BLUEBERRY YIELD PREDICTION

KSHITIZ SHAH & ZHANGBO CHENG

The Business Challenge

We want more blueberries!

Indulging in the delightful taste of blueberries, from pastries to milkshakes and jams, has made us curious about the art of blueberry cultivation. We embarked on a quest to uncover the key factors influencing blueberry yield, aiming to empower prospective farmers by providing accurate estimates of their crop..

BLUF

We propose to use an XGBoost regressor trained on 3 features: 'fruitset', 'seeds', and 'fruitmass' to predict yield with 82% accuracy.

EDA and Pre-Processing

No null values or duplicates. Skewness and outliers fixed through transformations and tukeys rule

Metrics Used

MSE and R^2 : to make accurate predictions by penalizing large errors and to decide models based on explained variance

Hyperparameter Tuning and CV

Parameters: learning_rate: 0.1, max_depth: 3, n_estimators: 100

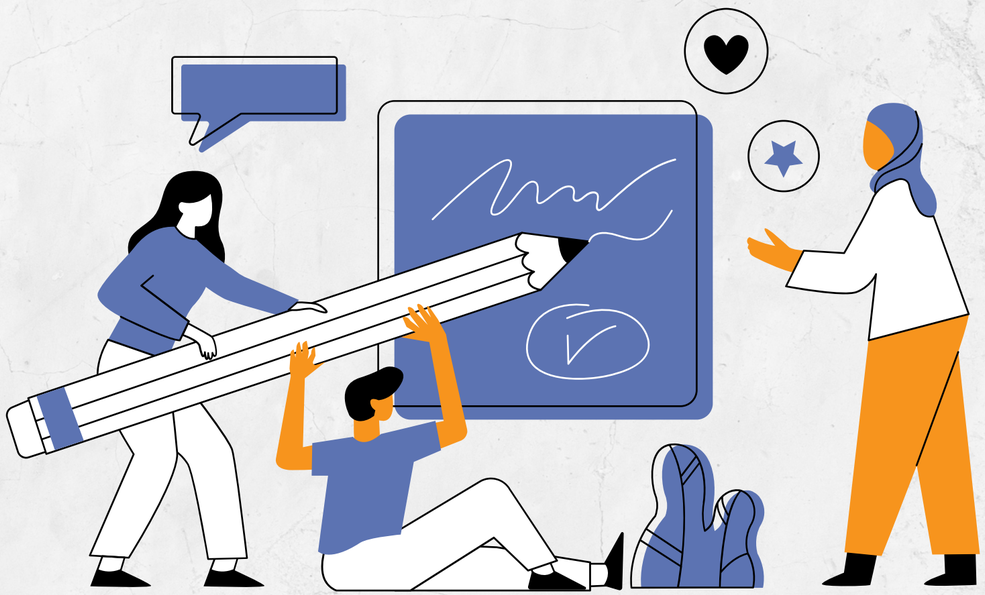
GAME PLAN

Low code solution
Pycaret

No code solution
H20.ai / aquarium

Regressors and Ensembles
Horse races to determine the winner

We wanted to explore all options to decide what gives us the most bang for our buck. The objective was to find a non taxing, simple model that is easily explainable to the end user: farmers growing blueberries



EDA : Data Description

A detailed description of all features of the dataset can be found in our notebook.
Our dataset had 18 features and 15,289 observations. The dataset was sourced from Kaggle. We performed an 80/20 split on our data to generate training and test sets.
We did not have to perform any imputations etc. on our data since the only issues were skewness and outliers which were fixed using tukey's rule and transformation.

→Pollination features:

- honeybee
- bumbles
- andrena
- osmia

→ Temperature and weather features:

- MaxOfUpperTRange
- MinOfUpperTRange
- AverageOfUpperTRange
- MaxOfLowerTRange
- MinOfLowerTRange
- AverageOfLowerTRange
- RainingDays
- AverageRainingDays

→ Plant features:

- id
- clonesize
- fruitset
- Fruitmass
- seeds
- yield

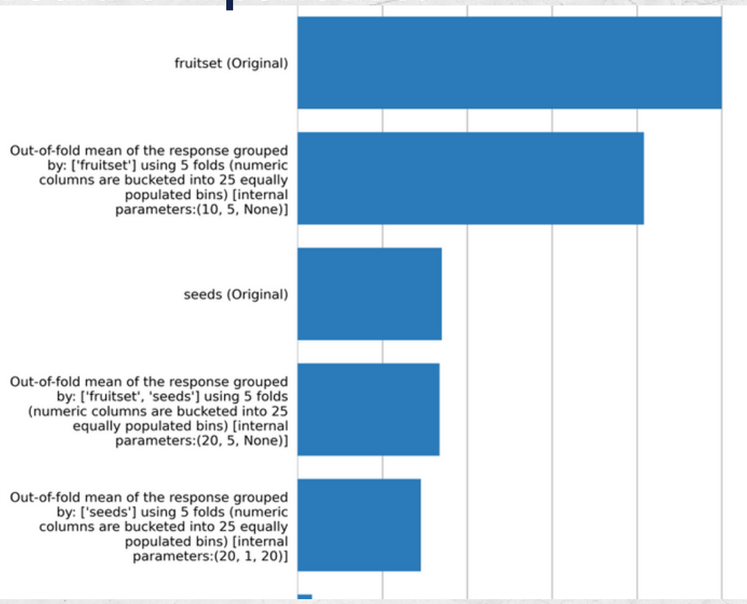
Solution 1: Driverless H20.ai

Generated a stacked ensemble of 1 LightGBMModel, and 1 XGBoostGBMModel to predict yield given 17 original features from the input dataset train.csv. This regression experiment was completed in 54 minutes and 29 seconds (0:54:29), using 16 of the 17 original features, and 21 of the 21 engineered features

