Financing and Training to Address the Gender Gap in Agriculture in Peru

A supervised learning, random forest regressor model
Predicting where the gender gap in agriculture in Peru is more pronounced, down to the district level

Jose Carlo Burga

Introduction

Financing and Training to Address the Gender Gap in Agriculture in Peru

A supervised learning model trained to predict locations where the gender gap more pronounced within the agricultural sector in Peru

LaGuardia Community College
Continuing Education Program
Data Analytics Program
Python for Data Analytics

Jose Carlo Burga

Abstract

This project addresses the gender gap in agriculture in Peru by focusing on financing and training. It explores the disparities faced by women in accessing agricultural resources and the impact of targeted financial support and training programs. Using a combination of statistical analysis and machine learning models, the research highlights significant barriers women encounter, such as high interest rates, lack of collateral, and red tape. The findings suggest that improving financial inclusion and providing tailored training can enhance women's productivity and economic empowerment in the agricultural sector.

The machine learning model, specifically a Random Forest model, is trained to predict locations where the gender gap is most pronounced. This prediction is based on various features such as socio-economic indicators, agricultural productivity, access to financial services, and availability of training programs. The model's predictions are based on public data from Peru's National Agrarian Census of 2012, and the strategy will be adjusted as necessary based on new data, model feedback, and results.

This approach combines data-driven insights with practical interventions, providing a powerful tool for tackling gender inequality in agriculture. By accurately predicting the locations where the gender gap is most severe initially to the region level, for resources can be allocated more effectively, thus leading to greater impact.

GitHub Repository

#Peru #Agriculture #Gender_Gap #Financing_Access #Training_Access #Machine_Learning #Supervised_Learning #Regression #Random_Forest #Scikit_Learn #Pandas #Jupyter_Notebooks #Studio_Visual_Code #Python

Audience: Stakeholders in the agricultural sector

- Bilateral and multilateral organizations focused on sustainable agriculture
- Financial institutions involved in sustainable investments in agriculture
- Private sector entities in agriculture
- Policy makers
- Researchers
- People, especially women, in agriculture

Benefits: Generate significant improvements in:

- Agricultural productivity
- Economic growth
- Social equality

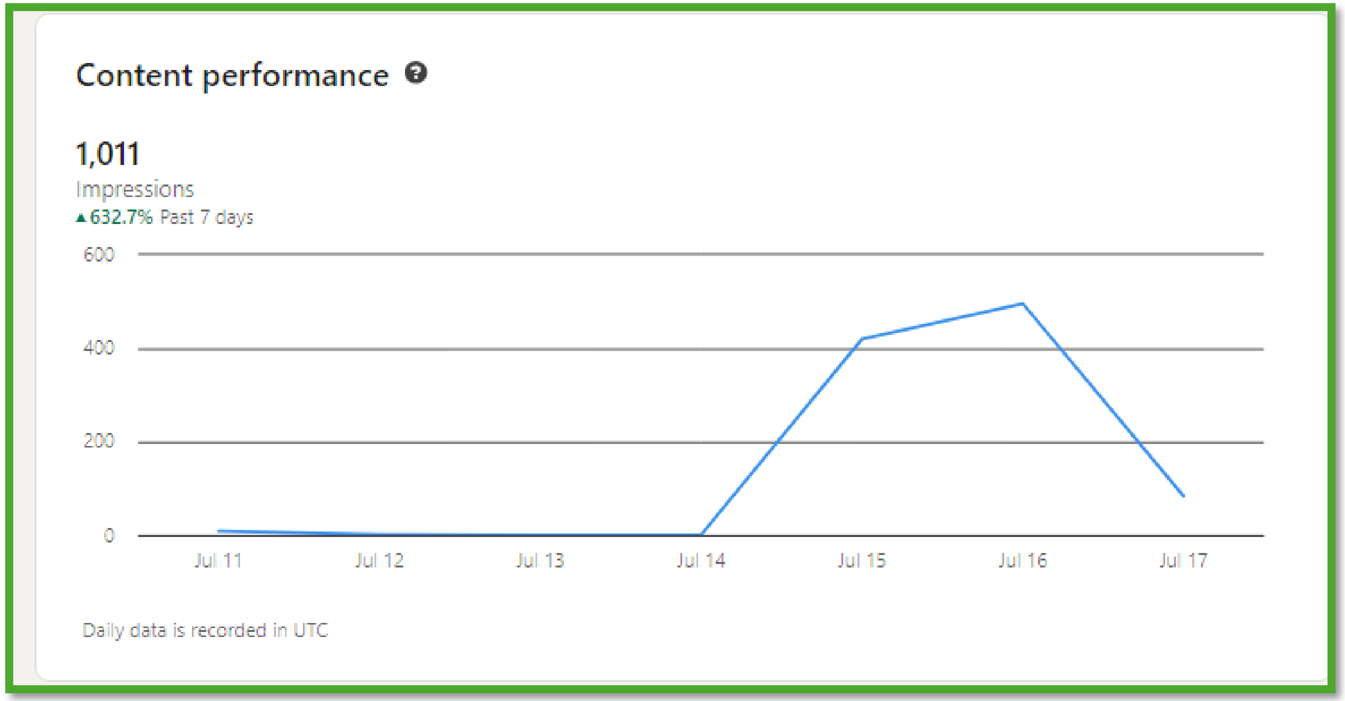
Call to Action:

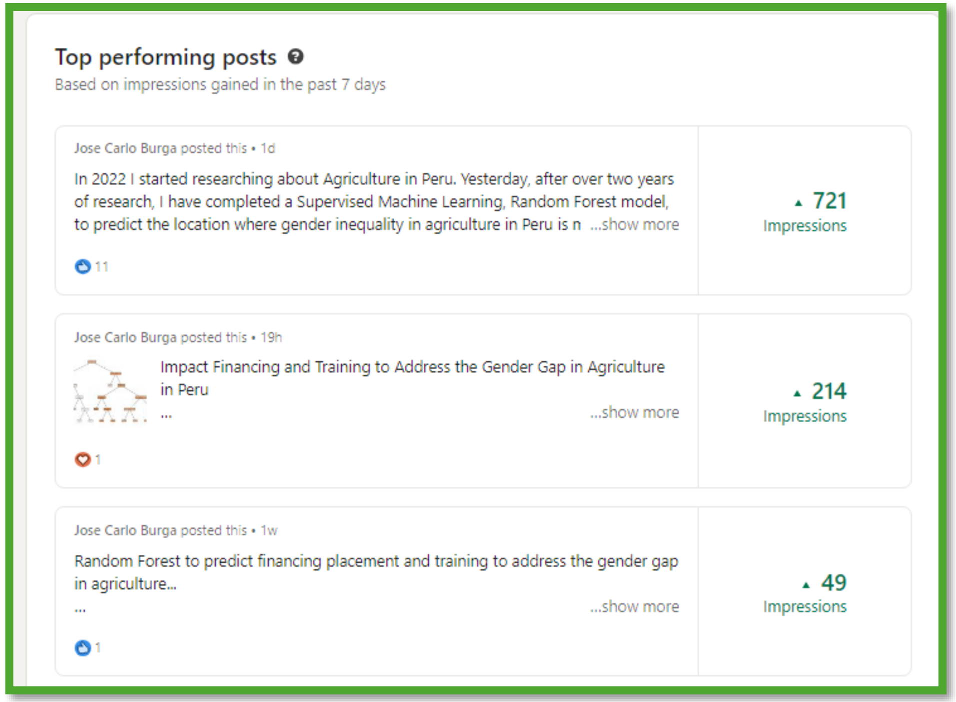
- To deliver financing and training for women in agriculture
- To further research in this area
- To reduce barriers for women in agriculture
- To validate and expand upon this research findings
- To developing financial products accessible and beneficial to women in agriculture.
- To include data measuring about immigrant women in agriculture in Peru (thank you Jakub)

