Machine Learning

Random Forests

Assignment 2

Gabor Lakatos, Gabor Zeke, Lukas Loidolt

Table of contents

Project Structure

Datasets explored

- **employee_salaries**

  Annual salary information including gross pay and overtime pay for all active, permanent employees of Montgomery County, MD paid in calendar year 2016.

- **toronto_rental**

  Toronto Apartment Rental prices from various sources in local websites

Algorithms

- **ScratchRandomForest**
- **LLMRandomForest**
- **RandomForestRegressor** (sklearn)
- **KNeighborsRegressor** (sklearn)

Implementation from scratch

- Inherits from abstract base class **`BaseRandomForest`**
- Uses **variance reduction** to determine the best split at each node
- Random selection of features for each tree
- Recursively builds decision trees in **`_build_dec_tree`**
- Parallelization via **`joblib`**
- **Splitting** continues until
  - `Max_depth` is reached
  - Number of samples is less than `min_samples`
  - No further variance reduction is possible

Implementation with an LLM

- Inherits from abstract base class **`BaseRandomForest`**
- Uses **Mean Squared Error (MSE)** reduction to determine the best split
- Random selection of features at each split
- Recursively builds the decision tree using the **`_build_tree`** method.
- **No parallelization**
- **Splitting** continues until
  - `Max_depth` is reached
  - Number of samples is less than `min_samples`
  - No split decreases the MSE.

Implementation with sklearn

- More options for **`max_features`** considered at each split: `sqrt`, `log2`, ...
- Seemingly more **optimized**
- Calculates **feature importance score**
- Hyperparameter tuning with **`GridSearchCV`**
- Supports **`warm_start`** to add trees to an already fitted model
- ...

Final algorithm: KNeighborsRegressor

- Unlike Random Forests: Requires Scaling
- Not well suited for large datasets
- Useful for smaller to medium-sized datasets
- Less "training" time – but more expensive prediction
- Simple and interpretable
- ...

Employee Salaries Dataset

Annual salary information including **gross pay** and **overtime pay** for all active, permanent employees of Montgomery County, MD paid in calendar year **2016**.

Description

- 9228 instances
  - 11169 missing values
- 13 features
  - Target: `current_annual_salary`

Preprocessing

- Splitting the `date_first_hired` column
- Imputation of missing values
  - Median imputation
  - Constant imputation for gender and job title
- One-hot-encoding for gender
- Label encoding for features with high cardinality
- For kNN: `StandardScaler`

ScratchRandomForest - Top 10 Results

trees	max_depth	min_samples	feature_subset_size	RMSE	Std. Dev.	R_squared
70	50	10	14	4718.48	28765.64	0.97
100	100	10	14	4722.83	28765.64	0.97
70	100	10	14	4763.43	28765.64	0.97
100	20	10	14	4795.35	28765.64	0.97
50	100	10	14	4815.43	28765.64	0.97
100	50	10	14	4850.88	28765.64	0.97
70	20	10	14	4854.17	28765.64	0.97
50	20	10	14	4886.53	28765.64	0.97
50	50	10	14	4903.54	28765.64	0.97
50	50	100	14	5390.63	28765.64	0.96

LLMRandomForest - Top 10 Results

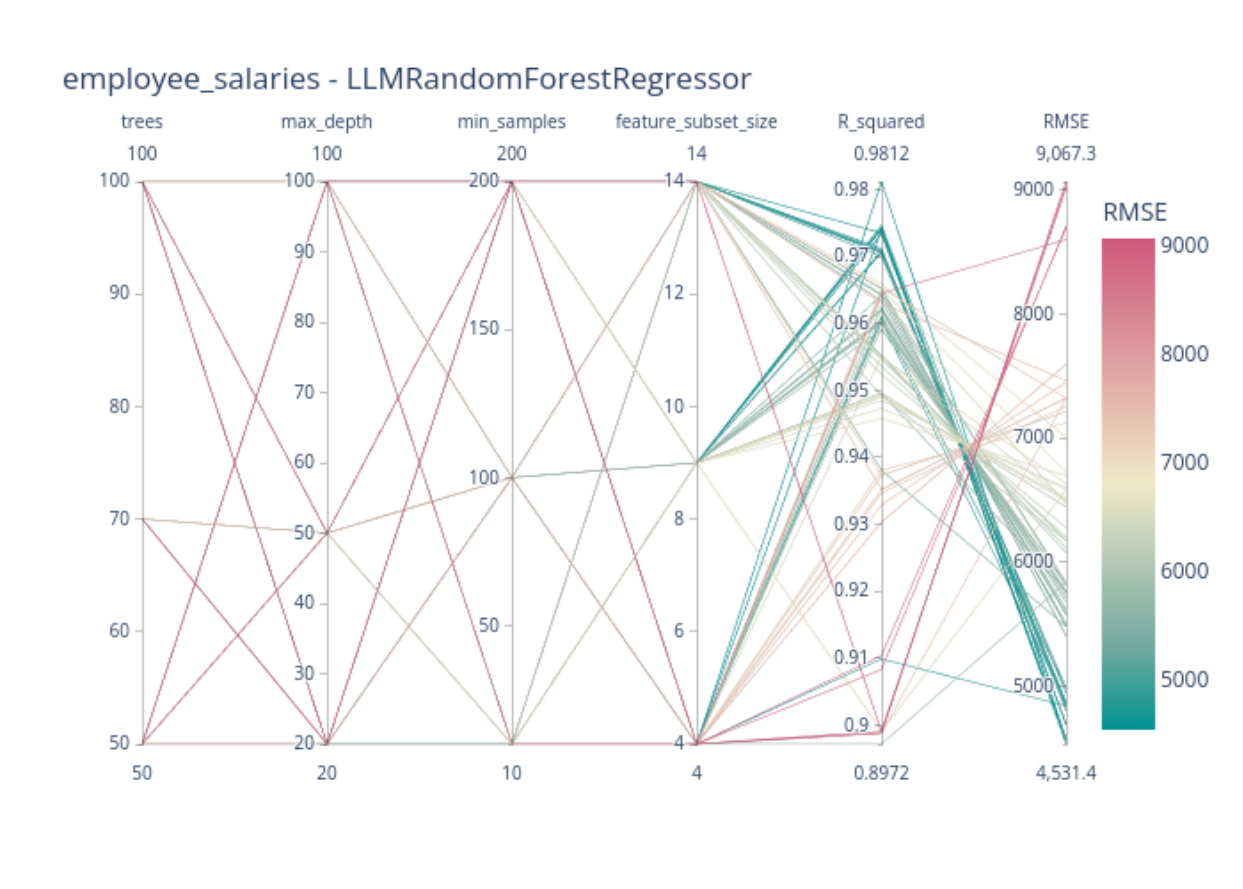
trees	max_depth	min_samples	feature_subset_size	RMSE	Std. Dev.	R_squared
100	50	10	9	4531.35	28765.64	0.97
100	20	10	9	4531.44	28765.64	0.97
70	50	10	9	4567.22	28765.64	0.97
70	20	10	9	4567.72	28765.64	0.97
100	100	200	4	4567.72	28765.64	0.98
50	100	100	4	4567.72	28765.64	0.97
50	100	10	14	4685.34	28765.64	0.97
50	50	10	9	4688.89	28765.64	0.97
50	20	10	9	4688.96	28765.64	0.97
100	100	100	9	4688.96	28765.64	0.97

