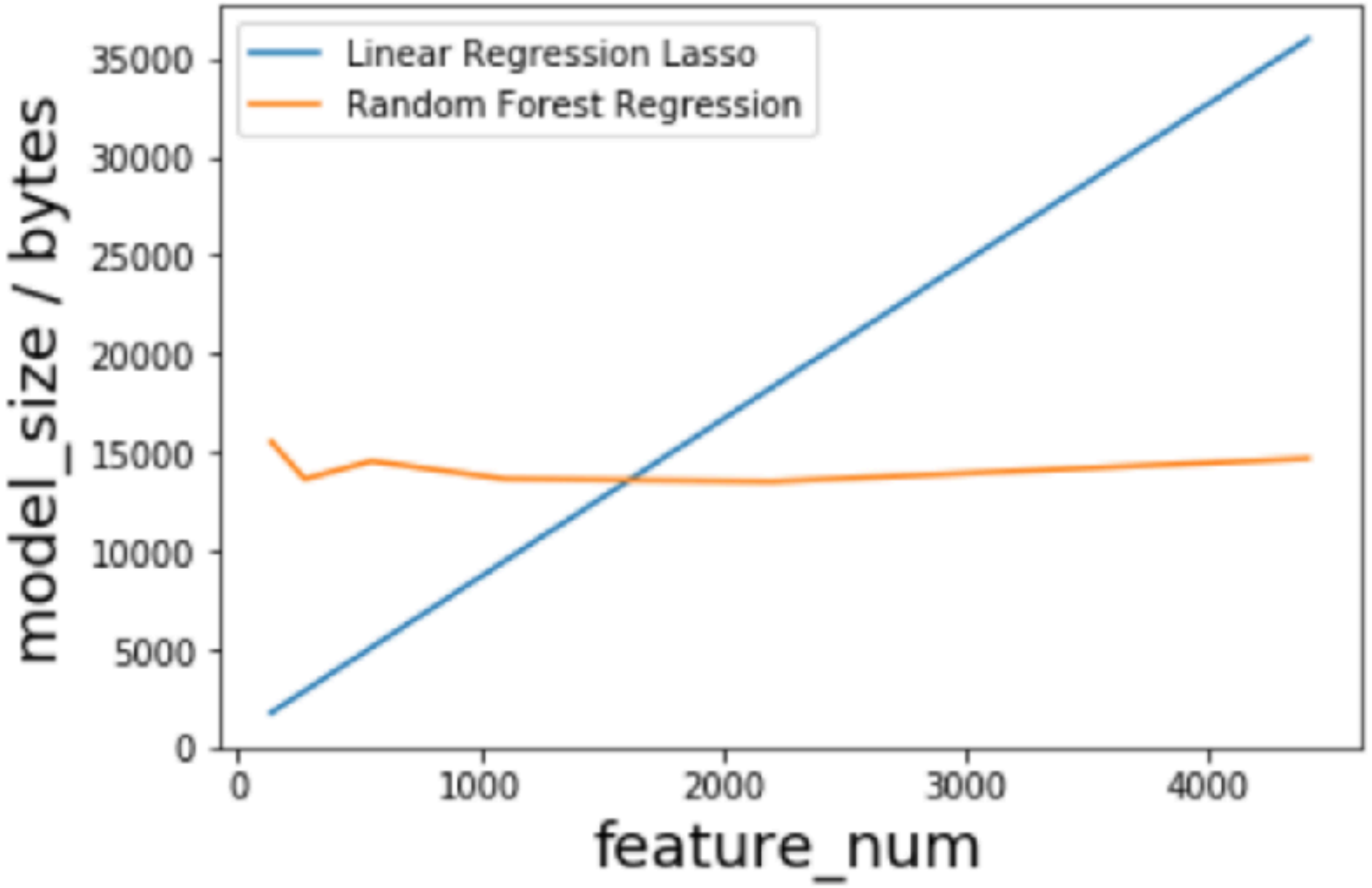
#sample



#feature

Lower: Linear Regression





Interpretability:

Better: Linear Regression

1. More intuitive visualizations for low-dimension data

2. Clearer relationship between features and parameters

3. Low interpretability for large random forests with tall trees.

Linear / Non-Linear:

Better: Random Forest

The linear regression can only capture linear relationship in the data if you do not use tricks like polynomial terms. However, for a random forest, it has the ability to detect the non-linear relationship in the data.