Next Steps

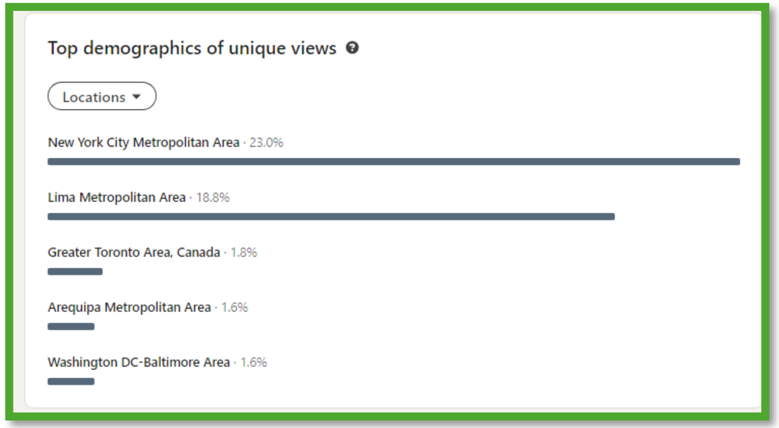
- New-data wrangling
- Advanced features
- Input (User interface) Thank you Jakub!
- Python for Tableau (Story Telling - Image Interface)
- Seaborn (**Statistical** Data Visualization)
- Geo-Spatial (Maps)
- Tensor Flow (Regressions)
- Keras (Neural Networks)

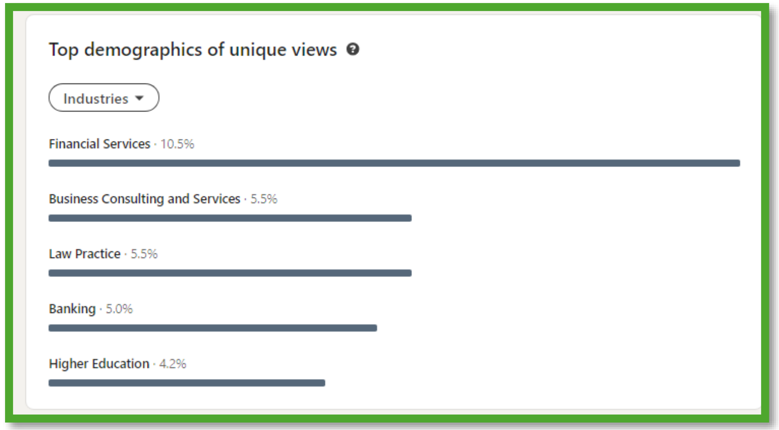
Audience (LinkedIn statistics)

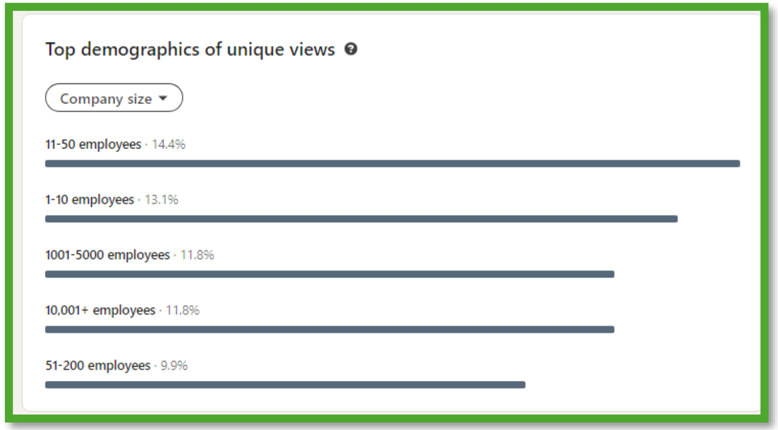




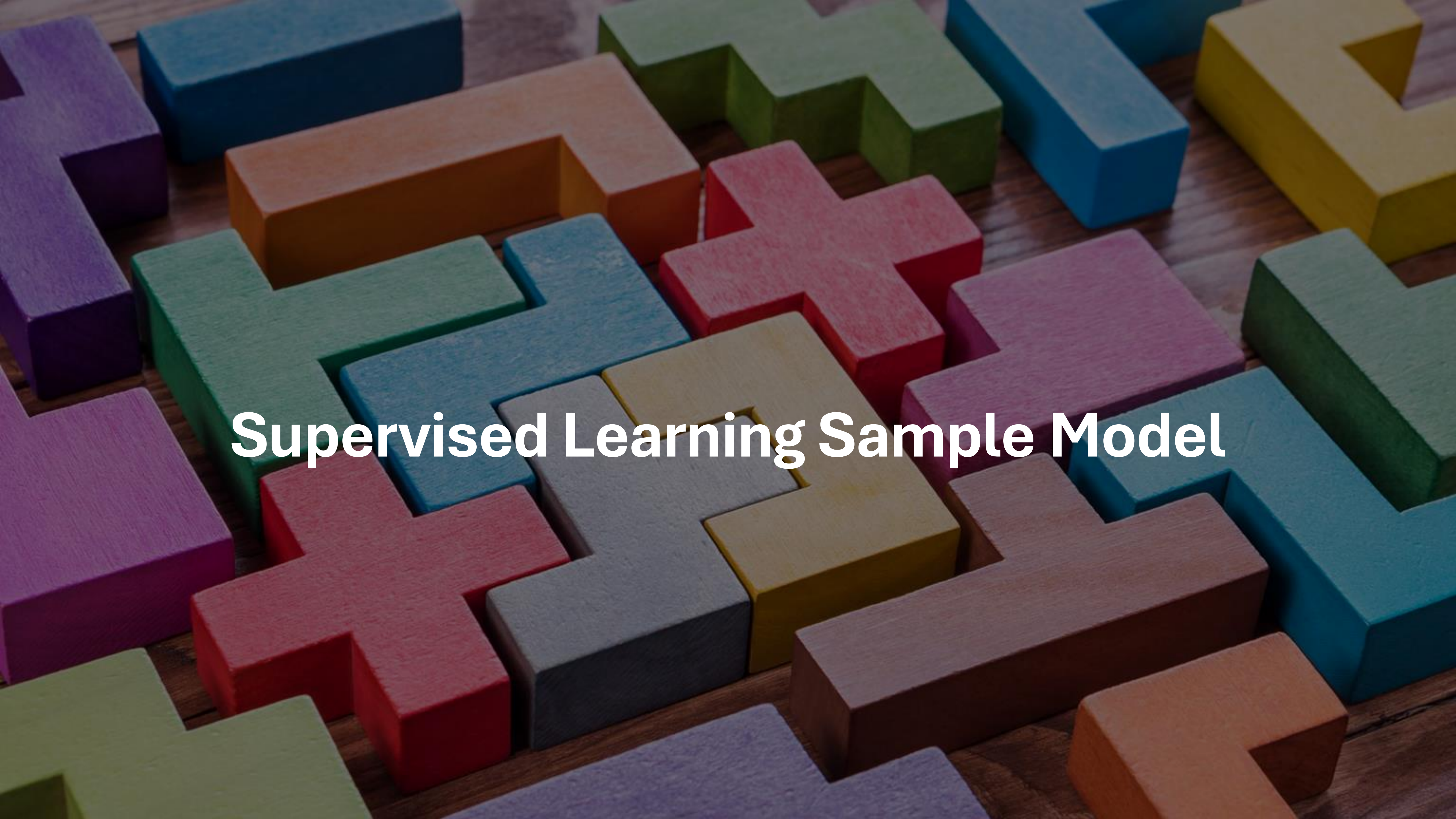
Audience (Top performing post statistics)