RandomForestRegressor - Top 10 Results

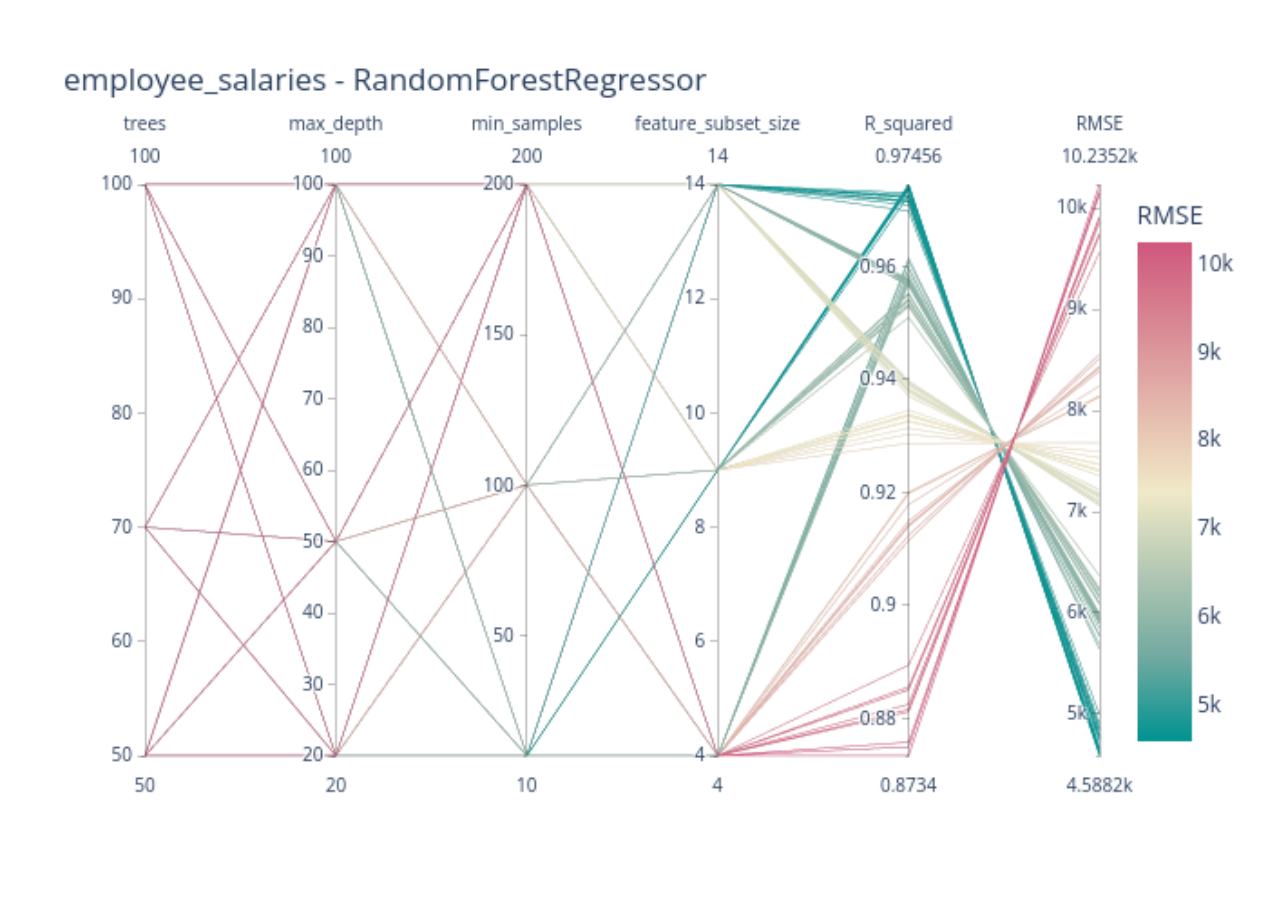
trees	max_depth	min_samples	feature_subset_size	RMSE	Std. Dev.	R_squared
70	20	10	9	4588.20	28765.64	0.97
50	50	10	9	4599.79	28765.64	0.97
50	100	10	9	4604.43	28765.64	0.97
100	100	10	9	4618.40	28765.64	0.97
100	20	10	9	4620.31	28765.64	0.97
70	100	10	9	4641.93	28765.64	0.97
100	50	10	9	4657.59	28765.64	0.97
50	20	10	9	4669.17	28765.64	0.97
70	20	10	14	4723.39	28765.64	0.97
70	50	10	9	4760.08	28765.64	0.97

