Report

KSAT Quest: Regression Runoff

Team : NERDY_POTATOES

Dataset Description:

The dataset contains multiple sheets (e.g., "Ref #1", "Ref #2", etc.) each representing different studies or sample sources. The sheets share a common structure and include:

- Physical and chemical soil properties
- Metadata like method, source, and location
- Measured saturated hydraulic conductivity (`Ksat`)
- Various units used for `Ksat`, e.g., `cm/hr`, `mm/h`, `in/hr`, etc.

Target Variable : The Target variable in our dataset is Ksat (Saturated Hydraulic Conductivity). It is the ease with which pores of a saturated soil transmit water. Formally, it is the proportionality coefficient that expresses the relationship of the rate of water movement to hydraulic gradient in Darcy's Law.

---

1. Data Preprocessing and Cleaning

- Sheet Filtering: Only "Ref #" sheets were included to ensure consistency in the dataset.
- Column Standardization: Column names were unified to lowercase, removing spaces for easy manipulation.
- Consolidation: Sheets were merged into one DataFrame with only shared columns to focus on relevant data.

- Missing Values: Columns with >80% missing data were dropped; remaining gaps were handled using median (numeric columns) or mode (categorical columns). This reduced noise while preserving core information.
- **Unit Conversion**: Normalized Ksat values into cm/hr using a dictionary of conversion factors. Rows with missing or invalid units were excluded to maintain data integrity.

---

2. Feature Selection

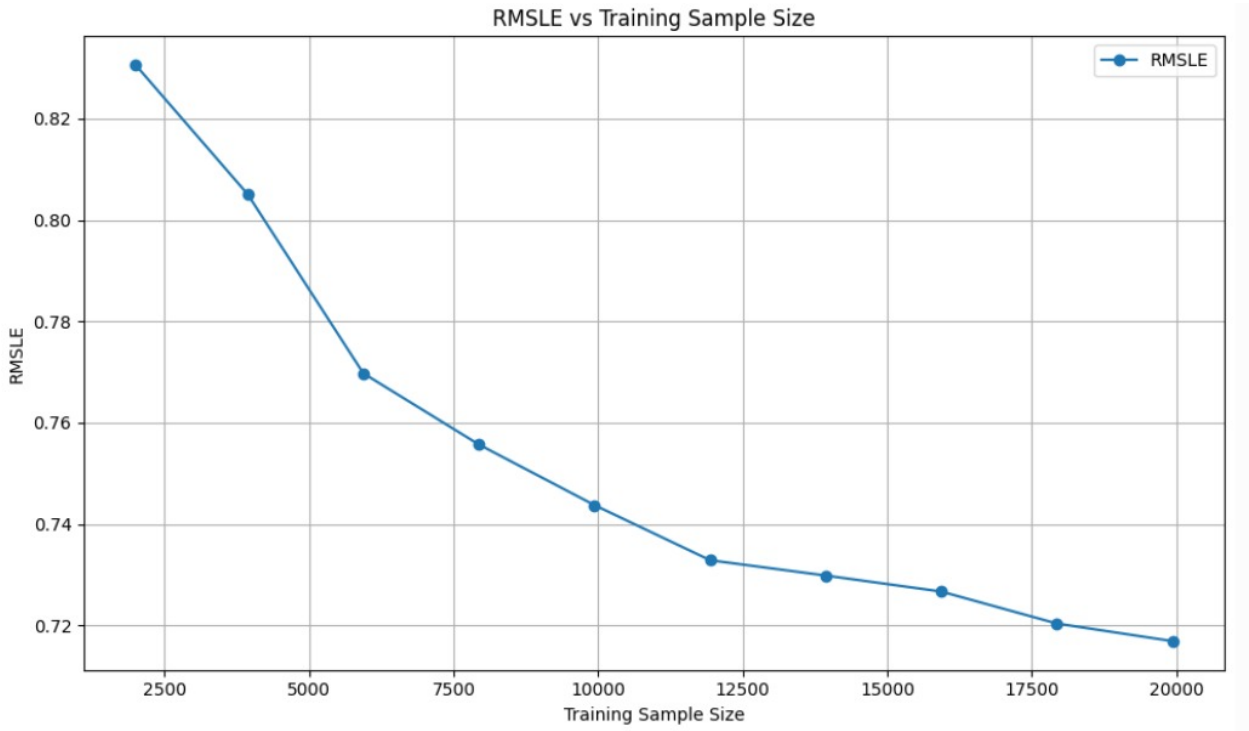
1. Numeric Features:
    - A correlation matrix was calculated between numeric features and the target variable (ksat)
    - Features with an absolute correlation value ≤ 0.1 were considered weakly related to the target and dropped
    - The column sourcereference was manually dropped.
2. Categorical Features:
    - Key categorical features (field/lab, method , and textural class) were retained.
    - Rare categories in (fewer than 10 samples) were grouped into "Other" to prevent overfitting.
    - These features were normalized (e.g., converting values to lowercase) and encoded using one-hot encoding