Supervised Learning Sample Model



Python Data Exploration & Feature Preparation: Jupyter Notebook via GitHub Repositories

Dataframes

Data source: Peru National Agricultural Census 2012

Datasets

Agriculture: Producer Profile

```
 #  Column                     Non-Null Count  Dtype
--- ------                     --------------  -----
 0  index                      52 non-null     int64
 1  year                       52 non-null     int64
 2  region_id                  52 non-null     int64
 3  region                     52 non-null     object
 4  gender_id                  52 non-null     int64
 5  gender                     52 non-null     object
 6  producers_thousands        52 non-null     float64
 7  granted_loans_thousands    52 non-null     float64
 8  illiteracy_percent         52 non-null     float64
 9  postgraduate_percent       52 non-null     float64
 10 primary_percent            52 non-null     float64
 11 secondary_percent          52 non-null     float64
 12 dont_need_loans_percent    52 non-null     float64
 13 high_interest_percent      52 non-null     float64
 14 no_collateral_percent      52 non-null     float64
 15 or_reasons_percent         52 non-null     float64
 16 red_tape_percent           52 non-null     float64
 17 will_not_get_it_percent    52 non-null     float64
 18 affiliation_percent        52 non-null     float64
 19 Spanish                    52 non-null     float64
 20 Quechua                    52 non-null     float64
 21 Aymara                     52 non-null     float64
 22 Amazonia                   52 non-null     float64
 23 requested_loans_percent    52 non-null     float64
 24 trained_thousands          52 non-null     float64
 25 purpose_sales              52 non-null     float64
 26 purpose_self_consumption   52 non-null     float64
 27 purpose_self_provision     52 non-null     float64
 28 purpose_animal_food        52 non-null     float64
```

Agriculture: Employment

```
 #  Column                     Non-Null Count  Dtype
--- ------                     --------------  -----
 0  index                      7140 non-null   int64
 1  year                       7140 non-null   int64
 2  region_id                  7140 non-null   float64
 3  region                     7140 non-null   object
 4  sector_id                  7140 non-null   int64
 5  sector                     7140 non-null   object
 6  gender_id                  7140 non-null   int64
 7  gender                     7140 non-null   object
 8  agriculture_employment     7140 non-null   float64
```

Agriculture: Main Indicators

```
 #  Column                     Non-Null Count  Dtype
--- ------                     --------------  -----
 0  index                      29 non-null     int64
 1  year                       29 non-null     int64
 2  region_id                  29 non-null     int64
 3  region                     29 non-null     object
 4  revenue_pen_2007           29 non-null     float64
 5  gdp_percent                29 non-null     float64
 6  added_value_PEN07          29 non-null     int64
 7  harvested_mill_ha          29 non-null     float64
 8  crops_mill_ha              29 non-null     float64
```

Agriculture: Natural Disasters

```
 #  Column                     Non-Null Count  Dtype
--- ------                     --------------  -----
 0  index                      470 non-null    int64
 1  year                       470 non-null    int64
 2  region_id                  470 non-null    int64
 3  region                     470 non-null    object
 4  crops_affected_by_disasters 470 non-null   float64
```

Common Category Columns

```
 #  Column                     Non-Null Count  Dtype
--- ------                     --------------  -----
 0  index                      52 non-null     int64
 1  year                       52 non-null     int64
 2  region_id                  52 non-null     int64
 3  region                     52 non-null     object
 4  gender_id                  52 non-null     int64
 5  gender                     52 non-null     object
```

Drop

```
 4  sector_id                  7140 non-null   int64
 5  sector                     7140 non-null   object
```

```
 #  Column                     Non-Null Count  Dtype
--- ------                     --------------  -----
 0  index                      7140 non-null   int64
 1  year                       7140 non-null   int64
 2  region_id                  7140 non-null   float64
 3  region                     7140 non-null   object
 6  gender_id                  7140 non-null   int64
 7  gender                     7140 non-null   object
```

```
 #  Column                     Non-Null Count  Dtype
--- ------                     --------------  -----
 0  index                      29 non-null     int64
 1  year                       29 non-null     int64
 2  region_id                  29 non-null     int64
 3  region                     29 non-null     object
```

```
 #  Column                     Non-Null Count  Dtype
--- ------                     --------------  -----
 0  index                      470 non-null    int64
 1  year                       470 non-null    int64
 2  region_id                  470 non-null    int64
 3  region                     470 non-null    object
```