ScratchRandomForest - Parallel Coordinates

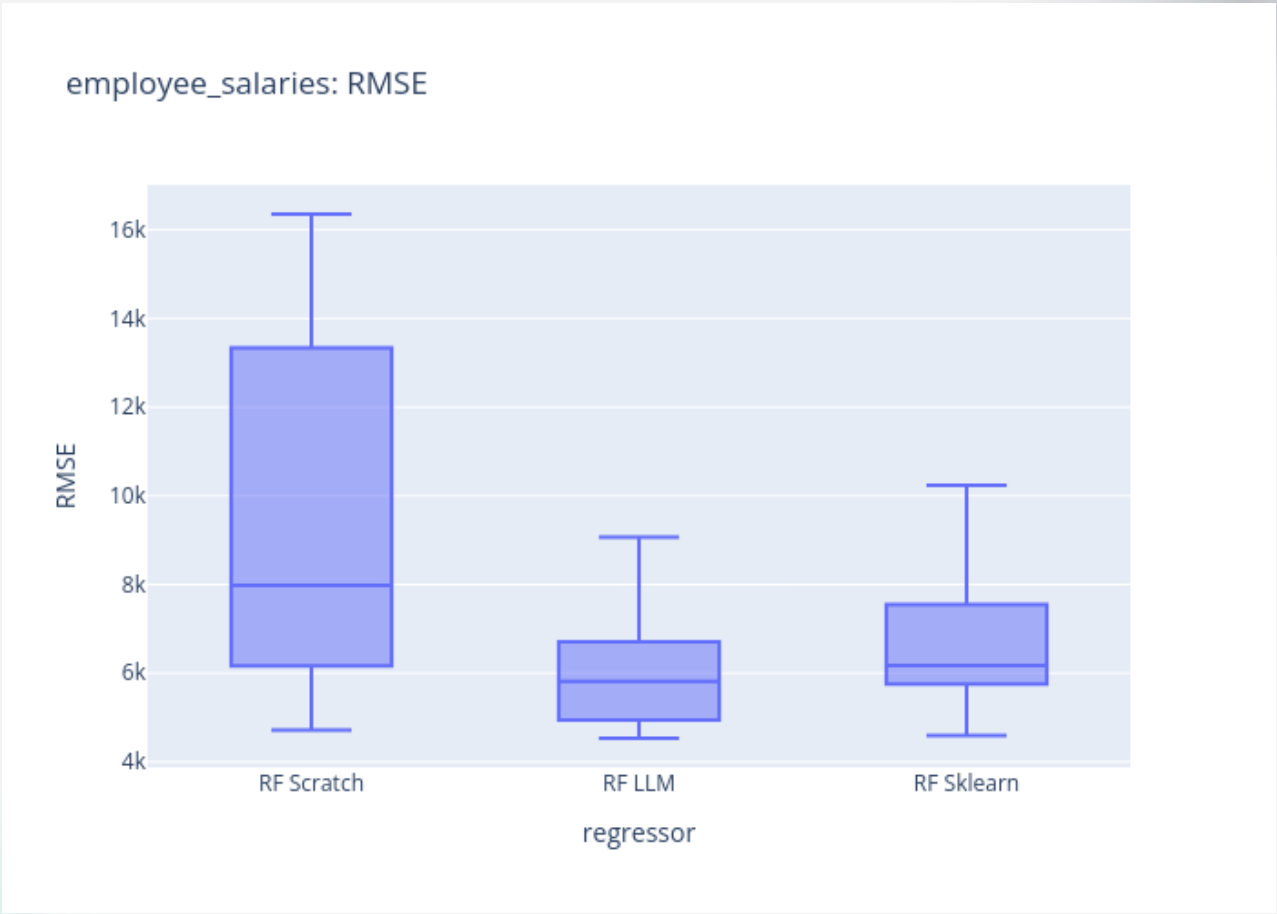
LLMRandomForest - Parallel Coordinates



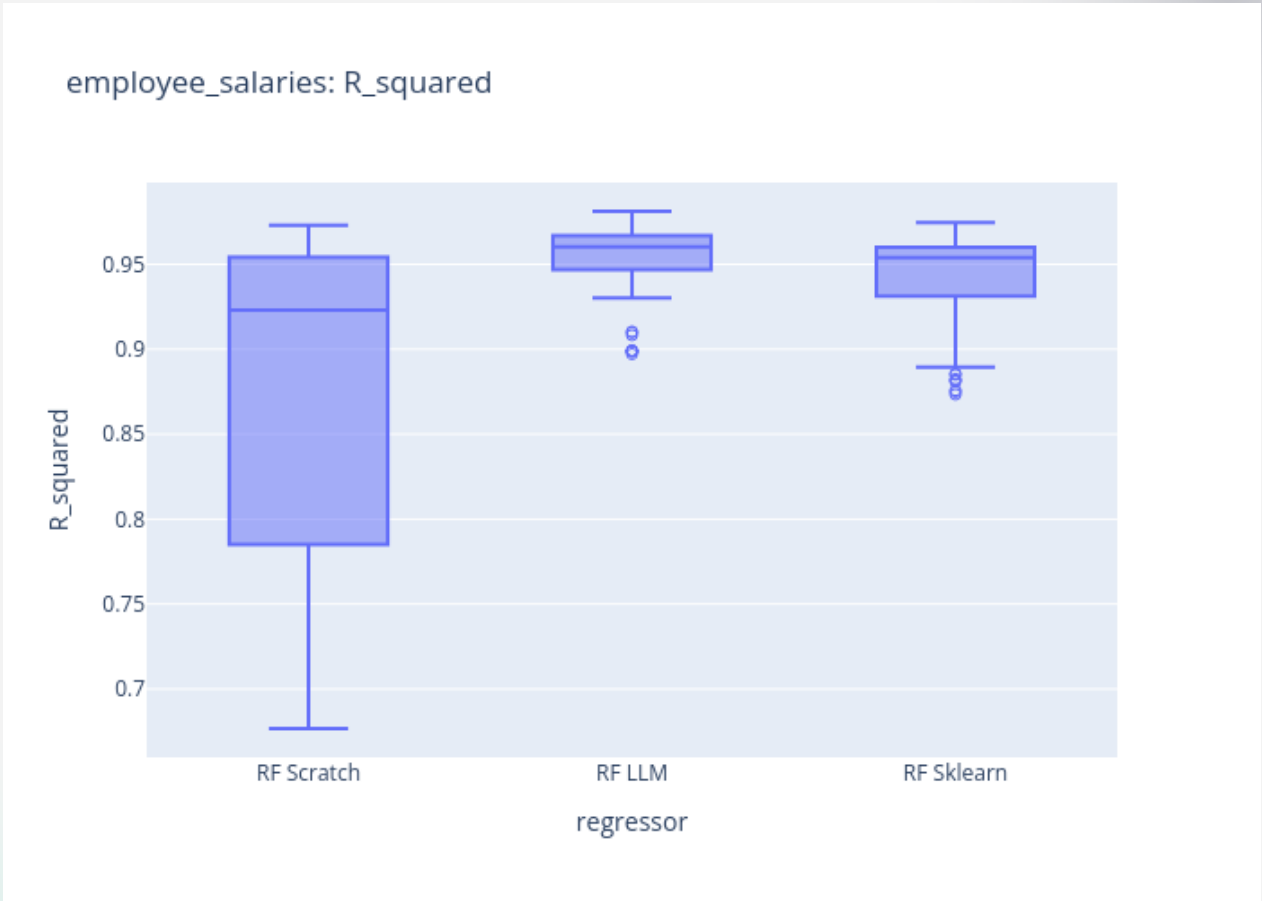
RandomForestRegressor - Parallel Coordinates



