Table of Contents

Torrero Agronomy: Creating a Worldwide Agricultural Ranking System for Individual Producers

Authors: Hunter Blum, Jacqueline Urenda, Ebad Akhter
Company Name: Torrero Agronomy
Company Industry: Agronomy/ Agriculture/ Agribusiness
Company Size: 50 employees

Abstract

Torrero Agronomy specializes in agronomic consultation and provides commodity recommendations to producers of crops and livestock. The company is expanding to worldwide consultation and wants a model to provide automated commodity recommendations to our customers.

Problem Statement

Since 1960, global food production has been substantially increasing to keep up with the food demand caused by population increases. As the agriculture industry is challenged with meeting these high demands, many farmers and ranchers are seeking ways to efficiently optimize their resources and increase sales. Various factors can affect food production such as rain, temperature, fertilizers, etc. It is important to understand how these challenges can affect land and its produce outputs (OECD, 2023). Our agronomy company, Torrero Agronomy, was tasked with creating a ranking model that will provide agricultural producers with the top commodities to produce based on the total value (USD).

To address this problem, we utilized a variety of data sources that can affect the production of crops and livestock. According to Mishra et al. (2016), environmental factors, mainly weather, have the greatest impact on the success or failure of a crop. They also discuss how agronomic practices such as fertilizers can offset any negative effects on production from the weather, and push the crop to achieve higher yields and production. 33 percent of the world's crops are used to feed livestock (FAO, 2012), so the weather's effects on crops should directly correlate with which livestock commodities will be the top producers. Torero Agronomy used this data to create a ranking of top commodities that will provide the best yield and total value for production companies.

Goals

1. Create a commodity recommendation system that gives producers the most valuable commodities based on location, weather, and fertilizer practices.
2. Explore how weather, agricultural practices, and production vary across the world.
3. Compare traditional and Amazon Web Services (AWS) models to evaluate AWS's value to the company.

Non-Goals

1. We did not try to predict any specific yields or profits for individual producers.
2. We did not try to recommend any changes in agricultural practices to producers.
3. We did not incorporate any currency exchanges or complex trade relationships between countries.

Data Sources

The first two data sources focused on the yearly production and prices of each agricultural commodity for each year and country. We sourced the data from the Food and Agriculture Organization of the United Nations (FAO). The data set was quite large, with nearly 80,000 observations, so the main concern was that the data (especially units) would not be standardized or have typos.

Our next data source also came from the FAO and focuses on fertilizer inputs by country and by year. Our only concern with this data set was the variability in some countries appeared to be quite low or zero. This was likely due to countries simply reporting the same numbers every year.

Our final data sources came from the World Bank (collected from Kaggle). The two datasets had the average annual temperature and precipitation for each country and year, dating from 1985-2017. The World Bank data set provided on Kaggle did not go as far into the past as our FAO data, so we did lose years of data in our final combined dataset.

Each data set was cleaned within the *Clean_Wrangle* notebook in the *Data* folder in our GitHub repository. Each data source was in CSV format and read into *pandas* for cleaning. In this cleaning phase, we transformed each of the FAO datasets to a long format, standardized country names between FAO and World Bank data, and inner merged each dataset to produce the final dataset, *df_merged.* This dataset was uploaded into a shared S3 bucket to be read into the main modeling file, *Cloud_Team1*.

- FAO-sourced datasets:
  https://www.fao.org/faostat/en/#data
- World Bank / Kaggle sourced datasets:
  https://www.kaggle.com/datasets/patelris/crop-yield-prediction-dataset

Data Exploration

Data Storage and Ingestion

After merging, the data was stored on a shared S3 bucket. We created an Athena table to load the final dataset to the modeling file. This dataset was stored as a *pandas* data frame within the modeling file.

Tools for Ingesting and Exploring

We utilized S3 to store the data, then created an Athena query to store the data in a *pandas* data frame. To create our geographic data, we used *GeoPy*, a library that returned each country's coordinates based on the country's name. *Geopandas* was used to create geo data frames for plotting maps. Maps and most other plots were made with *matplotlib.pyplot*, while our correlation heatmap was built with seaborn.