Relevant Feature Columns

```
 6  producers_thousands        52 non-null     float64
 7  granted_loans_thousands    52 non-null     float64
 8  illiteracy_percent         52 non-null     float64
 9  postgraduate_percent       52 non-null     float64
 10 primary_percent            52 non-null     float64
 11 secondary_percent          52 non-null     float64
 12 dont_need_loans_percent    52 non-null     float64
 13 high_interest_percent      52 non-null     float64
 14 no_collateral_percent      52 non-null     float64
 15 or_reasons_percent         52 non-null     float64
 16 red_tape_percent           52 non-null     float64
 17 will_not_get_it_percent    52 non-null     float64
 18 affiliation_percent        52 non-null     float64
 19 Spanish                    52 non-null     float64
 20 Quechua                    52 non-null     float64
 21 Aymara                     52 non-null     float64
 22 Amazonia                   52 non-null     float64
 23 requested_loans_percent    52 non-null     float64
 24 trained_thousands          52 non-null     float64
 25 purpose_sales              52 non-null     float64
 26 purpose_self_consumption   52 non-null     float64
 27 purpose_self_provision     52 non-null     float64
 28 purpose_animal_food        52 non-null     float64
```

```
 8  agriculture_employment     7140 non-null   float64
```

```
 4  revenue_pen_2007           29 non-null     float64
 5  gdp_percent                29 non-null     float64
 6  added_value_PEN07          29 non-null     int64
 7  harvested_mill_ha          29 non-null     float64
 8  crops_mill_ha              29 non-null     float64
```

```
 4  crops_affected_by_disasters 470 non-null   float64
```

Features

Concatenation / Final Columns

```
 #  Column                     Non-Null Count  Dtype
--- ------                     --------------  -----
 0  index                      52 non-null     int64
 1  year                       52 non-null     int64
 2  region_id                  52 non-null     int64
 3  region                     52 non-null     object
 4  gender_id                  52 non-null     int64
 5  gender                     52 non-null     object
 6  producers_thousands        52 non-null     float64
 7  granted_loans_thousands    52 non-null     float64
 8  illiteracy_percent         52 non-null     float64
 9  postgraduate_percent       52 non-null     float64
 10 primary_percent            52 non-null     float64
 11 secondary_percent          52 non-null     float64
 12 dont_need_loans_percent    52 non-null     float64
 13 high_interest_percent      52 non-null     float64
 14 no_collateral_percent      52 non-null     float64
 15 or_reasons_percent         52 non-null     float64
 16 red_tape_percent           52 non-null     float64
 17 will_not_get_it_percent    52 non-null     float64
 18 affiliation_percent        52 non-null     float64
 19 Spanish                    52 non-null     float64
 20 Quechua                    52 non-null     float64
 21 Aymara                     52 non-null     float64
 22 Amazonia                   52 non-null     float64
 23 requested_loans_thousands  52 non-null     float64
 24 trained_thousands          52 non-null     float64
 25 purpose_sales              52 non-null     float64
 26 purpose_self_consumption   52 non-null     float64
 27 purpose_self_provision     52 non-null     float64
 28 purpose_animal_food        52 non-null     float64
 29 agriculture_employment     7140 non-null   float64
```

```
 #  Column                     Non-Null Count  Dtype
--- ------                     --------------  -----
 0  index                      29 non-null     int64
 1  year                       29 non-null     int64
 2  region_id                  29 non-null     int64
 3  region                     29 non-null     object
 4  revenue_pen_2007           29 non-null     float64
 5  gdp_percent                29 non-null     float64
 6  added_value_PEN07          29 non-null     int64
 7  harvested_mill_ha          29 non-null     float64
 8  crops_mill_ha              29 non-null     float64

 4  crops_affected_by_disasters 470 non-null   float64
```

Model Development Step by Step: Full Code *

Statistical Analysis

mean	std	minimum	25% Q	50% Q	75% Q	maximum

Methodology

Method	Supervised Learning	A type of machine learning where the model is trained on a labeled dataset to make predictions.
Technique	Random Forest	An ensemble learning method that operates by constructing multiple decision trees during training and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees.
Model	Trained Random Forest Model	A model that has been trained using the Random Forest technique on a specific dataset.
Tool	Scikit-learn	A Python library that provides simple and efficient tools for predictive data analysis, equipped to work with numerical tables or data frames.

Model Implementation

The main objective of the Random Forest model is to improve prediction accuracy by reducing overfitting of the model and handling large data with higher dimensionality. It does this by creating multiple decision trees and merging them together.

Formula

There isn't a specific formula for Random Forest like there is for some other models. Instead, it's a collection of decision trees, each created from a different subset of your data. The final prediction is made by averaging the predictions of each tree if it's a regression problem, or by majority voting if it's a classification problem.

Steps

Bootstrap the data: Create multiple subsets of the original dataset, selecting observations with replacement.

Create the Random Forest: For each new data subset, create a decision tree. The optimal split at each node is found from a random subset of features.

Make a prediction: Each individual tree in the Random Forest spits out a class prediction and the class with the most votes becomes the model's prediction.

*** Model Development Step by Step**

1. Data Preparation for Random Forest Model
2. Data Pre-Processing for Random Forest Model (new file)
3. Feature Selection & Importance
4. Model Training
5. Printing the Decision Tree
6. Performing a Prediction
7. Model Evaluation