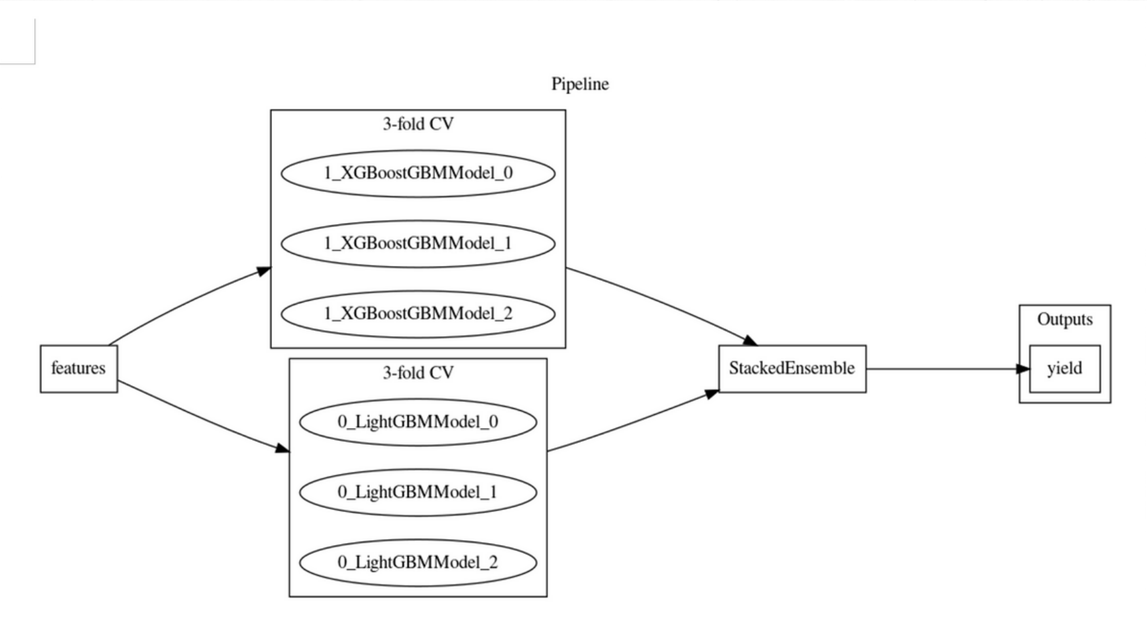
Result: MSE :318920.781

Control Settings: Accuracy: 8, Time: 2, Interpretability: 7

Feature importance:



Final Model:

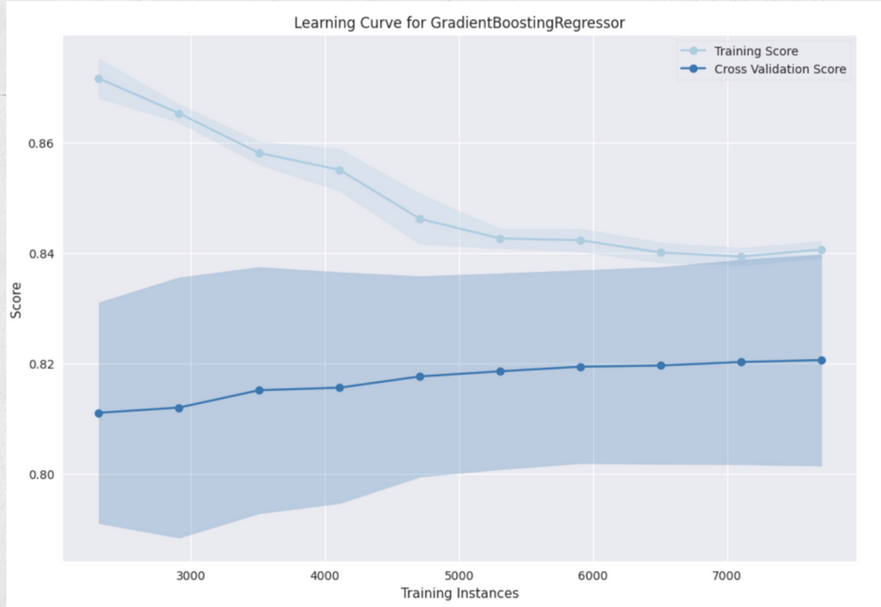


Solution 2: Low-Code Pycaret

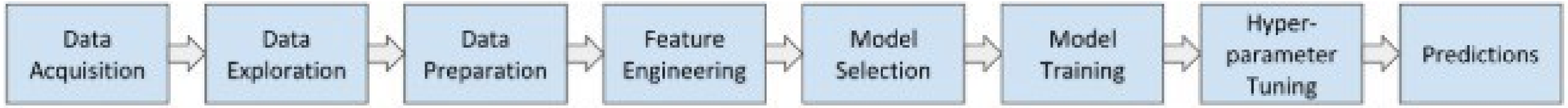
Generated a Gradient Boosting Regressor as its best model with an R^2 of 0.82

Note: Since pycaret was given standardized datasets, for a regression it's MSE cannot be compared as it is not on the same scale. Hence we looked at R^2 our second metric of choice. Pycaret uses its own preprocessing

We also tuned the pycaret GBR and performed K-fold cross-validation to give a final R^2 score of 0.8026



Our Workflow
Building regressors from scratch



-> The Data was acquired from Kaggle

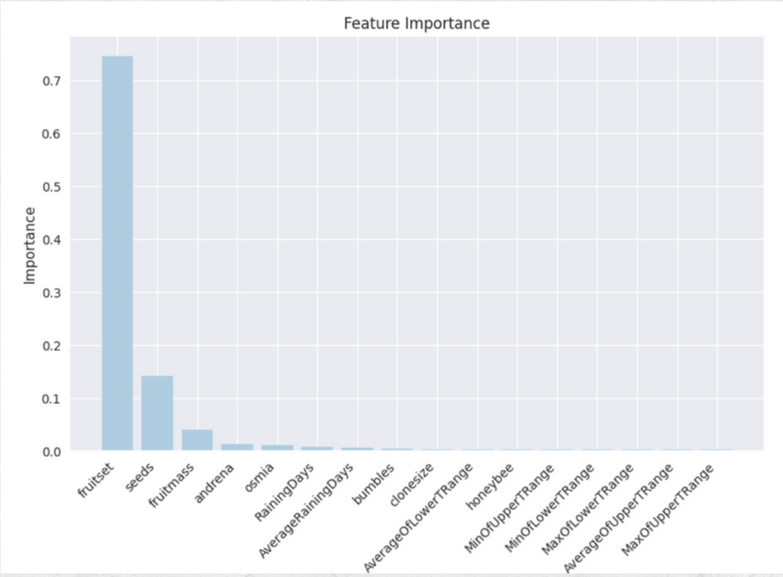
-> EDA was performed as described earlier, and we did not do feature engineering after looking at the driverless solution, as even after generating 21 new features, the performance improvement was negligible.