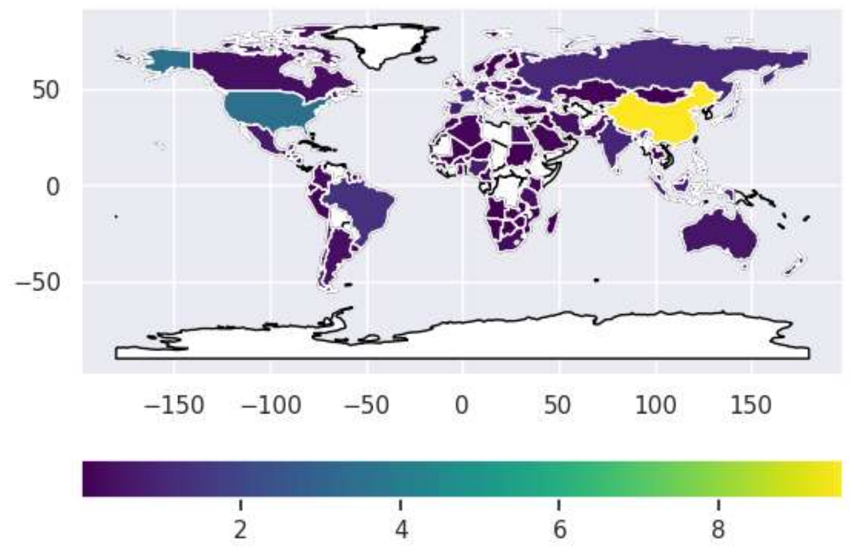
Exploratory Data Analysis

During the exploratory phase, the data was examined for null values, variable distributions, and correlations. Numerical features included *year*, *production*, *nitrogen*, *phosphate*, *potassium*, *value*, *avg_rain*, *avg_temp*, and *total_value_usd*. The selected target variable for our model was total_value_usd, which is the product of the production and value features. The categorical features in the dataset were *area* and *item*. The *area* feature contained country name data and the item feature contained crop data.

Using the *geolocator* function from *GeoPy*, we generated each country's latitude and longitude from the *area* feature. The geographical data was used to create several maps that displayed the total agricultural value, average temperatures, average rainfall, and average fertilizer inputs (nitrogen, potassium and phosphate) across each country between the years 1991-2013. As seen in *Figure 1,* our data has China as the world's largest producer by value with over 8 trillion USD produced. The United States produced the second most, followed by other notable countries such as Russia, India, and Brazil. *Figure 1* also displays the many missing countries from our data sources.

Figure 1

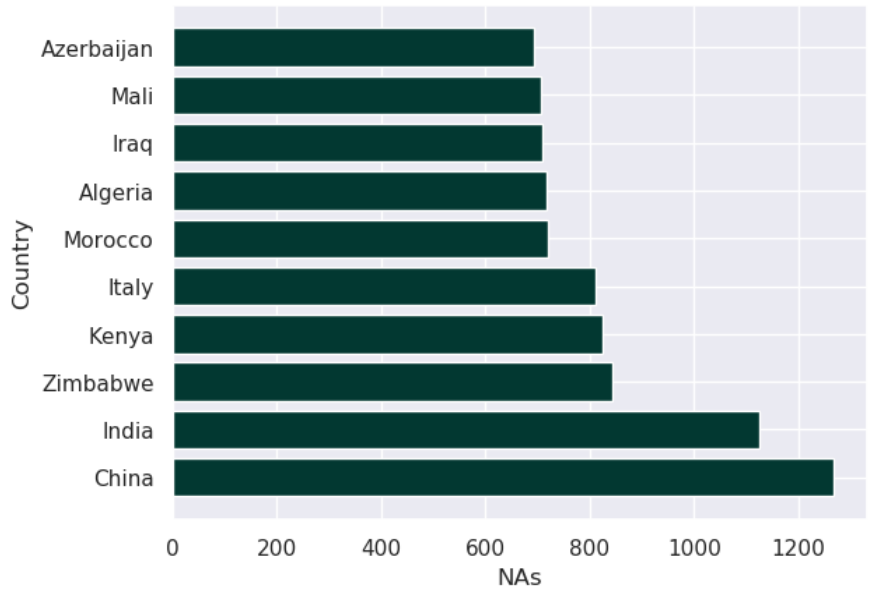
Total Agricultural Value Produced 1991-2013 (Trillions of USD)