Category vs Features

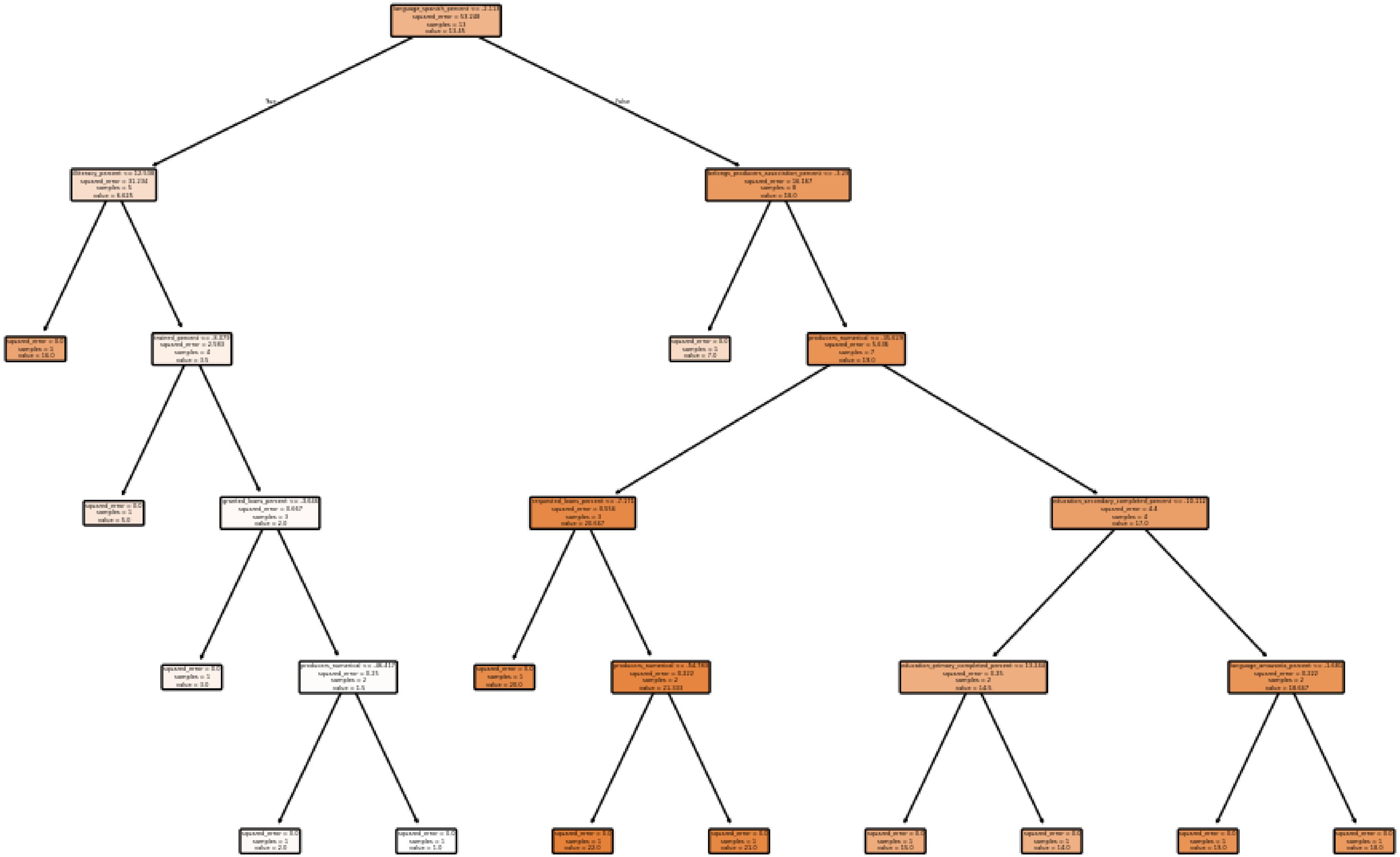
#	Column	Non-Null Count	Dtype	Final Feature
0	index	52 non-null	int64	category_index
1	year	52 non-null	int64	category_year
2	region_id	52 non-null	int64	category_region_id
3	region	52 non-null	object	category_region
4	gender_id	52 non-null	int64	category_gender_id
5	gender	52 non-null	object	category_gender
6	producers_thousands	52 non-null	float64	producers_numerical
7	granted_loans_percent	52 non-null	float64	granted_loans_percent
8	illiteracy_percent	52 non-null	float64	illiteracy_percent
9	postgraduate_percent	52 non-null	float64	education_postgraduate_completed_percent
10	primary_percent	52 non-null	float64	education_primary_completed_percent
11	secondary_percent	52 non-null	float64	education_secondary_completed_percent
12	dont_need_loans_percent	52 non-null	float64	don't_need_loans_percent
13	high_interest_percent	52 non-null	float64	high_interests_percent
14	other_reasons_percent	52 non-null	float64	other_reasons_percent
15	no_collateral_percent	52 non-null	float64	no_collateral_percent
16	red_tape_percent	52 non-null	float64	red_tape_percent
17	will_not_get_it_percent	52 non-null	float64	will_not_get_it_percent
18	affiliation_percent	52 non-null	float64	belongs_producers_association_percent
19	Spanish	52 non-null	float64	language_spanish_percent
20	Quechua	52 non-null	float64	language_quechua_percent
21	Aymara	52 non-null	float64	language_aymara_percent
22	Amazonia	52 non-null	float64	language_amazonia_percent
23	requested_loans_thousands	52 non-null	float64	requested_loans_percent
24	trained_thousands	52 non-null	float64	trained_percent
25	purpose_sales	52 non-null	float64	purpose_sales_percent
26	purpose_self_consumption	52 non-null	float64	purpose_self_consumption_percent
27	purpose_self_provision	52 non-null	float64	purpose_self_provision_percent
28	purpose_animal_food	52 non-null	float64	purpose_animal_food_percent
29	employment_agriculture_percent	7140 non-null	float64	employment_agriculture_percent

Feature Selection & Importance **

- category_region_id
- producers_numerical
- granted_loans_percent
- requested_loans_percent
- trained_percent
- belongs_producers_association_percent
- illiteracy_percent
- education_primary_completed_percent
- education_secondary_completed_percent
- employment_agriculture_percent
- language_spanish_percent
- language_quechua_percent
- language_aymara_percent
- language_amazonia_percent

**** Although the analysis includes numerical versions of this features, numerical columns have been excluded from this model**

Decision Tree



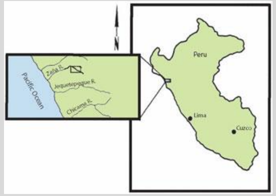
Supporting Research

Agriculture in Peru

- **Timeline:** 10 millennia of agricultural development in Peru
- **Demographics**: 25% of the population is dedicated to agriculture as of 2022

Gender Gap in Agriculture: Python Data Analysis: Jupyter Notebooks via GitHub Repositories

- Women in agriculture have limited access to training and financing
- Women in agriculture have a limited participation in producer associations



Agriculture in Peru: 9,240 – 5,500 years ago
Preceramic Adoption of Peanut, Squash, and Cotton in Northern Peru
Tom D. Dillehay , Jack Rossen, Thomas C. Andres, And David E. Williams

The early development of agriculture in the New World has been assumed to involve early farming in settlements in the Andes, but the record has been sparse. Peanut (Arachis sp.), squash (Cucurbita moschata), and cotton (Gossypium barbadense) macrofossils were excavated from archaeological sites on the western slopes of the northern Peruvian Andes. Direct radiocarbon dating indicated that these plants grew between 9240 and 5500 14C years before the present. These and other plants were recovered from multiple locations in a tropical dry forest valley, including household clusters, permanent architectural structures, garden plots, irrigation canals, hoes, and storage structures. These data provide evidence for early use of peanut and squash in the human diet and of cotton for industrial purposes and indicate that horticultural economies in parts of the Andes took root by about 10,000 years ago.

Agriculture in Peru (2,000 BCE – 1,400 CE): Irrigation and Land Use on the Arid North Coast of Peru: Assessing Ancient Agricultural Systems Through Drone Photography, Soil Analysis, and Local Knowledge
Authors: C. Prado, J. Eerkens, R. Beresford-Jones, and E. Van Valkenburgh

This paper explores the historical development of agriculture along Peru's arid north coast, focusing on the prehispanic timeline and agricultural products of different cultures. Intensive irrigation-based farming began in the second millennium BCE, featuring early canals and check dams. Early Andean cultures grew crops like squash, beans, and cotton. From the first millennium BCE to the first millennium CE, advanced water management techniques were developed, crucial for handling the El Niño Southern Oscillation (ENSO). Key crops included maize, beans, and manioc. Significant hydraulic engineering advancements and irrigation network expansions occurred from the first millennium CE to the 15th century. The Moche civilization (100-800 CE) built extensive canal systems, cultivating maize, beans, squash, and peanuts. The Chimu civilization (11th-15th century) further developed these systems, creating interconnected canals for diverse crops such as maize, cotton, and quinoa. These innovations supported large populations and complex societies. The paper concludes that the prehispanic agricultural timeline in Peru showcases a continuous evolution of water management and farming practices. By examining the specific crops and techniques used by different cultures, we gain insights into the adaptability and resilience of ancient agricultural systems. These historical practices offer valuable lessons for sustainable agriculture in arid regions globally, demonstrating effective responses to environmental challenges

Agriculture during the Colony (1681 – 1800 CE)
Title: Crecimiento Económico en el Espacio Peruano
Carlos Newland, Universidad Argentina de la Empresa
John Coatsworth, Rockefeller Center for Latin American Studies, Harvard University

In-depth analysis of the evolution of agriculture in Peru from 1681 to 1800. The authors highlight a period of economic crisis and agricultural decline from 1681 to 1750. This was followed by an improvement and overall growth in agriculture from 1750 to 1800.
In conclusion, despite the initial collapse, there was a subsequent expansion of agricultural production in the 18th century. This expansion was not homogeneous across regions, with Lima experiencing a decline. However, the overall trend suggests stability or improvement in per capita agricultural production and likely increases in real wages for the region, except in Lima.