-> We did work a lot on Model Selection by Running Multiple Horse Races

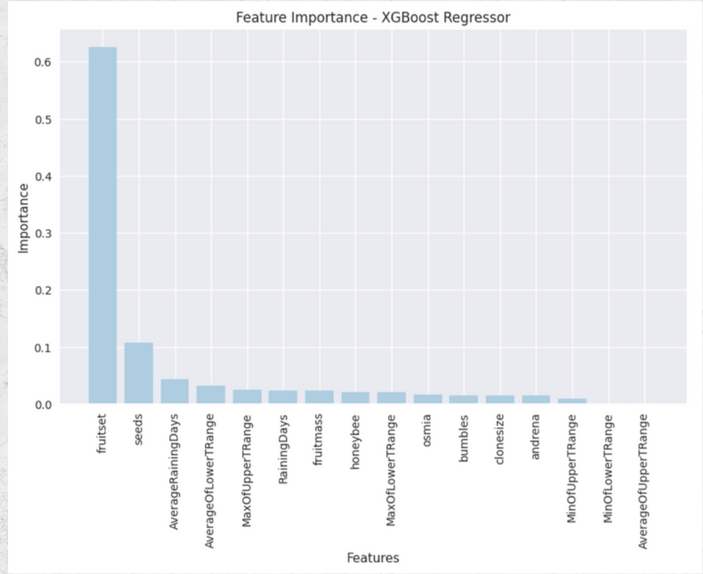
Feature Importance

We chose to build our models on 3 features: Fruitset, Seed, and fruit mass, after confirming importance across 3 different regressors. We chose to go for ElasticNet instead of Shapley as it forces the unimportant features to go to 0 providing clarity and confirmation to the previous feature importance.