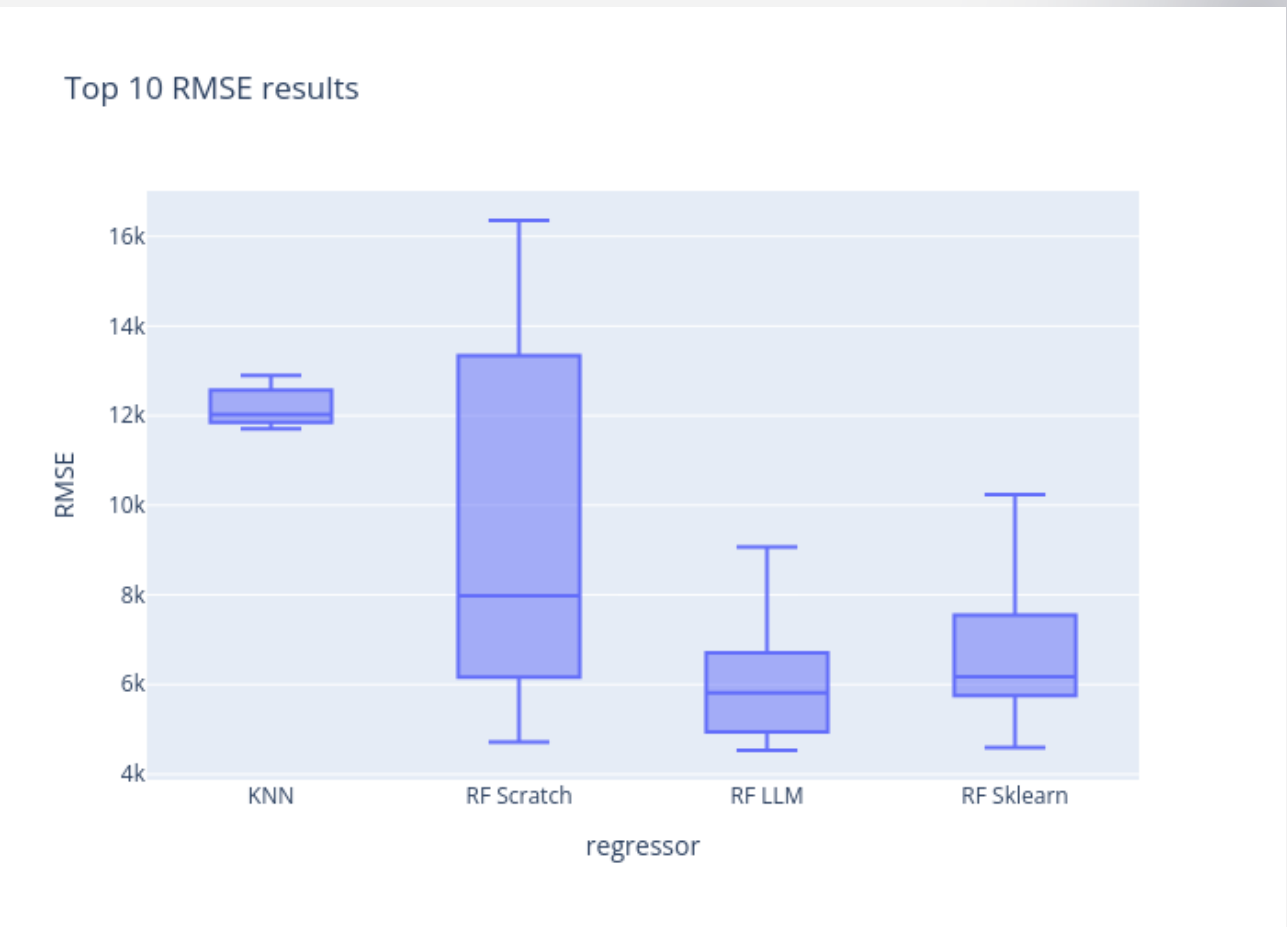
Random Forest RMSE Boxplots



Random Forest R^2 Boxplots



All algorithms RMSE Boxplots



Toronto Apartment Rental Dataset

Toronto Apartment Rental **prices** from various sources onlocal websites

Description

- 1124 instances
  - 0 missing values
- 7 features
  - Target: `Price`

Preprocessing

- Remove commas from the `Price` column
- Label encoding for features with high cardinality
- For kNN: `StandardScaler`

ScratchRandomForest - Top 10 Results

trees	max_depth	min_samples	feature_subset_size	RMSE	Std. Dev.	R_squared
70	70	10	5	5388.79	35513.74	0.98
50	20	100	5	7279.56	35513.74	0.96
100	70	100	5	7354.65	35513.74	0.96
100	20	10	5	7790.93	35513.74	0.95