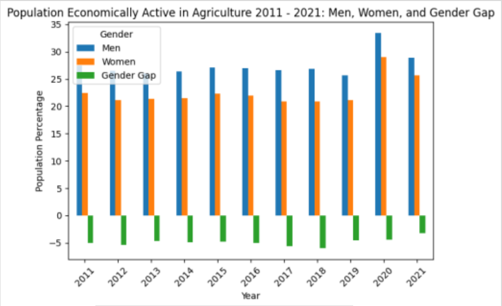
Agriculture during the Republic (1,875 – 1,933)
Title: Plantation Agriculture and Social Control in Northern Peru, 1875–1933
Author: Michael J. Gonzales

The author explores the development of plantation agriculture in Northern Peru from 1875 to 1933. Beginning with the economic and political transformation during the 1860s and 1870s, marked by the decline of the guano boom and the rise of coastal agriculture. By the late 19th century, sugarcane plantations had become significant economic entities, driven by technological advancements and the influx of capital from former guano traders.
In the early 20th century, the sugar industry continued to expand, with plantations adopting modern agricultural practices and machinery.
Author concludes with the impact of the War of the Pacific and the subsequent recovery of Peru's agricultural sector, as wells as the transition from traditional to modern practices, reflecting broader economic and social changes in Northern Peru.

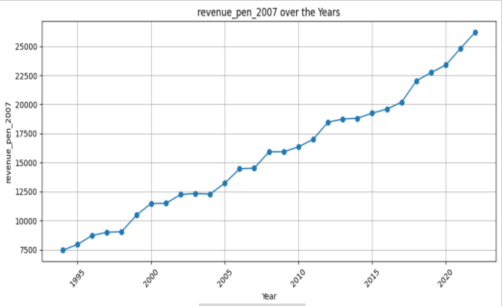
Agriculture in Peru: Data Analysis: 1994 - 2022

Gender Gap in Agriculture in Peru: Data Analysis: 2012

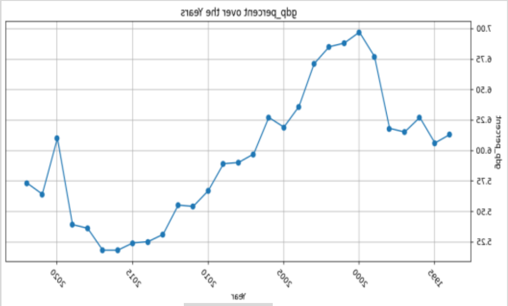
Total Population Dedicated to Agriculture