The dataset contained a large number of null values in the following features: *production, nitrate, phosphate, potassium, value,* and *total_value_usd*. Since *total_value_usd* was partly a product of the *value*, our target had the most missing obsservations. As shown in *Figure 2*, China and India accounted for the most null values of our target feature *total_value_usd*. Since we cannot impute the target feature, each of these null observations was deleted, however, each of the countries maintained enough observations for modeling.

Figure 2

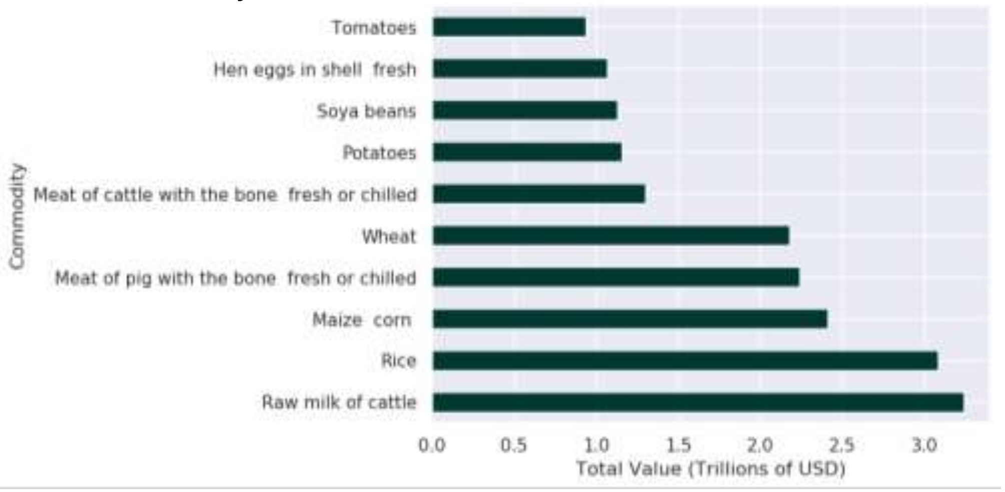
Total Value NAs by Country- Top 10



We also conducted an analysis of the categorical features, *area* and *item*. The *area* feature had 96 unique countries, while the *item* column had 189 different commodities. As shown in *Figure 3,* the most valuable commodities across the world were cow milk, rice, and corn.