Recommendation - Random Forest

	Accuracy	Training / Inference Time	Model Size	Interpretibility	Linear / Non-Linear
Linear Regression (Lasso)		√	√	√	
Random Forest	√		√		√

Accuracy has a high priority

The training and inference time may not necessarily require on-the-fly speed of response. It can be done in the back-end once in a while

The model size of the models depend on different things: one is number of features in data, another is hyper-parameters.

The interpretibility and linear / non-linear are the properties of the models themselves

?

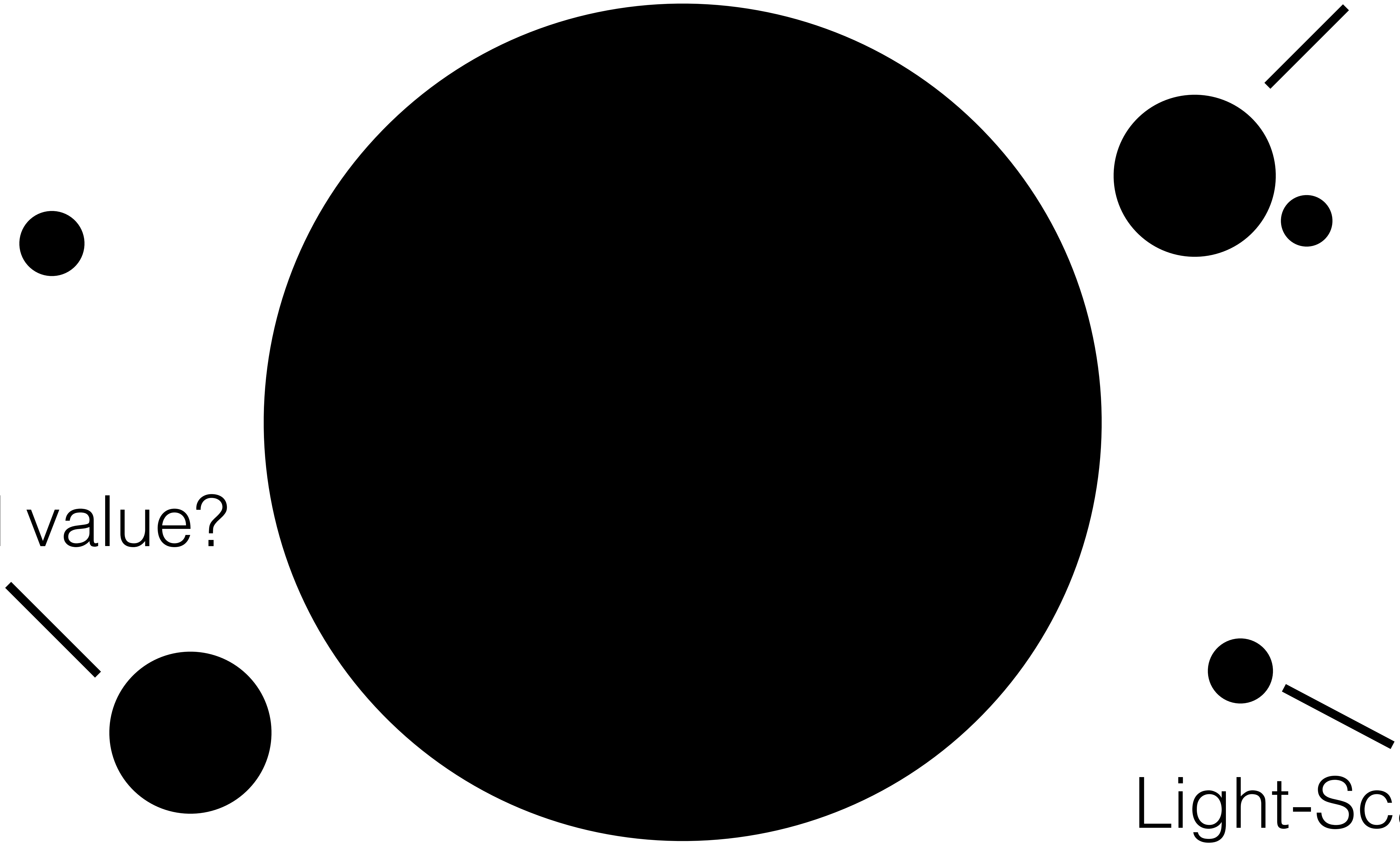
Alternative Scenario

Linear Regression is Better

Movie Rate

By

Users’ Age / Gender / Movie Genres



Consider Attributes + Constraints

This Scenario Inspired By…

User continuously watching new movies -> Model required to be **frequently trained**