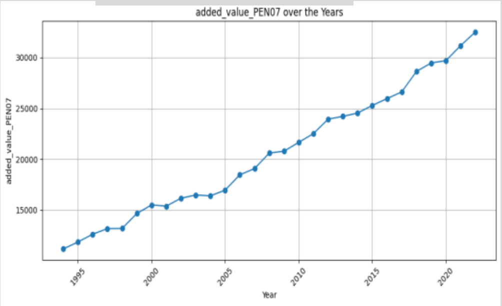
Total Agricultural Revenue



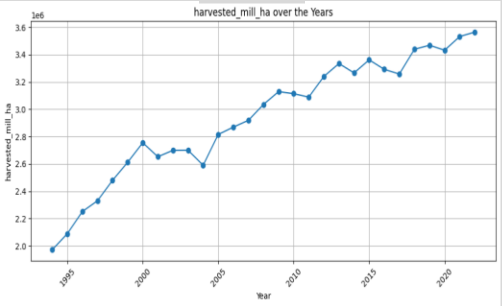
Agricultural GDP



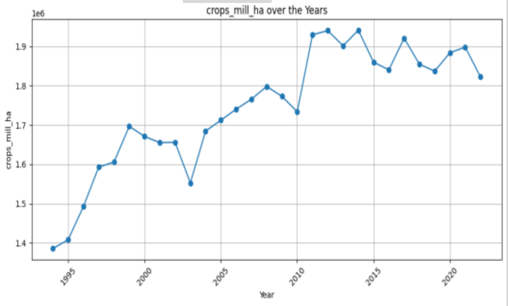
Agricultural Added Value



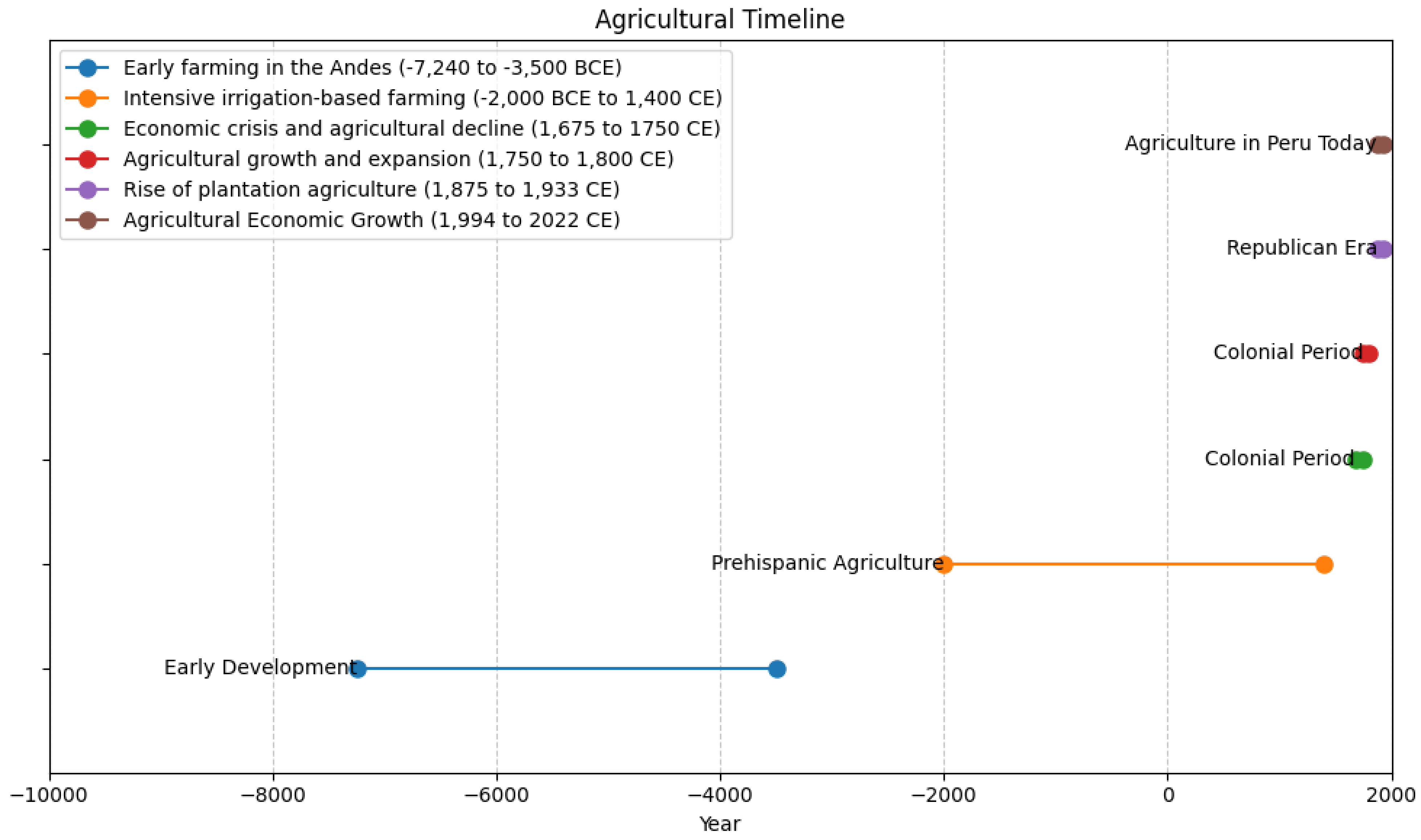
Harvest



Crops



Jose Carlo Burga, July, 2024



"Smith, J. (2005). Early Farming in the Andes. Archaeological Journal, 30(2), 45-58.",
"Garcia, R. (2010). Intensive Irrigation-Based Farming in Prehispanic Agriculture. Journal of Agriculture, 15(3), 120-135.",
"Brown, A. (2015). Economic Crisis and Agricultural Decline in the Colonial Period. Economic History Review, 42(4), 210-225.",
"Jones, M. (2018). Agricultural Growth during Colonial Expansion. Journal of Economic Development, 50(1), 80-95.",
"Rodriguez, P. (2021). Rise of Plantation Agriculture in the Republican Era. Agricultural History, 25(3), 300-315.",
"Ministry of Agriculture (2022). Economic Growth in Agriculture Today. Annual Report, Ministry of Agriculture."

Referential Research: Finance and Training in Peru

	Machine Learning **Access to Credit and Credit Risk**	**Machine Learning** **Financial Inclusion (Services & Literacy)**	**Gender Equality in Peru**
	Study on Peruvian Microfinance Institution	Financial Inclusion in Peru	Gender Equality in Peru: Unleashing the Potential of Women (2022)
Summary	This study was conducted at a Peruvian microfinance institution specializing in rural microcredits. The authors proposed a model for assessing microcredit applications using machine learning techniques. The goal was to improve the **assertiveness of the credit granting process** and **reduce the default rate**.	This paper by Rocío Maehara et al. explores the application of machine learning (ML) methods to assess **financial inclusion in Peru**. The study uses data from the National Survey of Demand for **Financial Services** and **Financial Literacy** 2019, covering a sample of 1205 Peruvian citizens.	This OECD's report outlines significant data on women's access to training in the agricultural sector. The report highlights that women in Peru's agricultural sector face substantial barriers to accessing training, which impacts their productivity and economic opportunities.
Methods	Data Pre-processing, Cross-validation, Supervised Learning	Data Pre-processing, Grid Search Procedure, Supervised Learning	Surveys and Questionnaires, Statistical Analysis, Regression Models, Qualitative Interviews, Focus Groups.
Techniques	Handling missing data, Normalizing variables, One Hot coding	10-fold cross-validation	n/a
Models	Logistic Regression (LR), Random Forest (RF), Support Vector Machine (SVM), Artificial Neural Network (ANN), Decision Tree (dTree), k-Nearest Neighbors (kNN)	Logistic Regression (LR), Artificial Neural Networks (ANNs), Decision Trees (DTs), Random Forest (RF), XGBoost, Support Vector Machine with RBF kernel (SVC RBF)	n/a
Tools	Scikit Learn, Keras, Pandas, Numpy, Matplotlib	Not explicitly mentioned	n/a

Comparative Hierarchical Methodological Decision-Making Matrix

	template model developed – scalable						selected	
Method	Supervised Learning	A type of machine learning where the model is trained on a labeled dataset to make predictions.	Supervised Learning	A type of machine learning where the model is trained on a labeled dataset to make predictions.	Supervised Learning	A type of machine learning where the model is trained on a labeled dataset to make predictions.	Supervised Learning	A type of machine learning where the model is trained on a labeled dataset to make predictions.
		Python for Tensor Flow Finance		**Overfitting?**		**Python for Impact Finance**		**Python Keras**
Technique	Random Forest	An ensemble learning method that operates by constructing multiple decision trees during training and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees.	Support Vector Machines (SVM)	A set of supervised learning methods used for classification, regression and outliers detection.	Logistic Regression	A statistical model that uses a logistic function to model a binary dependent variable.	Neural Networks	A series of algorithms that endeavors to recognize underlying relationships in a set of data through a process that mimics the way the human brain operates.
Model	Trained Random Forest Model	A model that has been trained using the Random Forest technique on a specific dataset.	Trained SVM Model	A model that has been trained using the SVM technique on a specific dataset.	Trained Logistic Regression Model	A model that has been trained using the Logistic Regression technique on a specific dataset.	Trained Neural Network Model	A model that has been trained using the Neural Network technique on a specific dataset.
Tool	Scikit-learn	A Python library that provides simple and efficient tools for predictive data analysis, equipped to work with numerical tables or data frames.	Scikit-learn	A Python library that provides simple and efficient tools for predictive data analysis, equipped to work with numerical tables or data frames.	Scikit-learn	A Python library that provides simple and efficient tools for predictive data analysis, equipped to work with numerical tables or data frames.	TensorFlow	An open-source platform for machine learning that provides a comprehensive ecosystem of tools, libraries, and community resources for developing and deploying ML models.