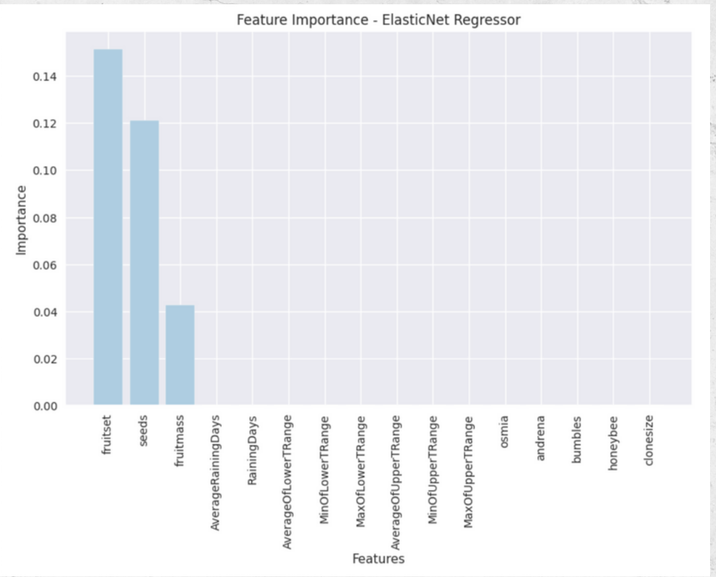
Random Forest



XGBoost



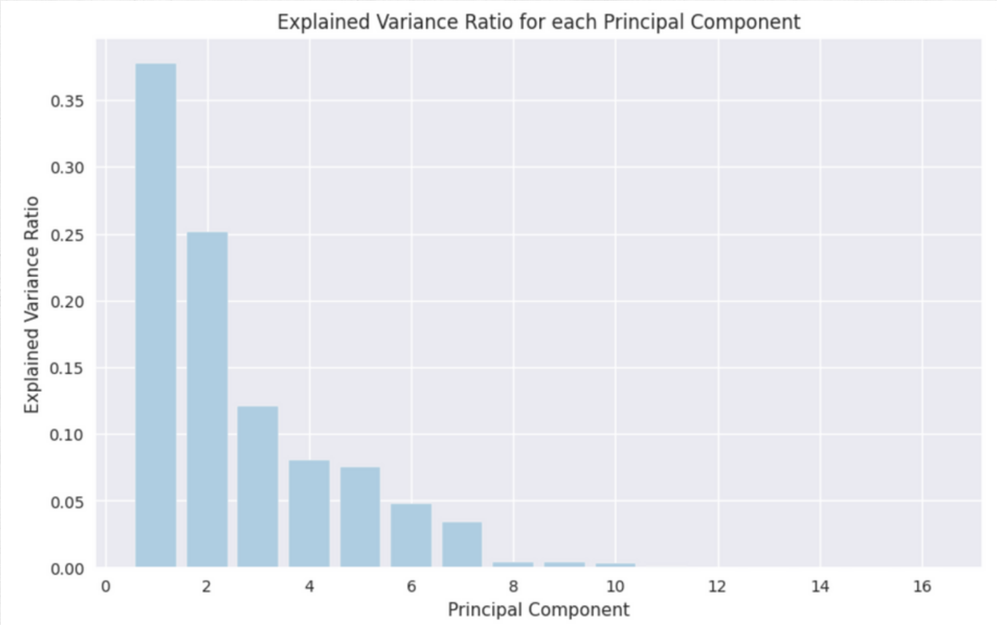
ElasticNet

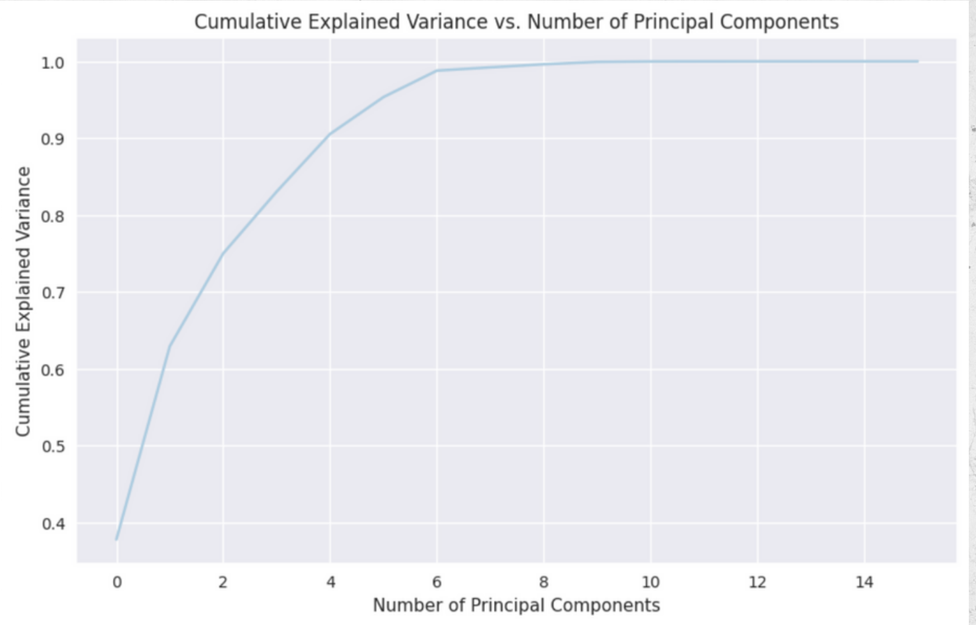


PCA

We managed to generate a decent PCA model achieving scores similar to our champion, however, we opted to now choose this due to the severe lack of explainability of the model. We were also worried about overfitting the model.