50	20	10	5	7862.51	35513.74	0.95
50	70	100	5	8401.61	35513.74	0.94
100	20	100	5	8435.88	35513.74	0.94
100	50	10	5	8492.84	35513.74	0.94
70	50	10	5	8521.46	35513.74	0.94
70	70	100	5	8792.96	35513.74	0.94

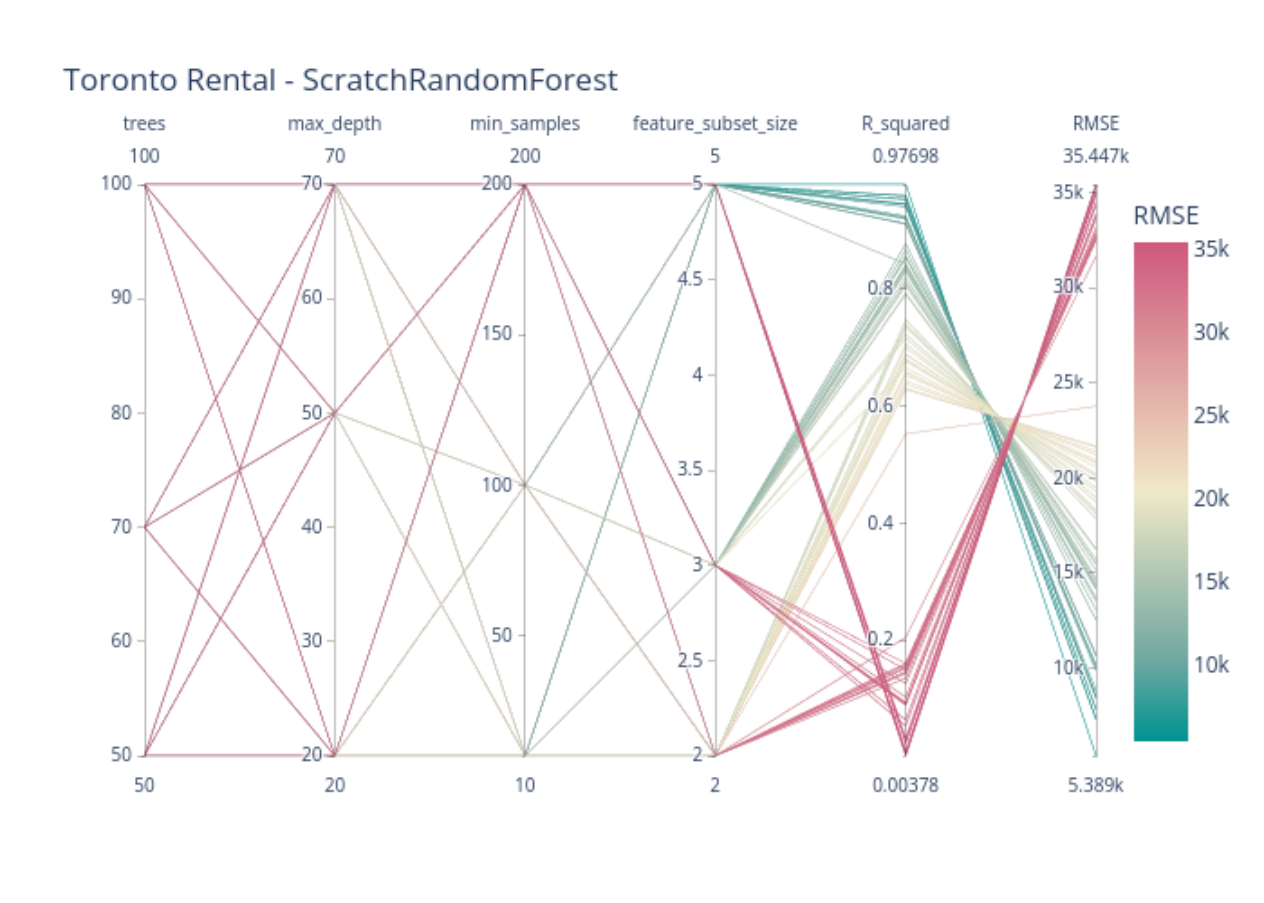
LLMRandomForest - Top 10 Results

trees	max_depth	min_samples	feature_subset_size	RMSE	Std. Dev.	R_squared
50	20	10	5	6108.82	35513.74	0.97
70	50	10	5	6591.54	35513.74	0.97
70	20	10	5	6880.24	35513.74	0.96
70	70	10	5	7601.83	35513.74	0.95
50	70	10	5	8280.82	35513.74	0.95
100	70	10	5	8281.22	35513.74	0.95
50	50	10	5	8561.72	35513.74	0.94
100	20	10	5	8727.94	35513.74	0.94
50	70	100	5	10944.80	35513.74	0.91
50	50	100	5	10982.72	35513.74	0.90