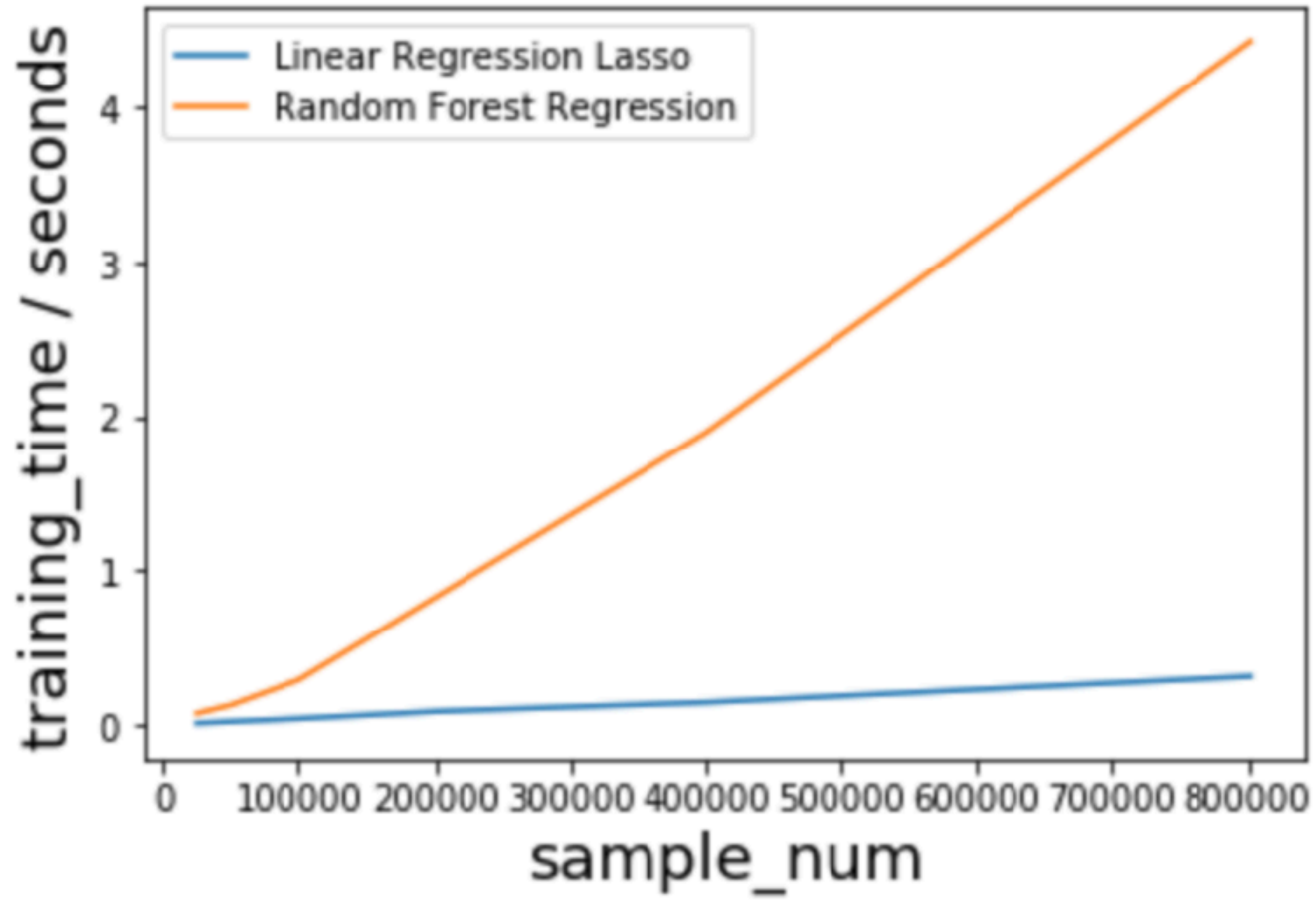
User see the recommendation result right after they visit the website (or APP) -> **Inference time** should be as **low** as unnoticeable