Final MSE:0.190 and R^2: 0.810



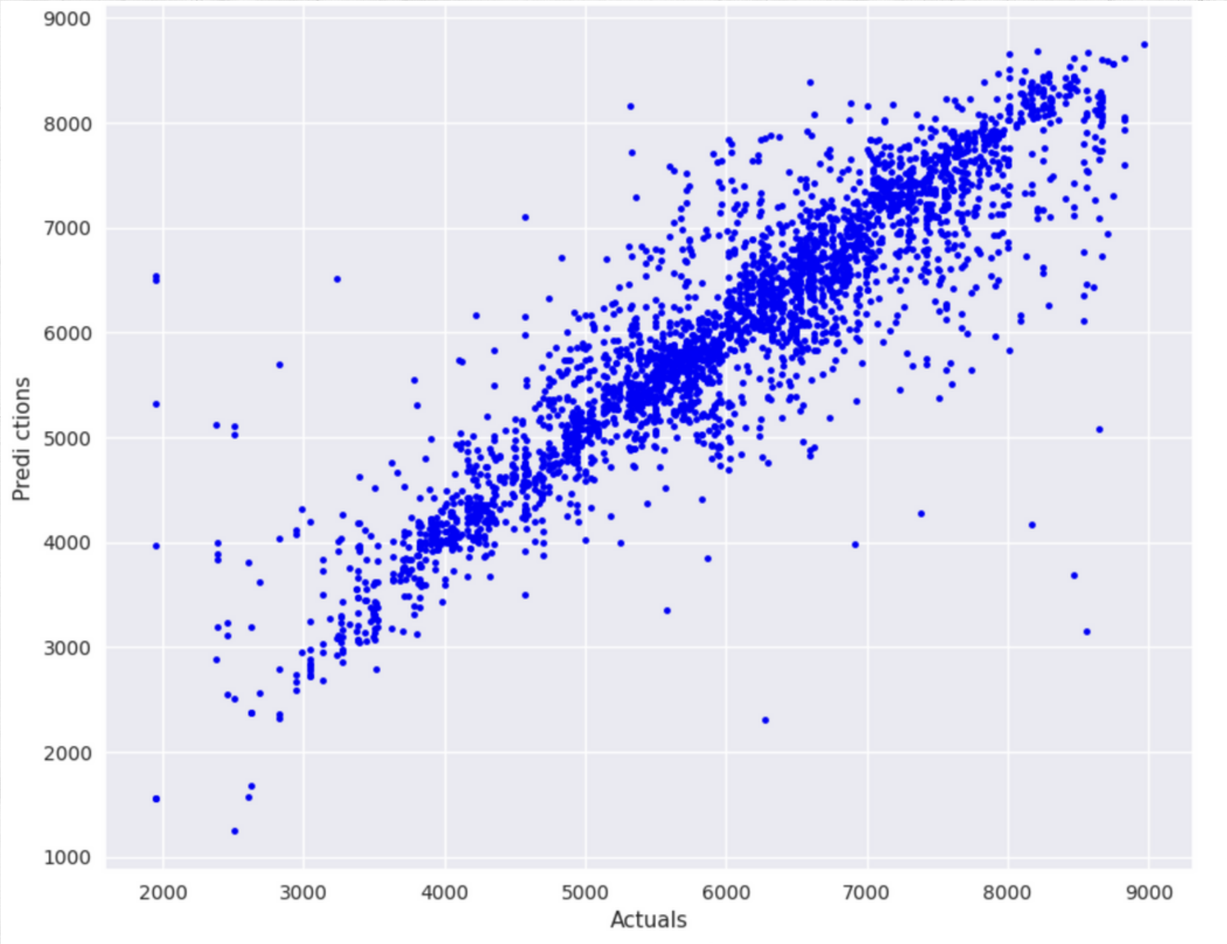


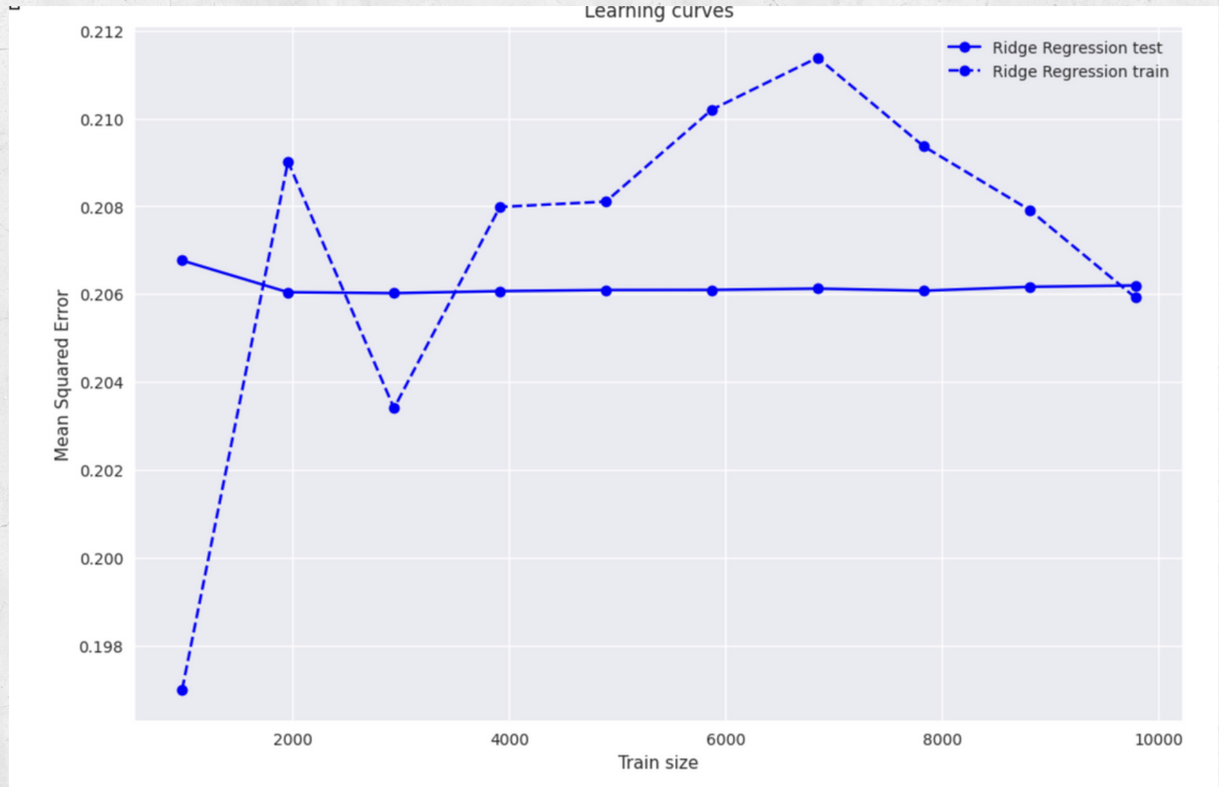
Horse Race 1: Base Regressors

Ridge Regression won, however, we were not satisfied with this since the MSE was way larger than our driverless solution. We plotted actuals v/s predicted graphs and learning curves for our models which can be seen in the next slides.

Regressor	MSE	MAE	R^2
Ridge Regression	3.392540e+05	371.198433	0.810670
Multiple Linear Regression	3.392626e+05	371.211562	0.810665
ElasticNet Regression	3.833547e+05	438.789791	0.786059
kNN	3.789035e+05	408.753801	0.788543
Multilayer Perceptron	3.503487e+05	382.181197	0.804479
Decision Tree	6.783415e+05	526.786215	0.621433
Lasso Regression	7.784511e+05	707.145180	0.565564
SVR	2.711572e+06	799.959226	-0.513265

Horse Race 1: Ridge Regression





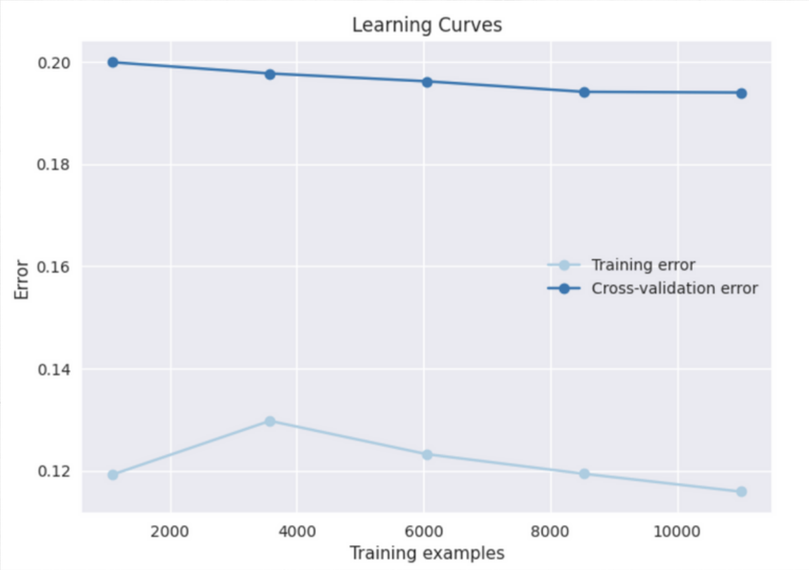
Horse Race 2: Ensembles

We pit Random Forest, XGBoost, AdaBoost,Keras Deep Learning, and a Stacked Ensemble against each other. While some like the XGBoost and the Stacked Ensemble performed better than the base regressors, other regressors were not suited for this dataset producing poor accuracies and worse Actuals V/S predicted graphs.

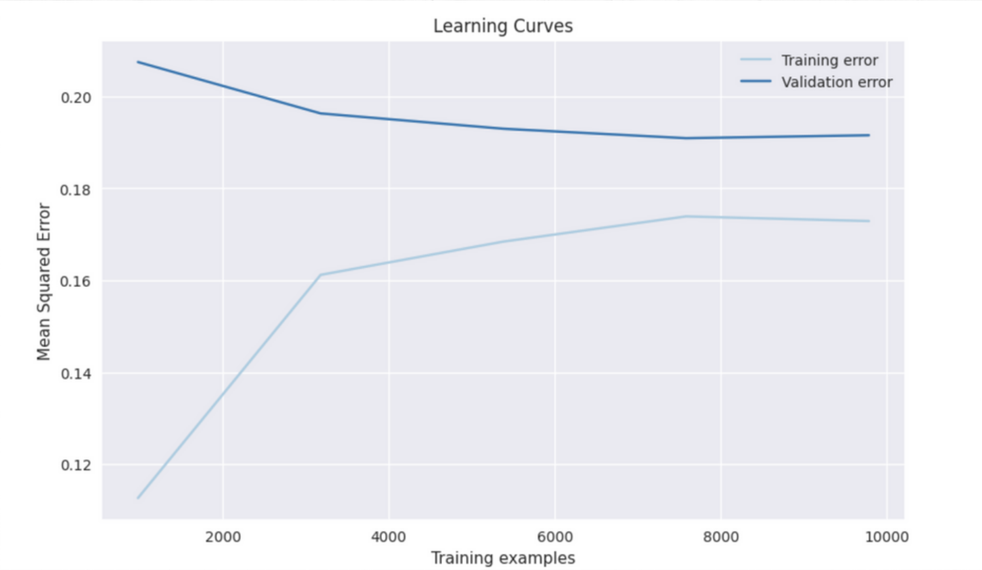
Overall, the stacked ensemble won with an MSE of 320849.70 far better than the second in place – the XGBoost with an MSE of 348716.49. However, we wanted to see if running such a complex model is worth it since it would cost our clients time and money, hence we ran a grid search and performed cross validation on both regressors.