Figure 3

Top Ten Commodities by Total Value

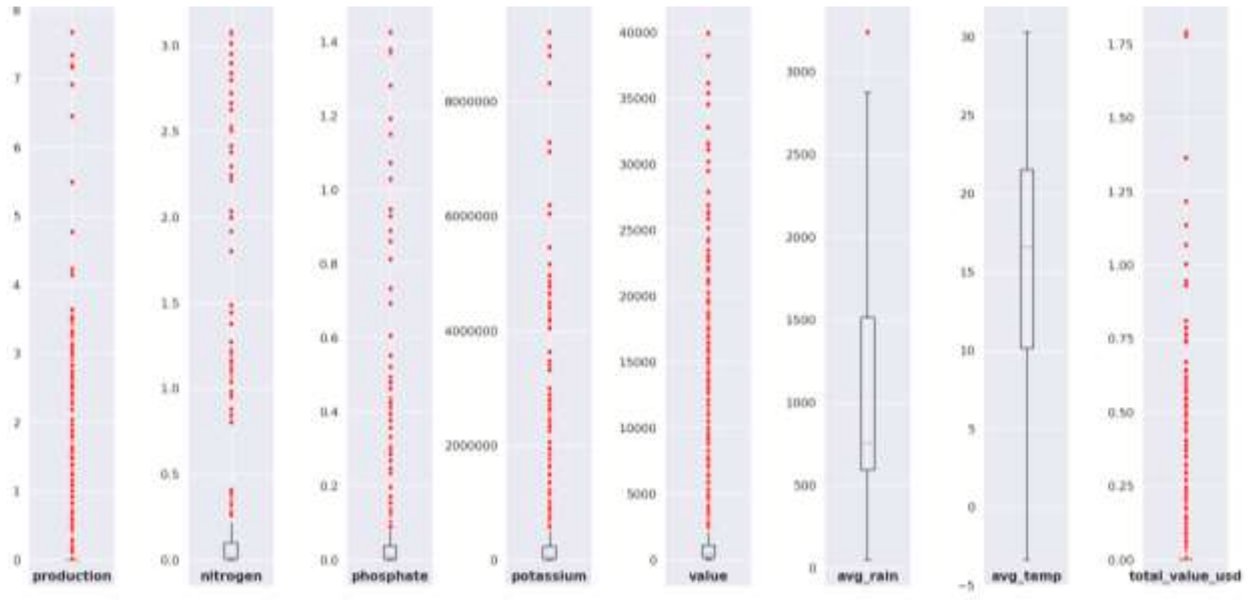


Conclusions of Exploratory Data Analysis

During exploratory data analysis, we wanted to take careful consideration of the null values because of the many different data sources spanning long time periods. We discovered that the target variable, *total_value_usd,* had many missing observations that required removal. Due to the large number of missing values in other features, we also implemented an imputer. Next, we took into consideration the distributions of our numeric features, especially *total_value_usd.* Given the vast difference between agricultural industries around the world, most features were skewed heavily to the right. Since our numeric data was heavily skewed, we also wanted to evaluate the outliers of the data. As shown in *Figure 4*, all of the features except the weather variables had a large number of outliers. To control the outliers and distributions, we also implemented a scalar to work in tandem with the imputer.

Figure 4

Boxplots of Numeric Features



GitHub Repository

The main notebook for the GitHub repository is titled *Cloud_Team1.* The needed installations to run this notebook can be found in the *Dependencies* notebook in the *Dependencies* folder. Each dataset and the cleaning notebook (*Clean_Wrangle)* are located within the *Data* folder. Finally, this report, a code addpenix, and the presentation slides are located in the *Documents* folder.

Github Repository Link:

https://github.com/hunterblum/CloudFinal-Team1.git

Measuring Impact

Since one of our non-goals was to not predict specific profits, it was difficult to quantify the current business impact this model could have. So, we focused on modeling metrics which will give a proof of concept that we can give relevant recommendations to producers.

Our first goal was to achieve a root mean square error (RMSE) of less than ten ranks. This means that, on average, a ranking from our model will be wrong by less than ten rankings. However, this metric could still provide poor results if the model provides two very irrelevant commodities as the top two recommendations every time, while most other rankings are correct. So, we also want to achieve a Spearman rank correlation of at least 0.7. This will indicate that our model has a moderate correlation with the ideal rankings.

Security Checklist, Privacy, and Other Risks

Since we were only storing country-aggregated data that was mainly provided through public United Nations datasets, we did not work with any PHI, PII, user, or credit card data. We read from the S3 bucket, *sagemaker-studio-998234604495-nyhifbo32oo*, which was completely open to the public due to the nature of our data. We are also writing our Athena queries to a separate S3 bucket, *aws-athena-query-results-998234604495-us-east-1.*

The largest bias that we needed to consider is availability bias. Our data set was very broad, only focusing on country-wide data. So our recommender may struggle to give relevant commodities to an individual in some areas. We attempted to mitigate this by utilizing the latitude and longitude of each country's center. However, this could give worse model results to a producer who lives far away from the center of any country. We also did not implement any other data that could be relevant to agriculture such as topography or soil profiles, which could also affect individuals that live in atypical places like a river valley. Our main ethical concern tied in closely with the bias. Since we are giving individuals advice on their agricultural business, providing them with relevant recommendations is paramount for their financial well-being. We need to be vigilant in constantly monitoring and improving the model's performance and coverage.