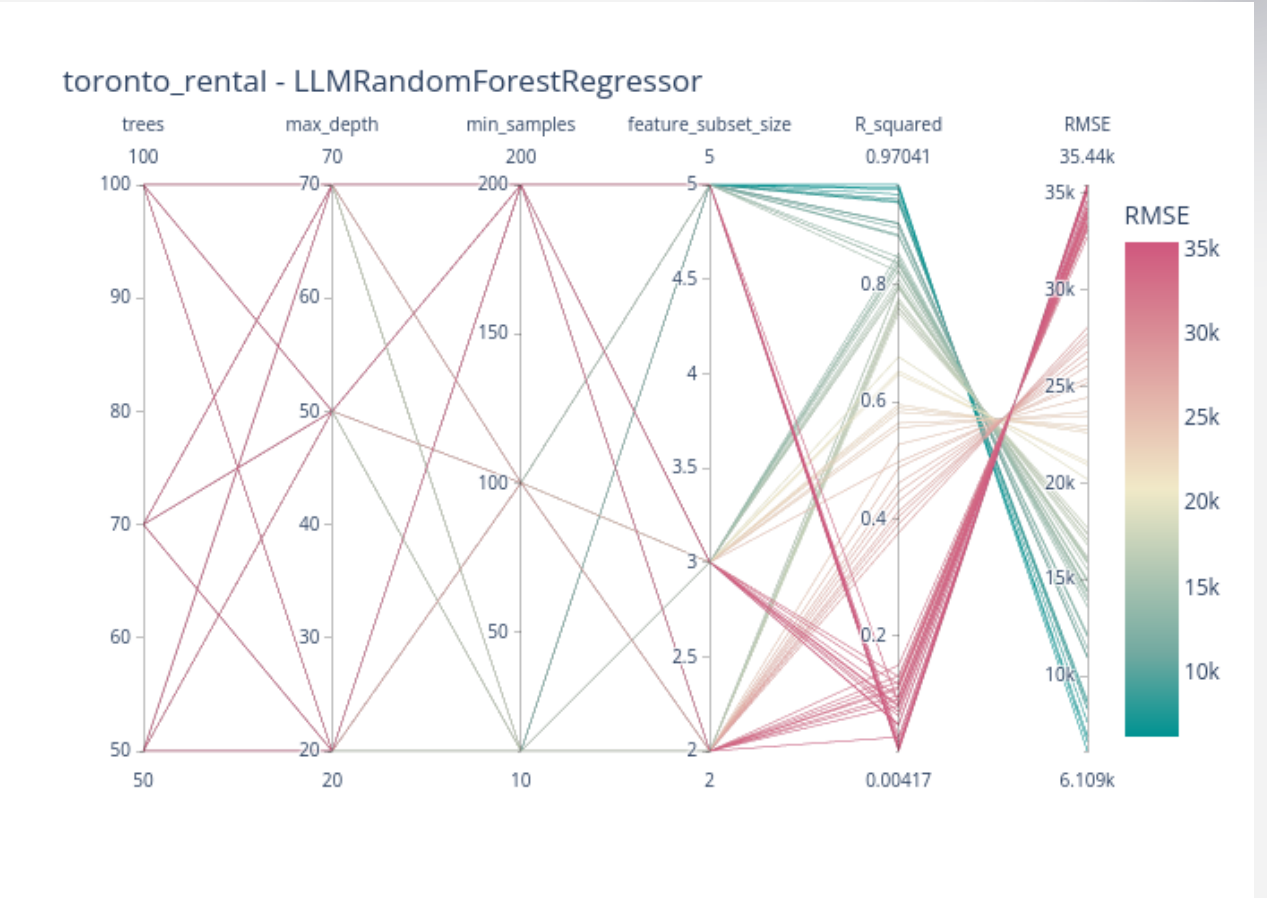
RandomForestRegressor - Top 10 Results

trees	max_depth	min_samples	feature_subset_size	RMSE	Std. Dev.	R_squared
70	20	10	5	7431.48	35513.74	0.96
50	70	10	5	7637.14	35513.74	0.95
70	70	10	5	9200.48	35513.74	0.93
100	70	10	5	9415.34	35513.74	0.93
50	50	10	5	9799.13	35513.74	0.92
100	20	10	5	10003.61	35513.74	0.92
100	50	10	5	10208.10	35513.74	0.92
50	20	10	5	10631.56	35513.74	0.91
70	50	10	5	10704.94	35513.74	0.91
50	50	100	5	13443.22	35513.74	0.86