Final MSE:0.179 and R^2: 0.8207

Stacked Ensemble



V/S



Our winner was XGBoost, mainly because of the following:

- Learning curves converged better
- Lower compute power needed: Faster training and results

XGBoost

Final MSE:0.1793 and R^2: 0.8212

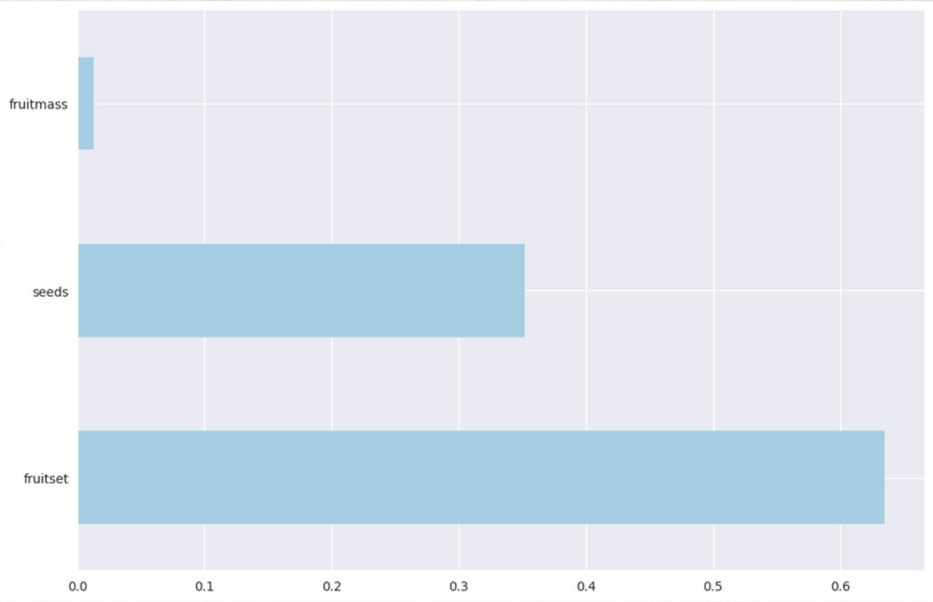
Final Model: XGBoost

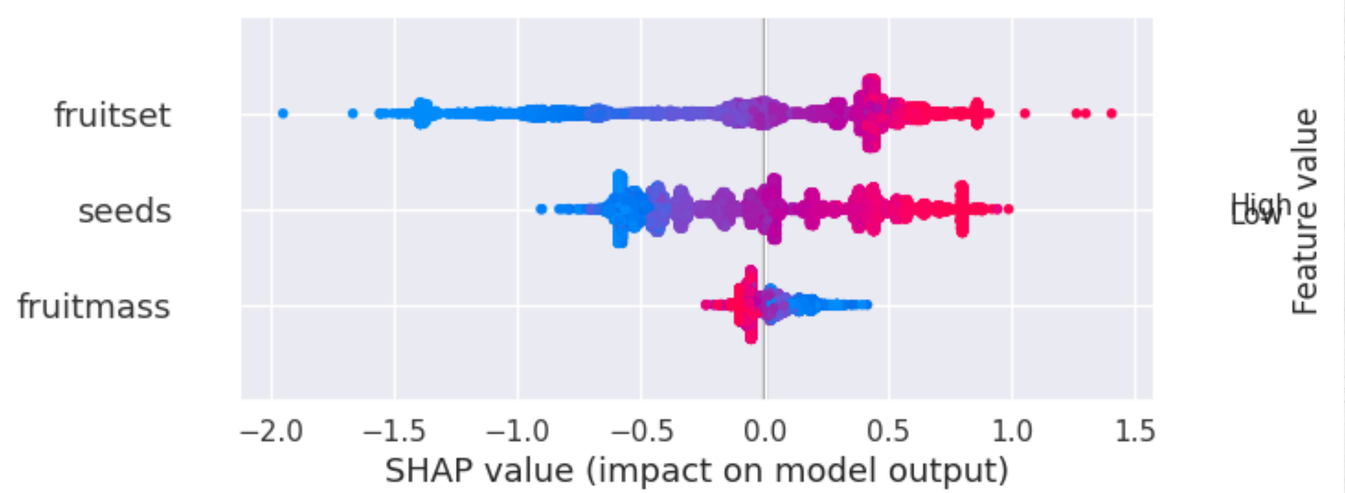
Tuned Parameters:{'learning_rate': 0.1, 'max_depth': 3, 'n_estimators': 100} | After grid search and 5 fold CV