Data Preparation

Data Scrubbing

Most data cleansing was performed before writing the merged dataset to our S3 bucket. Each of the UN datasets (crops, prices, and fertilizers) had the year variable in a wide format, so we transformed each dataset to a long format using *pandas' melt* function. We only kept the year, country, and each dataset's relevant measure (production from *crops*, price from *prices*, and fertilizer use from *fertilizers*). The temperature and rainfall data had duplicates for some countries, so we only kept the first observation for each set of duplicates.

Each of our *pandas* data frames was inner merged using the *merge* function. The fertilizer and crops data frame was merged on the country and year, and this dataset was merged with prices using the country, year, and commodity. Before joining the combined UN data with the weather data, we changed the country names in the UN dataset to match those in the weather data. Finally, we completed two more inner joins on the country and year columns to bring our temperature and rainfall data into the final dataset.

Once the data was read from our S3 bucket into our main script, we removed any missing data from our target variable, *total_value_usd*. The fertilizer variables also had missing data, but these will be imputed in our modeling pipeline using the mean of the feature.

Feature Selection

Feature selection mainly revolved around removing irrelevant features from the dataset. Irrelevant features such as the unit and description columns were removed from each of the UN datasets before the final merge and S3 upload. Before modeling, we also eliminated the *production, value, area,* and *year* columns. The *production* and *value* columns were only needed to create our target feature and the *area* column was only needed to create the coordinate columns. The *year* column is excluded from modeling because our goals do not relate to time-series forecasting.

Feature Creation

We created the target feature *total_value_usd*, by multiplying the *production* and *value* features. We also made the geographic coordinate columns (*lat* and *long*), with the *GeoPy* library. *GeoPy* takes the name of each country and provides the coordinates for the geographic center. No other fields were either combined or bucketed.

Feature Transformation

The fertilizer variables were transformed into ratios since fertilizer is typically sold with Nitrogen-Phosphorus-Potassium (NPK) ratios on the label. Each numeric predictor was z-score standardized in our modeling pipeline.

Balancing

We will not be balancing the data set between the hundreds of commodities as no commodity appears to dominate the data.

Splitting

The data was split with *scikit-learn*’s *train_test_split* function. We split the data into a training, validation, and testing set with a 90-5-5 ratio, ensuring to keep the random state constant.

Data Training

Modeling Tools

We decided to try multiple models on the training dataset. We wanted to compare models from *scikit-learn* and pre-built SageMaker Image URI models. We chose to create our own models with *scikit- learn* to establish a baseline. Then we selected the pre-built algorithms to have highly accurate and deployable models.

Algorithms

The first algorithm used was the multiple linear regression from *scikit-learn*. We took the total value predictions from the model and ranked them within their respective country and year. This allowed us to establish baseline metrics for the other ranking models. Since we added many inputs from one-hot encoding the commodities, we also performed an elastic net regression. The combination of L1 and L2 regularization penalties from elastic net gave us a model that could be more resistant to overfitting, a potential problem we faced from adding dimensionality.

We then used two prebuilt Amazon Sagemaker regression models. First, we transferred the training data into an S3 bucket. Next, we set up the container for the first model, XGBoost, and defined the hyperparameters before training and deploying the model. XGBoost was useful for our business goals since boosting can prevent overfitting, one of our main concerns with the models. XGBoost is also able to capture non-linear relationships in our data, so it could capture patterns that our baseline models could not uncover.

Finally, we performed the same process for the prebuilt Linear Learner algorithm. With Linear Learner, we hoped to take advantage of automated feature scaling. Amazon SageMaker Linear Learner only takes a small sample to scale the data on, so it is potentially more efficient than our baseline scaling pipeline (Mishra et al., 2019).