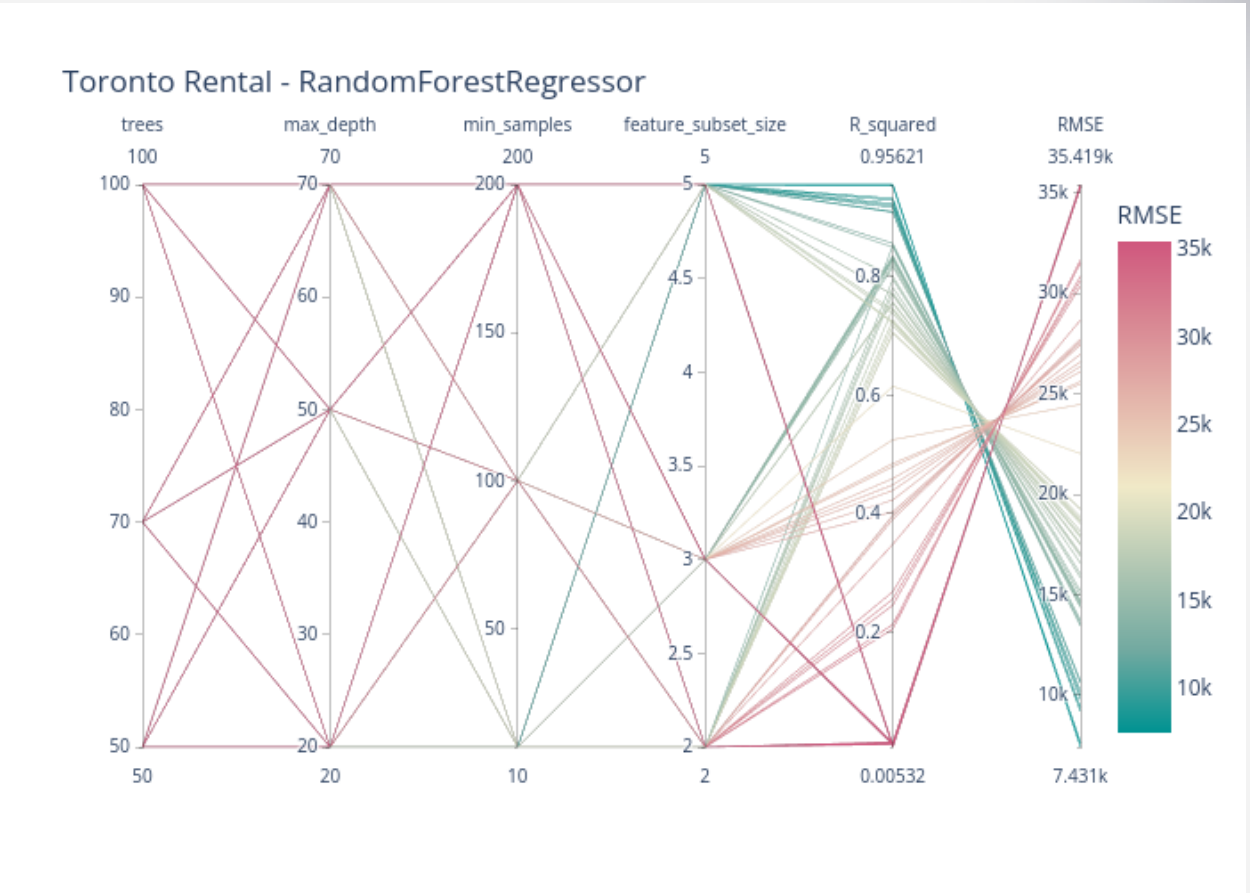
ScratchRandomForest - Parallel Coordinates