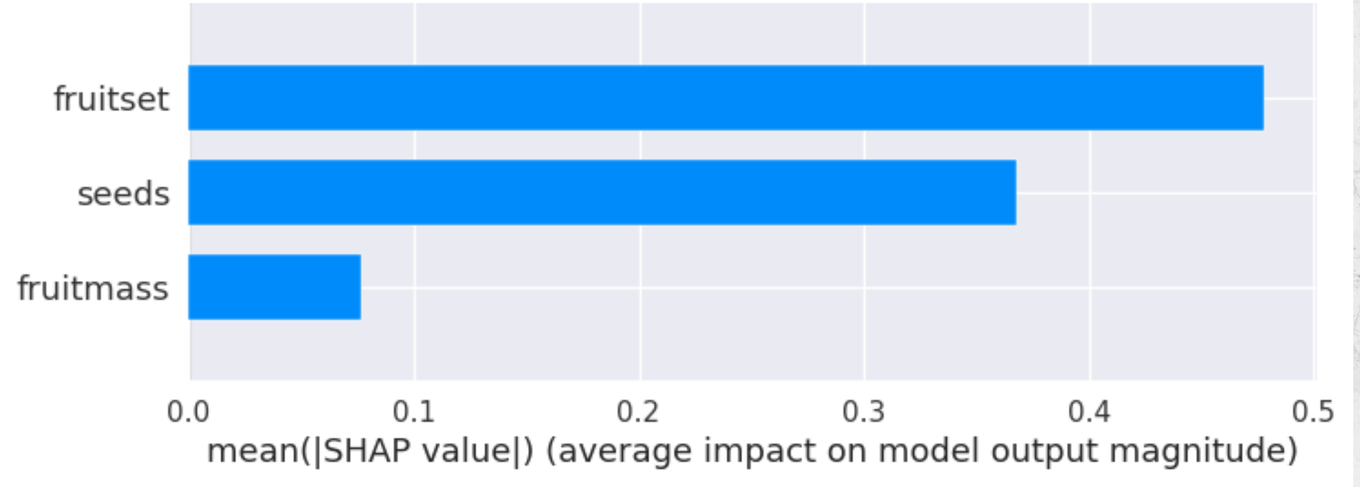
Actuals V/S Predicted

XAI

As can be seen with the feature importance, shapley values and the decision tree as well, fruitset is by far the most important feature. It explains the most variance and we can also see that in the shapley diagram, it has the most clear seggregation, which makes it valuable for yield prediction.



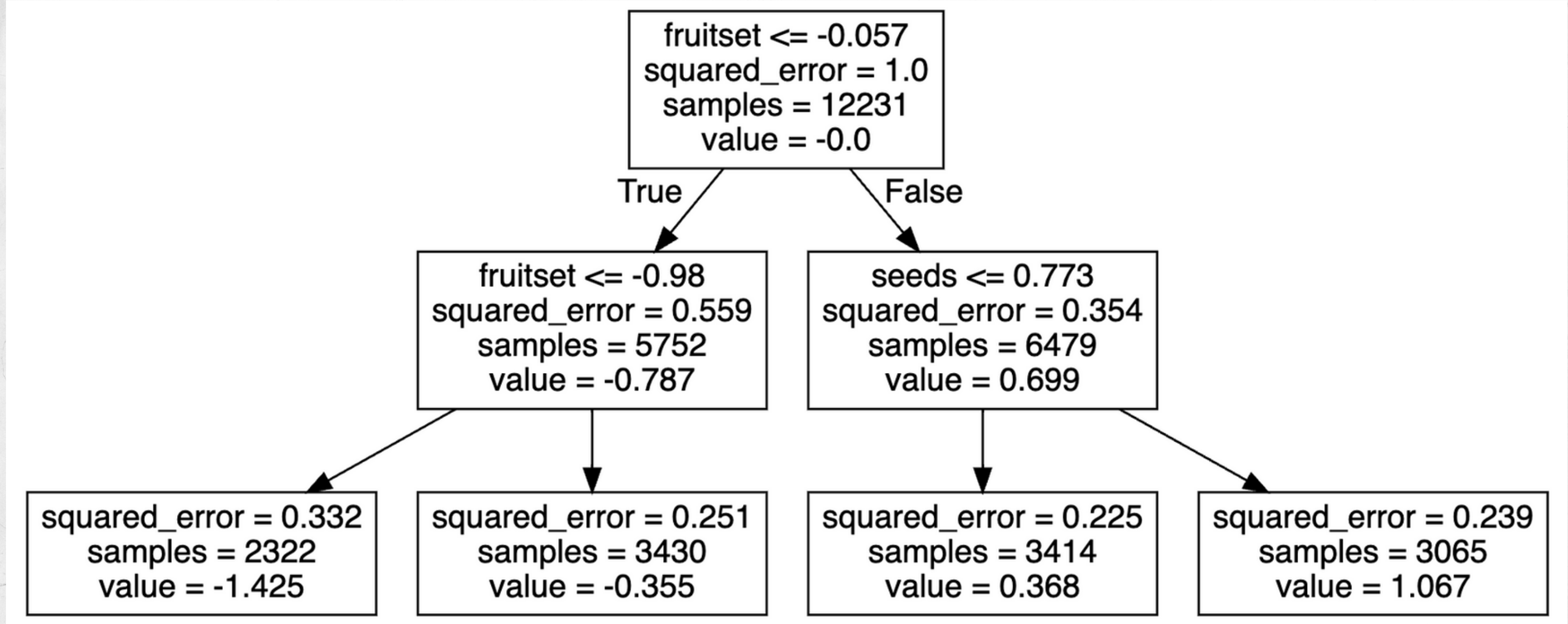




Surrogate Models

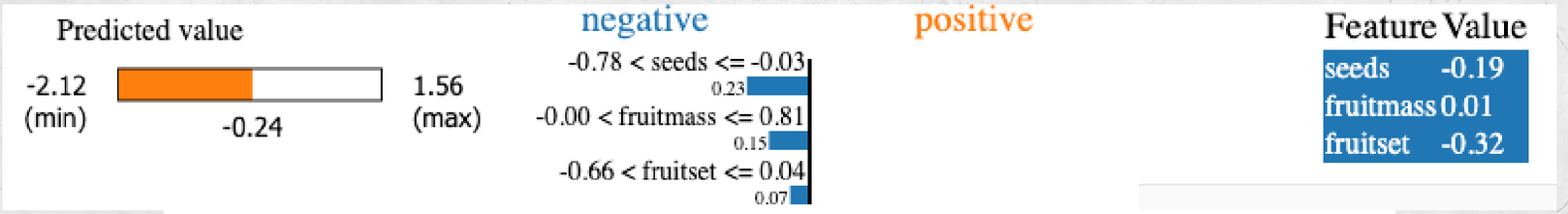
We built a decision tree model whose predictions correlated 95.9% with the predictions of our own model, enabling us to explain the working of the XGBoost model.