Parameters

We passed 195 parameters into our models. The first two, *avg_rain* and *avg_temp*, let us establish a relationship between a producer's local weather and the crops that thrive in those environments. Next, we considered the geographic coordinates which can capture the non-weather factors that could impact the producer. For example, a producer’s soil type is highly related to their location. So we can essentially capture unknown factors that are still key to agricultural production based on a producer’s location.

Then, we input the fertilizer ratios (nitrogen, phosphate, and potassium). These ratios capture what fertilizers a farmer either has on hand or has future access to. Since fertilizers are key to the commercial growing of many plants, we included the ratios to capture the relationship between specific mixes and the most valuable plants. Finally, we included the one-hot encodings for each of the 189 commodities. The commodity inputs let us create unique weights for each commodity in our models. With these weights, we can create predictions for each commodity and sort those predictions to generate our rankings.

Instance Size and Count

We used an *ml.m5.large* instance, which is optimized for general-purpose workloads. Specifically, the *m5* instances use the latest generation of Intel Xeon processors and are designed for compute-intensive workloads (AWS, n.d.).

The *large* instance size indicates that it has relatively high memory and CPU resources, making it suitable for a wide range of workloads. In terms of pricing, the cost of an instance varies based on the type and size of the instance, as well as the location in which it is deployed (AWS, n.d.). Since we trained models requiring high memory and computing resources, we found this instance to be the most appropriate for our use case. However, we did need to carefully monitor our budget using this large of an instance size.

Other instance types we considered in SageMaker include *ml.p2.xlarge* which is optimized for deep learning workloads and includes specialized hardware for training deep neural networks, and *ml.c5.4xlarge* which is optimized for compute-intensive workloads and includes high-performance networking capabilities (AWS, n.d.).

Model Evaluation

Since our model serves as a prototype to eventually expand towards individual producers, our model evaluation and business goals are essentially the same. We will be evaluating our models based on two metrics, the root mean square error (RMSE) of the rankings and the Spearman rank correlation. Each model's prediction will be compared to the rankings of the testing data to calculate these metrics. The RMSE will allow us to determine how many rankings any commodity is wrong by, on average. Spearman rank correlation is a measure of the correlation between our model rankings and the true recommendations, which will penalize us more for having irrelevant recommendations than the RMSE. Our goal was to achieve an RMSE of less than 10 ranks and a Spearman rank correlation of at least 0.7. Achieving these metrics will give a proof of concept for our model, instilling confidence that we can apply the same methods for individual clients.

Unfortunately, we fell just short of our goals. Our best-performing model, XGBoost, achieved an RMSE of 14.53 and a Spearman rank correlation of 0.63. However, the XGBoost, a Sagemaker pre-built model, did outperform both our baseline multiple linear regression and elastic net models (Figure 6). This indicates that AWS could be a useful tool for Torrero Agronomy to efficiently train the highest accuracy models. Comparing the scatterplot of the model and test rankings, we did see a moderate positive correlation between the rankings (Figure 7). However, *Figure 7* also exhibits many commodities that were ranked top 10 by our model that were not even in the top 20 of the actual rankings. So, we will need to continue to improve our model to meet our customers' needs in the future.

Figure 6:

Comparison of Selected Metrics between the Four Models

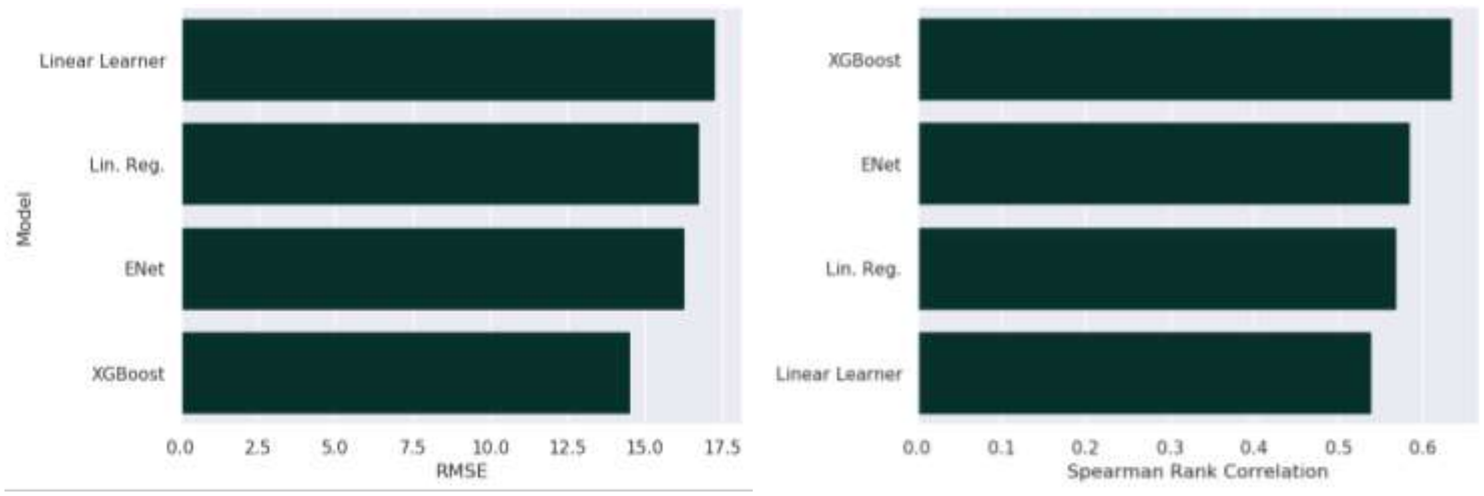
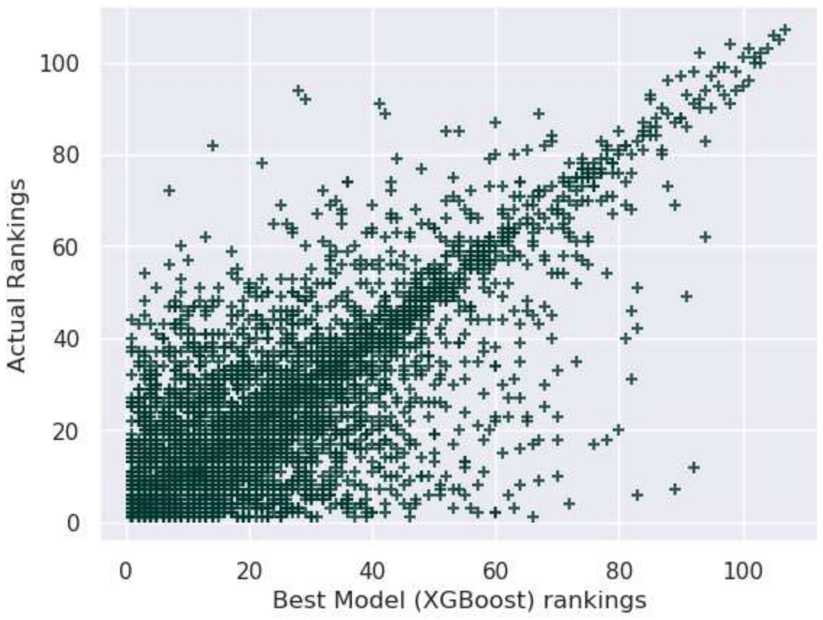


Figure 7:

Scatterplot of XGBoost Predicted Rankings versus Actual Rankings



Future Enhancements

Data and Feature Collection

One challenge that we faced throughout data ingestion and feature engineering was merging our five datasets to keep as much data as possible. However, even with manually changing country names to match between datasets, we only had 103 out of 193 countries that the United Nations recognizes (United Nations, n.d.). We also were missing data temporally, as our final data set only goes to 2013 because of our temperature data. So, we want to search for data sources that cover more recent data and countries. Getting more recent data ensures that we are basing our recommendations on commodities that are valuable for the producer in the present day. Covering more countries will allow us to provide more accurate models to users across the world.