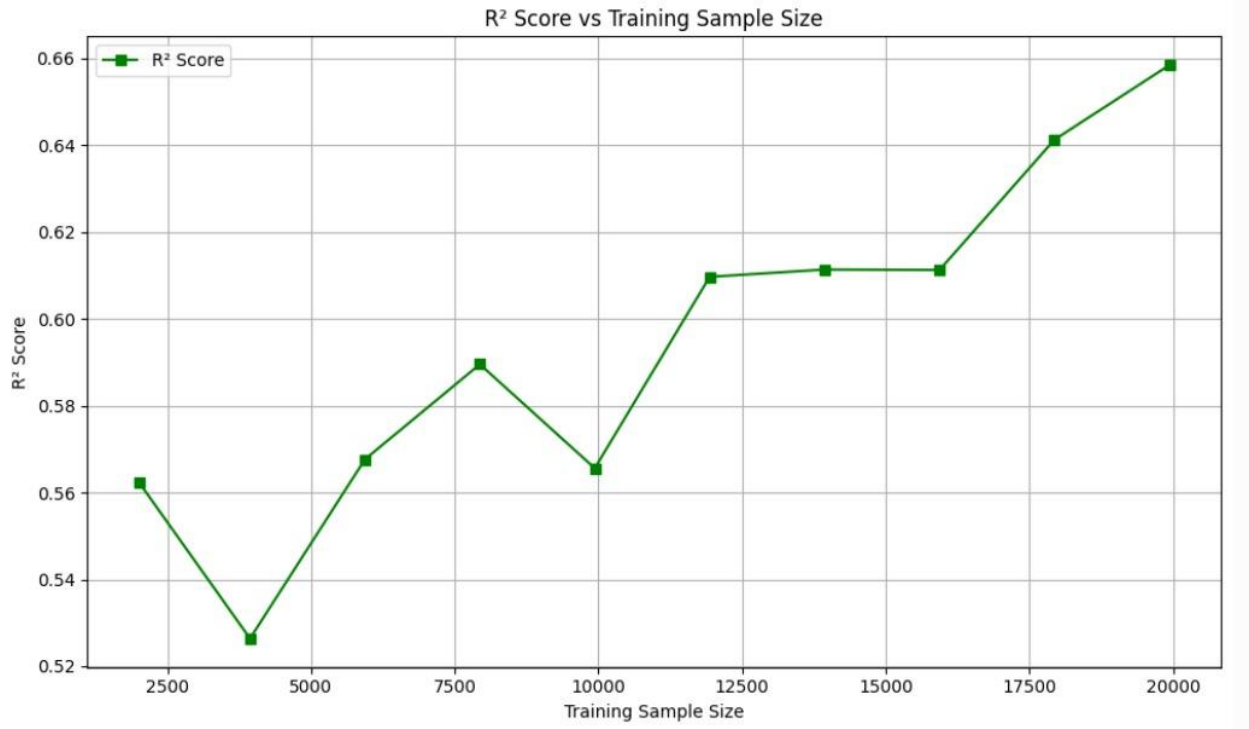
By focusing on relevant predictors and removing noisy or redundant features, the feature selection process ensured the dataset was lean and ready for accurate modeling