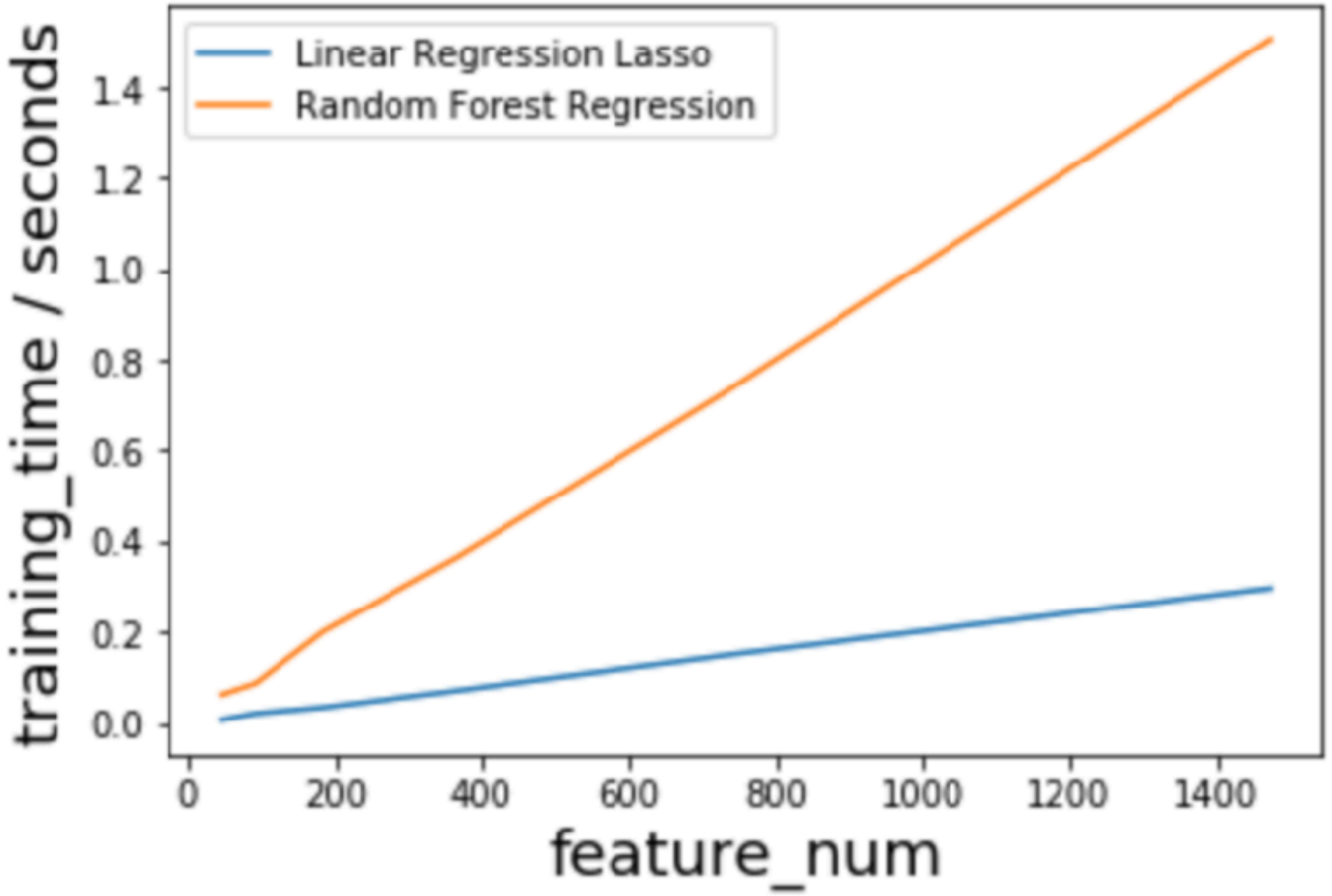
Privacy Issue -> User not want to upload their age or gender information to server -> Model may not be able to deployed on server or cloud side -> deploy on users' local device -> **light-scale, small-size** model

Output is numerical -> Movie Rate is numerical

Why Linear Regression Better?