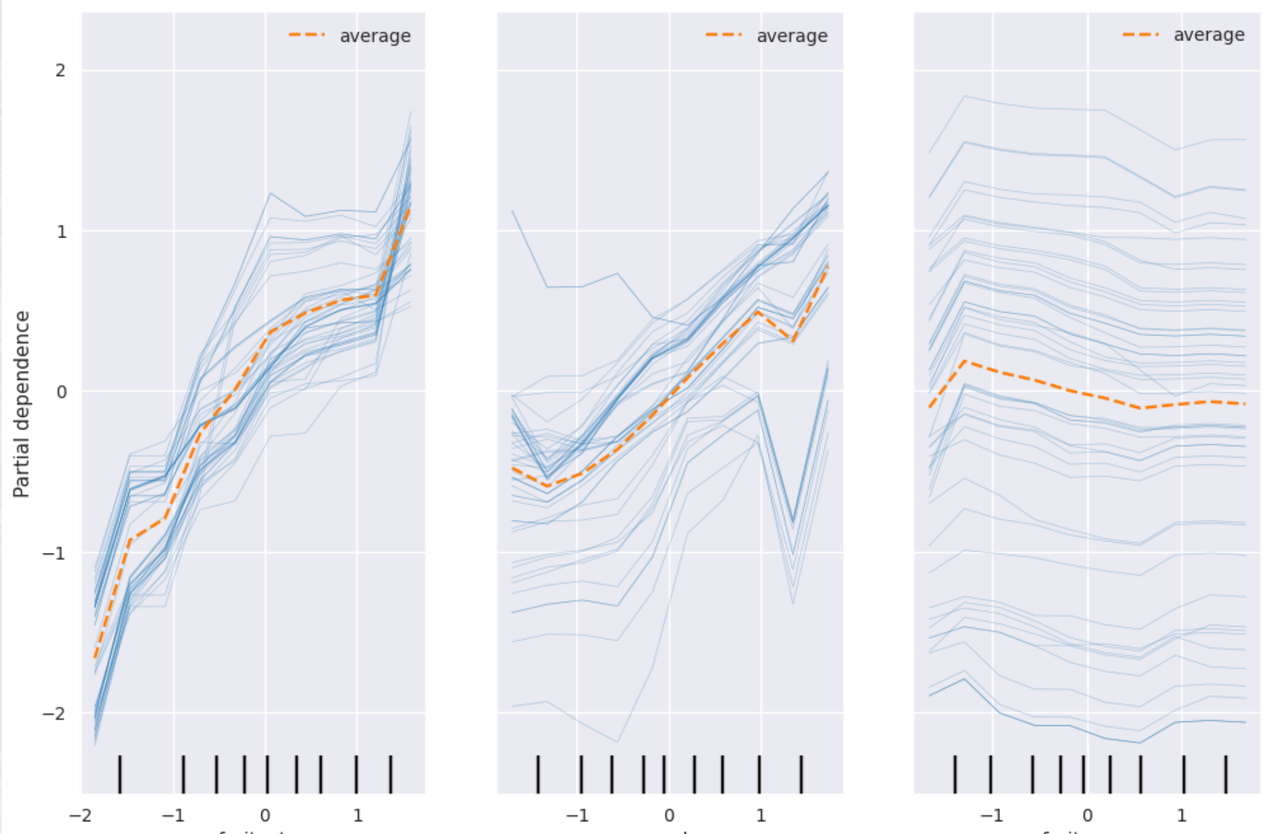
decision tree model



PDPs

In general, fruitset and seeds positively correlated, while the plot for fruitmass seems rather flat, or even slightly negatively correlated. Except for a few outliers, the ICE plots for most of the observations in the dataset appear to follow the trend showcased by the average line.





SWOT ANALYSIS

Strengths:

1. **Robust selection process**: Comprehensive analysis, rigorous criteria.
2. **Performance**: Outperformed others, including automated solutions.
3. **Model interpretability**: XAI tools, surrogate Decision Tree.
4. **Efficiency**: Balanced performance, resource utilization.

Weaknesses:

1. **Multicollinearity**: There is multicollinearity among the temperature features, which could potentially affect the model's ability to provide accurate predictions.
2. **Reliance on Few Features**: The model is built on just three features, which might limit its ability to capture complex relationships in the data.

Opportunities:

1. **Hyperparameter Tuning**: The team successfully applied grid search for hyperparameter tuning and validated the results using K-fold cross-validation. This approach could be refined further to enhance model performance.
2. **Feature Engineering**: Given the multicollinearity among temperature features, there might be opportunities to further engineer these features or explore additional feature selection methods.

Threats:

1. **Data Limitations**: The dataset, being artificially generated, may not accurately reflect real-world patterns and relationships, potentially affecting the model's applicability and generalizability.
2. **Auto AI:** Although we were able to beat the state of the art driverless AI, there may soon be a time where our model becomes insufficient.

Conclusion

We conducted thorough analysis and experimentation to select our champion model.

- **Feature Importance:**
  - Utilized multiple techniques, including elasticNet, to identify important features. Ended up with only 3 features for our model, excluding unimportant ones.
- **Data Quality:**
  - No missing values or duplicates in our dataset. No need for bias correction techniques like SMOTING or undersampling.
- **Multicollinearity:**
  - Observed multicollinearity among temperature features. Likely due to artificial generation of the dataset on Kaggle.
- **Model Evaluation:**
  - Chose XGBoost regressor based on performance metrics: MSE and R^2.
  - Required R^2 > 0.8 and minimized MSE for model selection.
- **Comparison with Other Solutions:**
  - Explored Pycaret and Aquarium H20 Auto AI solutions.
  - Our chosen model outperformed them, with simpler implementation and lower computational requirements.
- **Model Tuning:**
  - Conducted grid search to tune XGBoost model.
  - Validated results through K-fold cross-validation.
- **Explainability and Model Understanding:**
  - Generated a simpler Decision Tree Surrogate model. Achieved 95.9% correlation with our predictions. Leveraged XAI tools like LIME, PDPs, and Shapley Values for model understanding.

We are now open for your questions!

References

- ChatGPT (https://chat.openai.com/)
- Class Notebooks
- Medium.com (https://medium.com/) : exact references available in notebook