3. Model Development

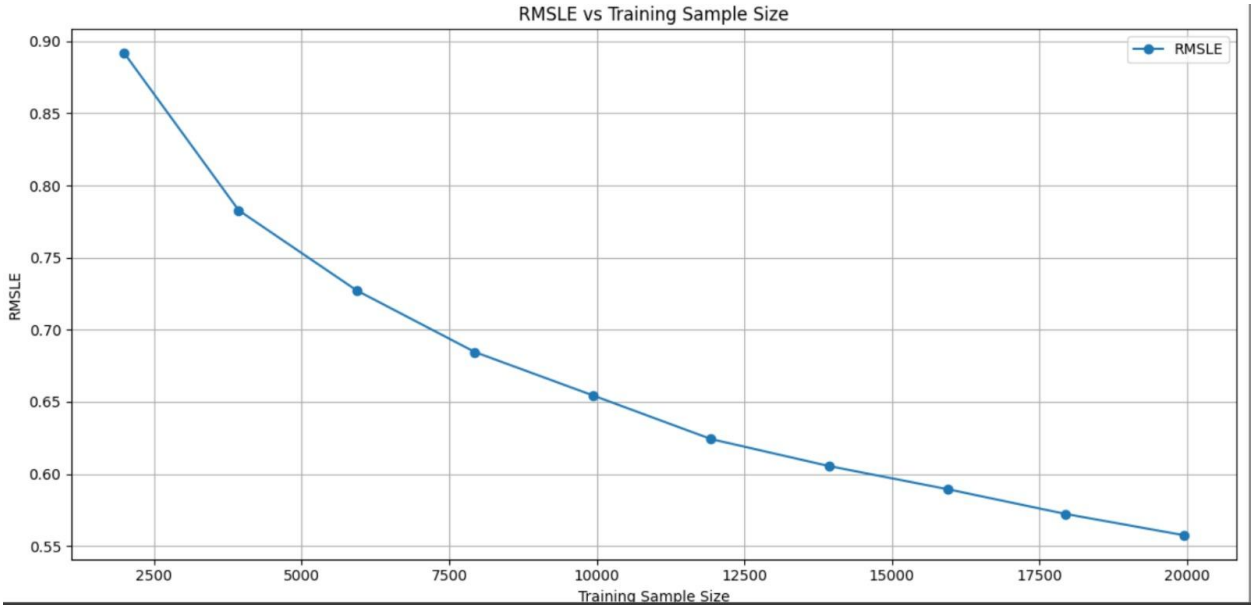
We ran the dataset in 2 different models, Random Forest and LightGBM Regressor. We found that the RandomForest Regressor model works better for this dataset. Please find below the plots of both Random Forest and LightGBM models.

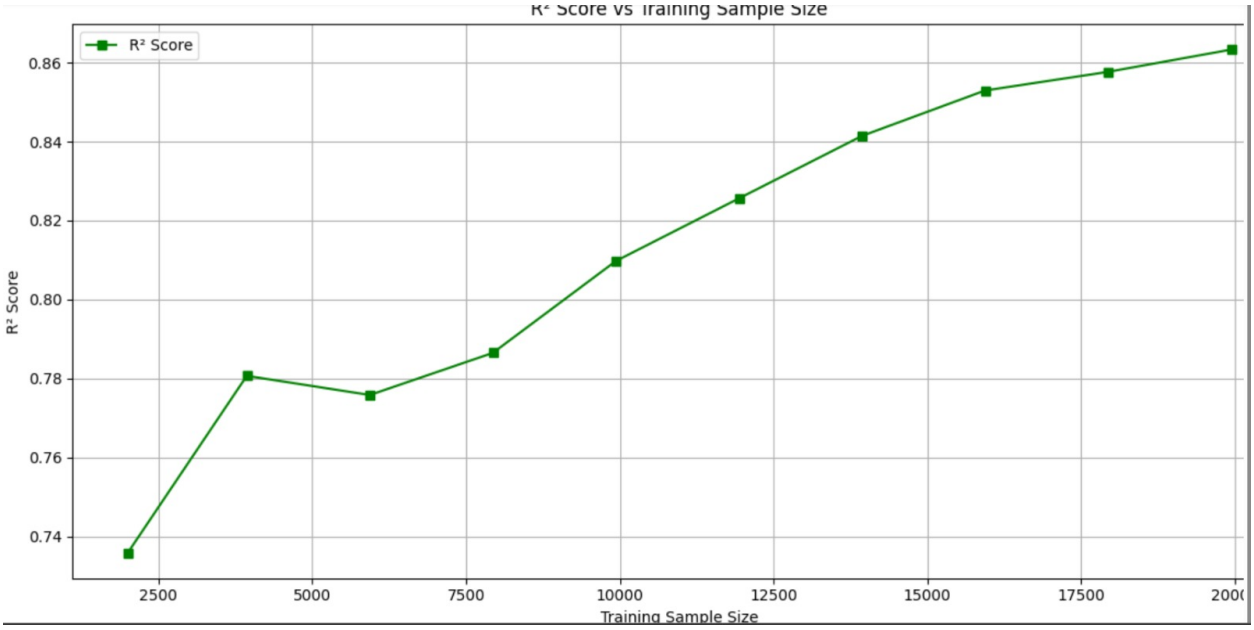
LightGBM :