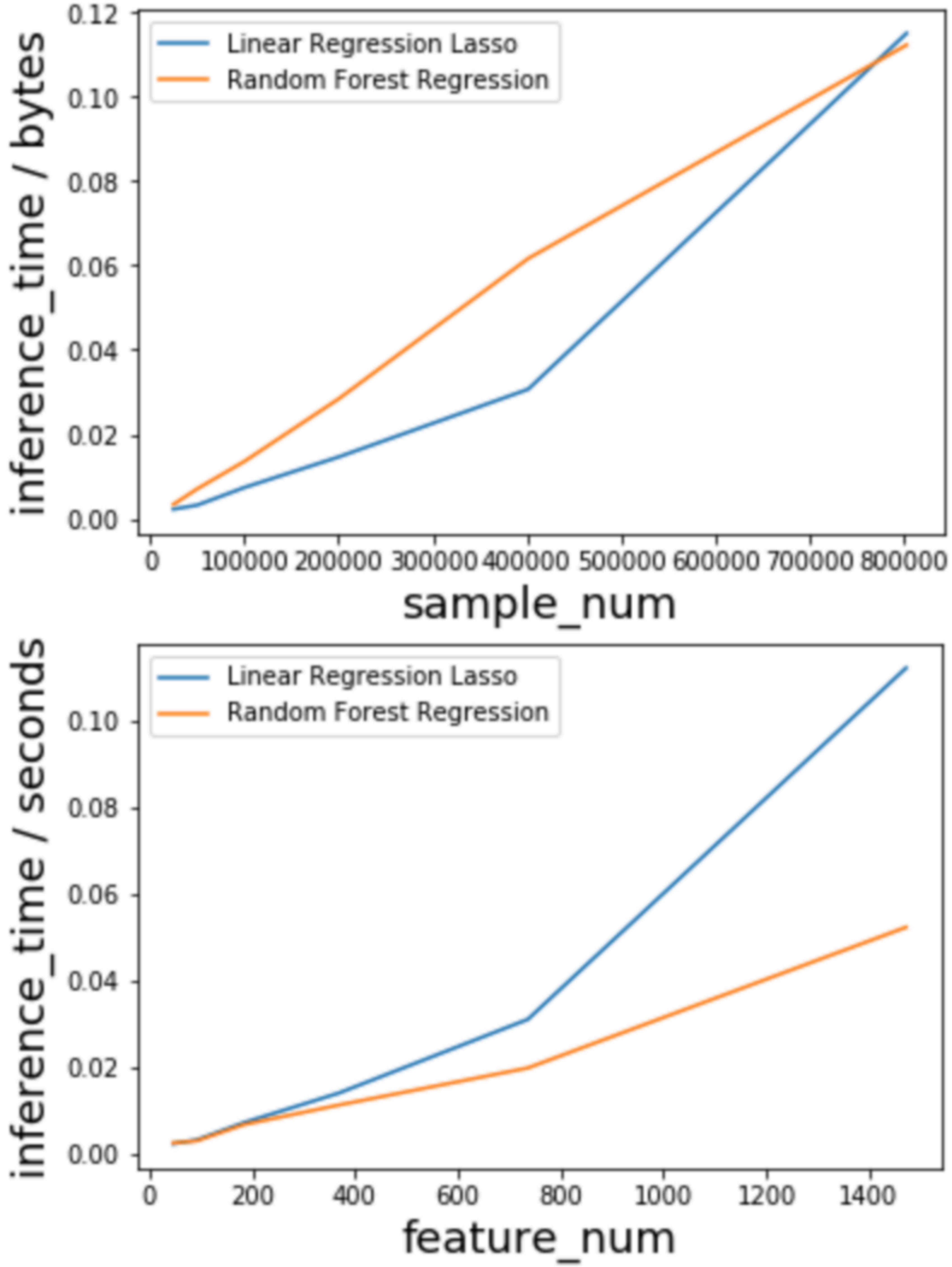
1. Low Training Cost
2. Short Inference Time
3. Small Model Size