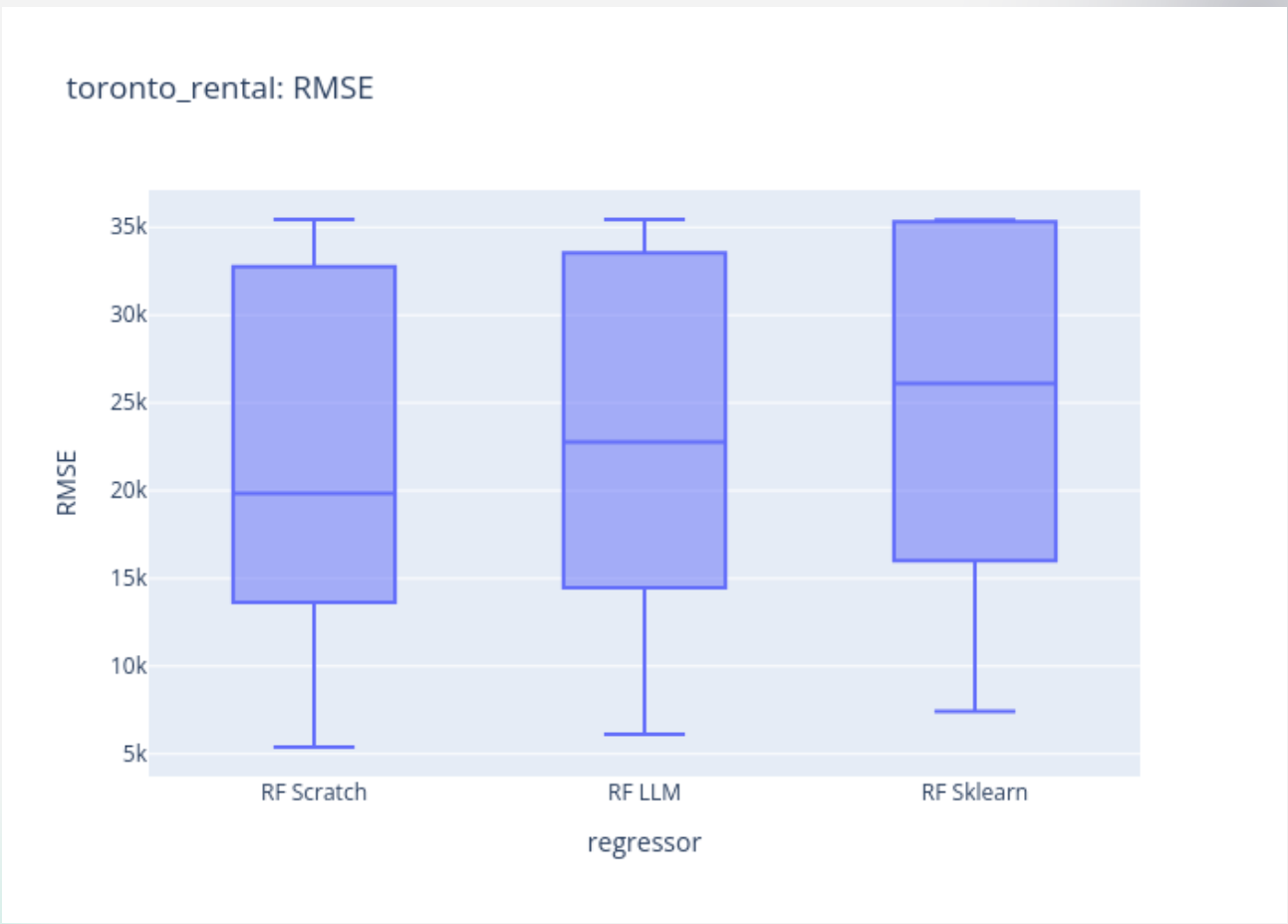
LLMRandomForest - Parallel Coordinates



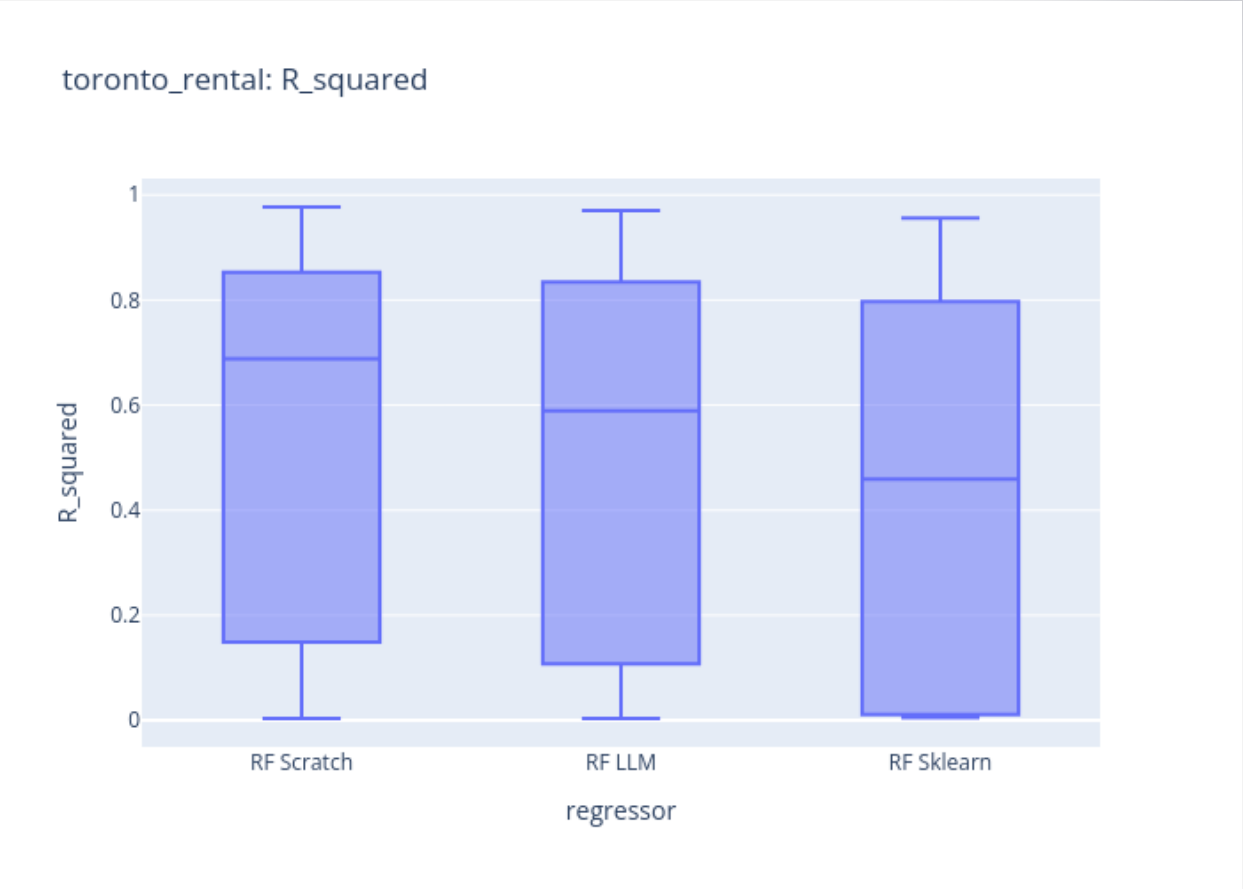
RandomForestRegressor - Parallel Coordinates



Random Forest RMSE Boxplots



Random Forest R^2 Boxplots



All algorithms RMSE Boxplots

Conclusions

- Generally: Similar(-ish) performance between the three random forest implementations
- Interestingly, `ScratchRandomForest` performed slighly better on the small `Toronto Rentals`, but slighly worse on the larger `Employee Salaries` dataset
- LLM implementation worked well, but lack of parallelization is non-ideal
- `Sklearn` implementation the most optimized
- KNN lagged behind RandomForest for the larger `Employee Salaries` dataset, but was competitive or even superior on the smaller `Toronto Rentals` dataset