RMSLE vs Training Sample Size
RMSLE
0.82
0.80
0.78
0.76
0.74
0.72
RMSLE
2500
5000
7500
10000
12500
15000
17500
20000
Training Sample Size


R² Score vs Training Sample Size
R² Score
0.66
0.64
0.62
0.60
0.58
0.56
0.54
0.52
R² Score
2500
5000
7500
10000
12500
15000
17500
20000
Training Sample Size

Random Forest :





- Choice: Random Forest Regressor is selected for its speed, scalability, and ability to handle both numeric and categorical data efficiently.

- Split: Data was divided (80% training, 20% testing) to ensure unbiased evaluation.
- Metrics: R^2 and RMSLE were chosen to measure performance in explaining variance and handling outlier impacts effectively.

---

4. Hyperparameter Tuning

- **RandomizedSearchCV:** Explored the hyperparameter space using predefined ranges for key parameters, including: 'n_estimators', 'max_depth', 'max_features', 'learning_rate', 'min_samples_leaf','min_samples_split'.
- Hyperparameter combinations were optimized based on the R^2 score during cross-validation.
- Best hyperparameters found: {'n_estimators': 200, 'min_samples_split': 2, 'min_samples_leaf': 1, 'max_features': 'sqrt', 'max_depth': None}

---

5. Subset Experimentation

- Ranged from the full dataset size down to 2000 samples, decreasing in increments of 2000
- For each subset size, the experiment was repeated 50 times with random sampling to ensure results are robust and not biased by specific splits.
- Used optimal configurations (determined via RandomizedSearchCV) consistently across all subset sizes.

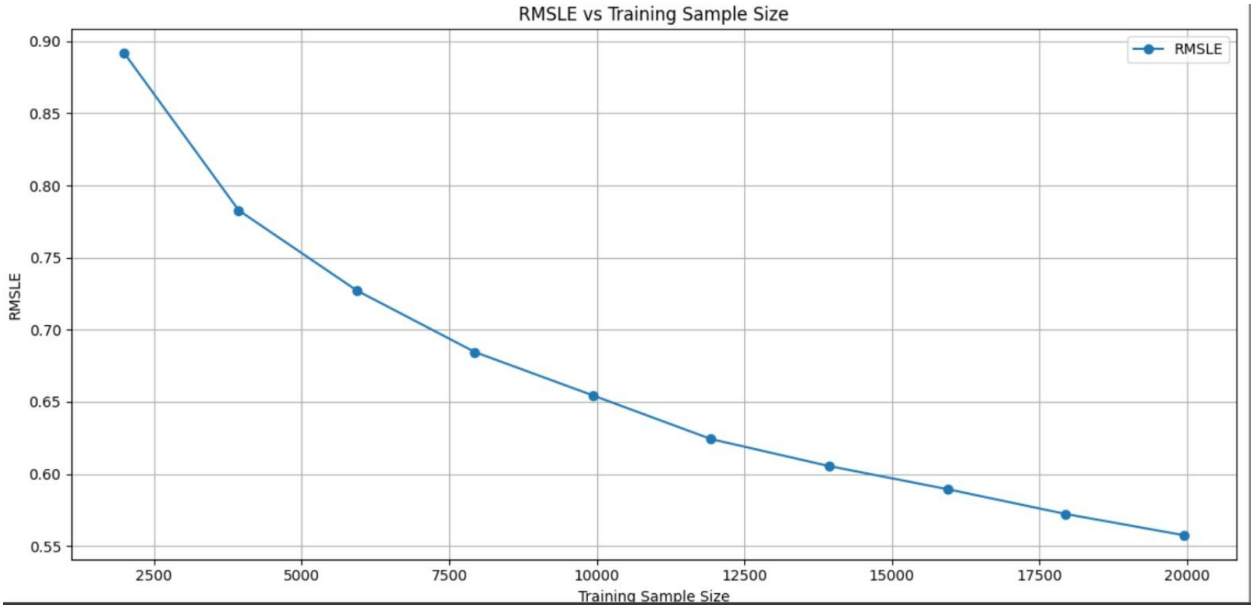
---

6. Evaluation Metrics

- RMSLE: Assess predictive accuracy.
- R^2: Measure the variance explained.