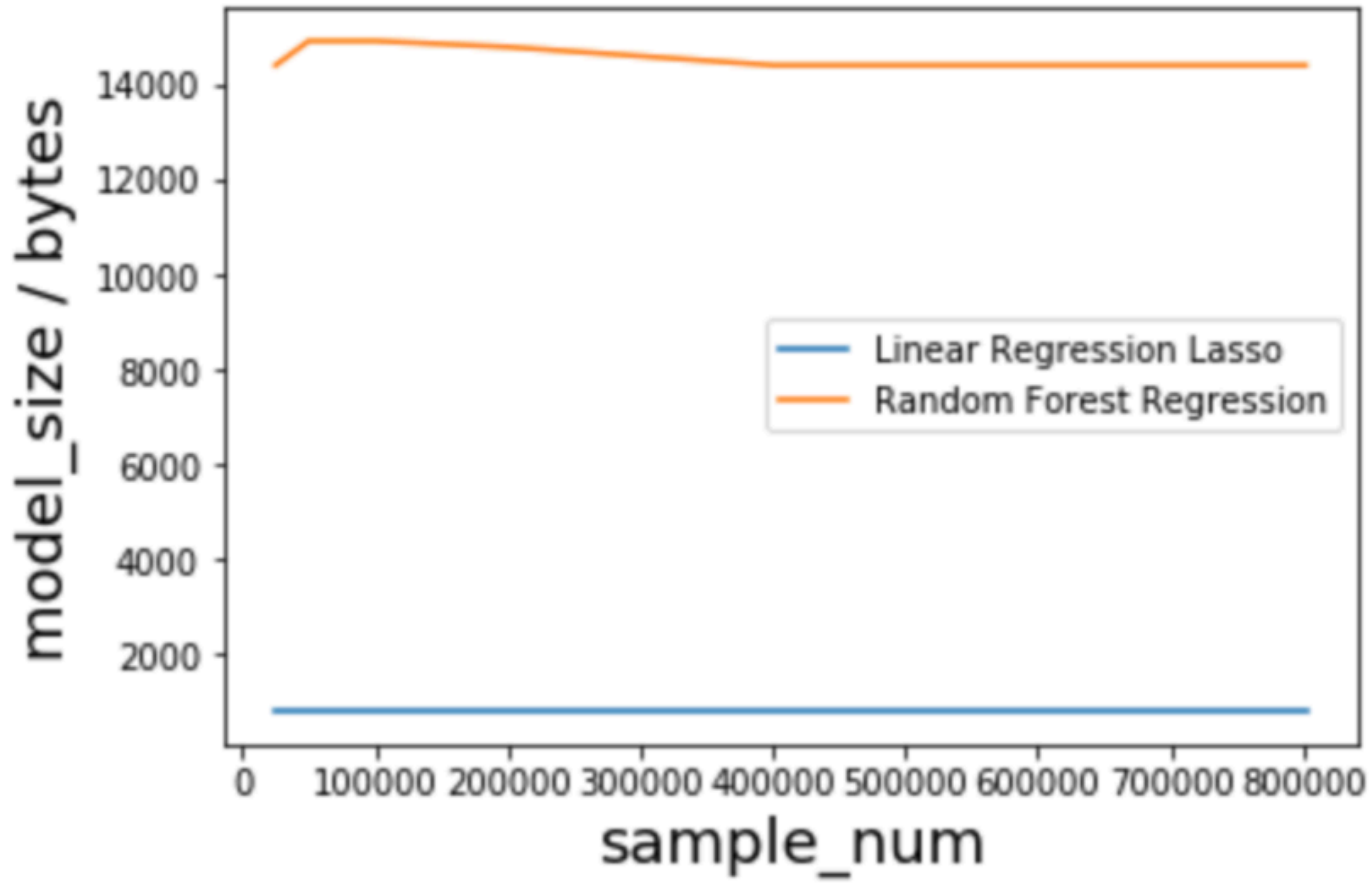
Linear Regression

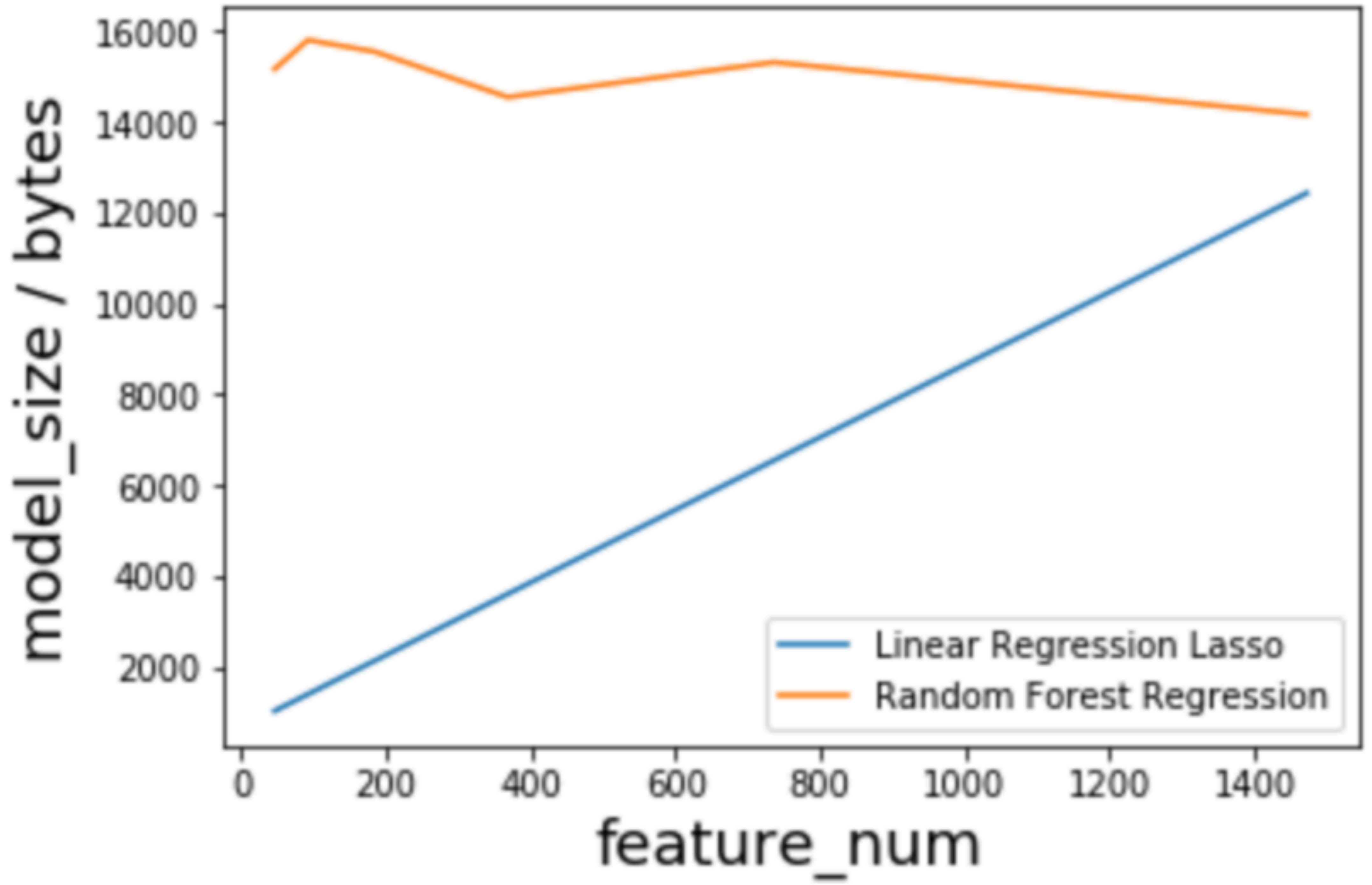
- ☑ 1.Low Training Cost
- ☐ 2.Low Inference Time
- ☐ 3.Small Model Size



Linear Regression

- ☑ 1.Low Training Cost
- ☑ 2.Low Inference Time
- ☐ 3.Small Model Size





Linear Regression

- 1.Low Training Cost
- 2.Low Inference Time
- 3.Small Model Size

Split the data into train-test split of 75% : 25%

Linear Regression Lasso MSE: 0.533825

Random Forest MSE: 0.533390

* MSE: Mean Square Error

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2.$$

Recommendation - Linear Regression

	Accuracy	Training / Inference Time	Model Size	Interpretibility
Linear Regression (Lasso)	Tie	√	√	√
Random Forest	Tie			

Short Training Time / Short Inference Time / Small Model Size has a high priority

Accuracy are almost the same

Summary

When doing AI tradeoff, we should alway consider from attributes and constraints.

- Quality Attributes
- Project Attributes
- Design Attributes
  - **ML Task**
  - **Accuracy**
  - **Training Time**
  - **Inference Time**
  - **Memory Usage**

Linear Regression

- Advantages:
  - The model size is very concise
  - Faster to run
  - Easy to understand
- Disadvantages:
  - Accuracy is relatively low
  - cannot capture non-linear relationship

Random Forest

- Randomly construct lots of decision trees
- Final output is the mode or mean of individual trees
- High accuracy and reduced overfitting, incremental
- Reduced interpretability, large number of trees can take up space

THANK YOU

Q & A