Further, we would improve both the number of features and the scale we collect the current model’s data on. The main features would pertain to the soil of the area. Of the 17 essential elements for plant growth, 14 are mineral elements that must come from the soil (Parikh and James, 2012). So, soil features should serve as a great predictor of which crops to plant, or if a producer should focus on livestock instead of crops. Lastly, it would be useful to find more localized data rather than for entire countries. Since our model is currently based around each country as a whole, a user may get an irrelevant crop in an area with diverse geography and weather. More local data would allow users to have even more relevant recommendations. However, we would also need to consider the budget of the future project, as having more data requires more computing power.

Modeling & Autopilot

Another method for future enhancements that we would consider is the SageMaker Autopilot ML selection tool. Since the tool automatically selects best-performing models for the data, this method would help us quickly explore other models and hyperparameters that may have better results than our current XGBoost model.

Using Autopilot could also provide us with automatic pre-processing and feature engineering methods we haven't considered. Additionally, this tool could give us an explainability report of feature importance, which could have helped us understand how relevant our features were (Mishra et al., 2019). Overall, using Autopilot could be a faster and more efficient way of determining which model works best for our data.

Individual User Data

Since our current model is a prototype based on the entire country, we want to expand our data collection efforts to individuals around the world. Collecting user data would help us capture unique geographic regions that cannot be captured by our country-wide data. With more diverse data, we could create more accurate rankings, which hopefully will maximize our users' incomes. However, since we are collecting personal data, we will need to create a new security policy to ensure the data is safe.

Collecting individual data could also allow us to expand the purposes of the model. We could start asking producers about their specific profits and costs associated with their commodity. This would allow us to generate rankings based on the producer's profit rather than the total value. We could collect data on whether users are currently practicing sustainable or organic farming practices, and if so, what methods they are using. With this data, our model would now also include organic or sustainably produced commodities in its rankings. Combined with having specific profits, our new ranking system could encourage producers to adopt more sustainable commodities and practices. For example, if the model knows that a user is located in an area with high demand for organic meat, it could recommend that the user focuses on raising livestock for organic production. Similarly, if the model knows that the user is located in an area with harsh weather conditions, it could recommend livestock breeds that are more resistant to those conditions. All in all, incorporating individual user data will help make more relevant recommendations to our users, leading to an increase in customer loyalty and satisfaction.

References:

Amazon Web Services (AWS). (n.d.). Amazon EC2 Instance Types. Amazon Web Services. Retrieved April 14, 2023, from https://aws.amazon.com/ec2/instance-types/

Food and Agriculture Organization of the United Nations (FAO). (2012). *Sustainability and Organic Livestock*. Sustainability Pathways: Sustainability and organic livestock. Retrieved March 13, 2023, from https://www.fao.org/nr/sustainability/sustainability-and-livestock

Organization for Economic Co-operation and Development (OECD). (2023). *How we feed the world today*. OECD. Retrieved April 1, 2023, from https://www.oecd.org/agriculture/understanding-the-global-food-system/how-we-feed-the-world-today/

Mishra, S., Mishra, D., & Santra, G. H. (2016). Applications of machine learning techniques in agricultural crop production: a review paper. Indian J. Sci. Technol, 9(38), 1-14.

Mishra, A. (2019). *Machine learning in the AWS cloud: Add intelligence to applications with Amazon Sagemaker and Amazon Rekognition*. AWS. Retrieved from https://aws.amazon.com/sagemaker/autopilot/?sagemaker-data-wrangler-whats-new.sort-by=item.additionalFields.postDateTime&sagemaker-data-wrangler-whats-new.sort-order=desc

Parikh, S. J. & James, B. R. (2012) Soil: The Foundation of Agriculture. Nature Education Knowledge 3(10):2 https://www.nature.com/scitable/knowledge/library/soil-the-foundation-of-agriculture-84224268/\

United Nations. (n.d.). *About Us*. United Nations. Retrieved April 8, 2023, from https://www.un.org/en/about-us#:~:text=The%20UN's%20Membership%20has%20grown,members%20of%20the%20General%20Assembly.