7. Visualization

- RMSLE vs sample size

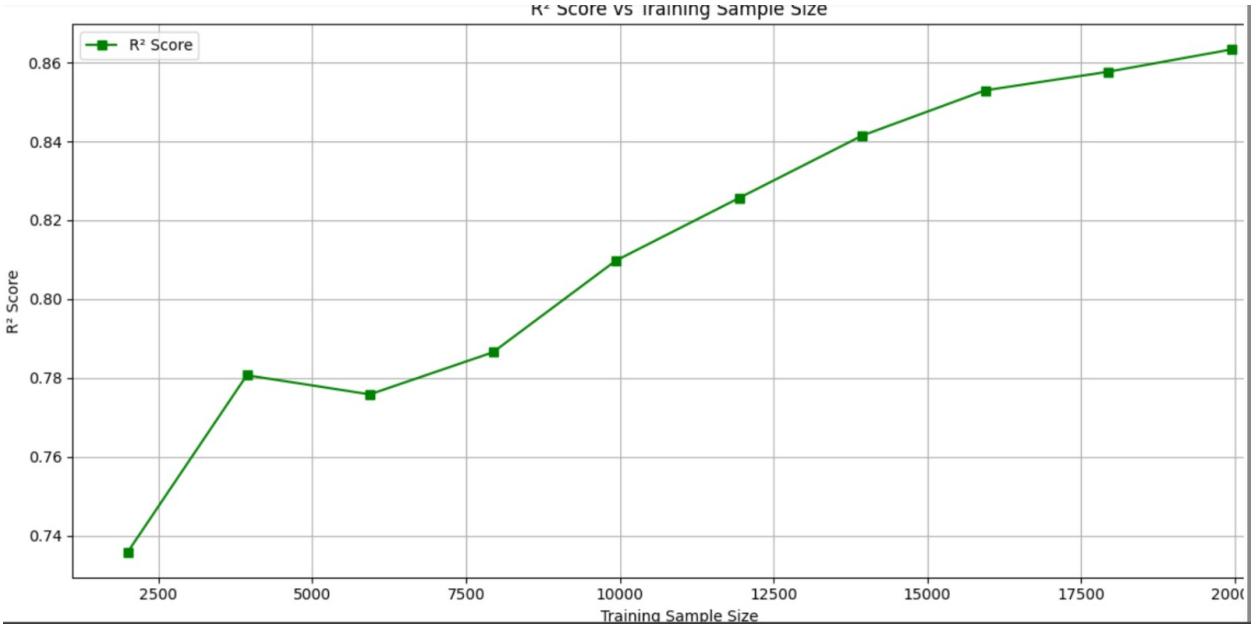


The plot clearly shows a monotonically decreasing RMSLE as training sample size increases. This indicates that the model is benefiting significantly from having more data, with performance (in terms of RMSLE) improving consistently across all tested subset sizes.

Notably:

- The steep drop in RMSLE from 2,000 to 10,000 samples suggests **high data efficiency** in that range.
- Beyond ~15,000 samples, the RMSLE still decreases but at a **diminishing rate**, implying **approaching saturation** of model learning capacity.
- This trend reflects the model's capability to generalize better with more data, while also hinting at the potential upper bound of performance with this feature set and model type.

- R^2 vs sample size



The plot illustrates a generally **increasing R^2 score** with training sample size, confirming that the model's predictive power improves as more data becomes available.

Observations:

- There's **initial variance** at smaller sample sizes (2,000–5,000), likely due to instability from limited training data or sensitivity to subset randomness.

- From 6,000 onward, R^2 steadily climbs, showing a clear **positive correlation between data volume and explained variance**.

- The model reaches an R^2 of about **0.645 at 20,000 samples**, indicating decent fit, though not perfect — room still exists for improvement via feature engineering or model tuning.