References:

1. Kotsiantis, S. B. (2007). Supervised Machine Learning: A Review of Classification Techniques. Informatica, 31, 249-268.
2. Breiman, L. (2001). Random Forests. Machine Learning, 45(1), 5-32.
3. Cortes, C., & Vapnik, V. (1995). Support-Vector Networks. Machine Learning, 20(3), 273-297.
4. Cox, D. R. (1958). The Regression Analysis of Binary Sequences. Journal of the Royal Statistical Society: Series B (Methodological), 20(2), 215-242.
5. Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep Learning. MIT Press.
6. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... & Vanderplas, J. (2011). Scikit-learn: Machine Learning in Python. Journal of Machine Learning Research, 12, 2825-2830.
7. Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., ... & Kudlur, M. (2016). TensorFlow: A System for Large-Scale Machine Learning. In OSDI (Vol. 16, pp. 265-283).

Data from original csv files into sql tables, and from sql tables into a database 2017, 2018, 2019, 2021, 2022

Step	Method	Description	Example Code
1	Databases	Data is extracted from a database using SQL queries or a database API.	import sqlite3 import pandas as pd conn = sqlite3.connect('database.db') df = pd.read_sql_query("SELECT * FROM table_name", conn)
2	Web Scraping	Data is extracted from a website using web scraping tools.	import requests from bs4 import BeautifulSoup response = requests.get("https://www.website.com") soup = BeautifulSoup(response.content, 'html.parser') data = soup.find_all('div', class_='class-name')
3	APIs	Data is accessed in a structured format using APIs provided by websites and platforms.	import requests response = requests.get("https://api.website.com/data") data = response.json()
4	Surveys and Questionnaires	Data is collected using surveys or questionnaires.	N/A
5	CSV/Excel Files	Data is loaded from CSV or Excel files into a DataFrame.	import pandas as pd df = pd.read_csv('data.csv')
6	Preexisting Datasets	Data is collected from preexisting datasets available on the internet.	N/A

References:

Garcia-Molina, H., Ullman, J. D., & Widom, J. (2009). Database Systems: The Complete Book. Pearson.
Mitchell, R. (2015). Web Scraping with Python: Collecting More Data from the Modern Web. O'Reilly Media.
Pautasso, C., Zimmermann, O., & Leymann, F. (2008). Restful Web Services vs. Big Web Services: Making the Right Architectural Decision. In Proceedings of the 17th International Conference on World Wide Web (pp. 805-814).
Fink, A. (2013). How to Conduct Surveys: A Step-by-Step Guide. Sage Publications.
McKinney, W. (2012). Python for Data Analysis: Data Wrangling with Pandas, NumPy, and IPython. O'Reilly Media.
Kelleher, J. D., Mac Namee, B., & D'Arcy, A. (2015). Fundamentals of Machine Learning for Predictive Data Analytics: Algorithms, Worked Examples, and Case Studies. MIT Press.

Mean to “calculate” a missing 2020 dataset “period”

Statistic	Description	Citation
Mean	The average value of a dataset, calculated by summing all values and dividing by the number of values.	“Python for Data Analysis” by Wes McKinney1
Standard Deviation (std)	A measure of the amount of variation or dispersion in a set of values.	“Python Data Science Handbook” by Jake VanderPlas2
25th Percentile (25Q)	The value below which 25% of the data falls. Also known as the first quartile.	“Statistics with Python” by Peter Bruce and Andrew Bruce3
50th Percentile (50Q)	The median value, which separates the higher half from the lower half of the data.	“Python for Data Analysis” by Wes McKinney1
75th Percentile (75Q)	The value below which 75% of the data falls. Also known as the third quartile.	“Statistics with Python” by Peter Bruce and Andrew Bruce3
Median	The middle value of a dataset when it is ordered from least to greatest.	“Python Data Science Handbook” by Jake VanderPlas2
Minimum	The smallest value in a dataset.	“Python for Data Analysis” by Wes McKinney1
Maximum	The largest value in a dataset.	“Python for Data Analysis” by Wes McKinney1

Skew Reduction Techniques

Technique	Description	Example Code Snippet	Citation
Log Transformation	Reduces positive skew by applying a logarithmic transformation.	df['column_name'] = np.log(df['column_name'] + 1)	Osborne, J. W. (2010). Improving your data transformations: Applying the Box-Cox transformation. Practical Assessment, Research, and Evaluation, 15(12), 1-9.
Square Root Transformation	Reduces skew by applying a square root transformation.	df['column_name'] = np.sqrt(df['column_name'])	Osborne, J. W. (2010). Improving your data transformations: Applying the Box-Cox transformation. Practical Assessment, Research, and Evaluation, 15(12), 1-9.
Box-Cox Transformation	Flexible transformation that can handle both positive and negative skew.	df['column_name'], _ = stats.boxcox(df['column_name'] + 1)	Box, G. E. P., & Cox, D. R. (1964). An analysis of transformations. Journal of the Royal Statistical Society: Series B (Methodological), 26(2), 211-243.
Winsorization	Limits extreme values to reduce the effect of outliers.	df['column_name'] = winsorize(df['column_name'], limits=[0.05, 0.05])	Tukey, J. W. (1962). The future of data analysis. The Annals of Mathematical Statistics, 33(1), 1-67

Programming Paradigm

Programming Paradigm	Definition	Characteristics	Examples
Imperative Programming	Focuses on how to achieve a task by using statements that change the program’s state.	Involves loops, variables, and explicit control flow. Simple to implement. Cannot efficiently solve complex problems.	Python
Declarative Programming	Emphasizes what needs to be accomplished, rather than specifying how to achieve it.	Focuses on expressing the problem domain. Well-suited for complex problems or parallel execution.	SQL (when used in Python via libraries like SQLite), HTML, and CSS (when used in web development frameworks like Flask or Django).
Functional Programming	Revolves around using functions to develop software. It relies on expressions and declarations instead of statements.	Avoids shared states, mutable data, and side effects. Efficient for certain problem domains.	Functional constructs in Python (like lambda functions, map, filter, and reduce).
Procedural Programming	A type of imperative programming that organizes a program into smaller parts called methods or procedures. These methods are used for code reusability.	Divides the program into logical units. Widely used due to ease of writing and interpretation.	Python
Structured Programming	Employs structured control flow constructs (such as loops and conditionals), block structures, and subroutines.	Improves clarity, quality, and development time. Avoids “goto” statements.	Python

Citations

Lutz, M. (2013). Learning Python (5th ed.). O’Reilly Media. Beazley, D. M. (2009). Python Essential Reference (4th ed.). Addison-Wesley Professional.

Grinberg, M. (2018). Flask Web Development: Developing Web Applications with Python (2nd ed.). O’Reilly Media. Ford, M. (2015). Learning SQL (3rd ed.). O’Reilly Media.

Lutz, M. (2013). Learning Python (5th ed.). O’Reilly Media. Hetland, M. L. (2005). Beginning Python: From Novice to Professional. Apress.

Lutz, M. (2013). Learning Python (5th ed.). O’Reilly Media. Hetland, M. L. (2005). Beginning Python: From Novice to Professional. Apress.

Lutz, M. (2013). Learning Python (5th ed.). O’Reilly Media. Beazley, D. M. (2009). Python Essential Reference (4th ed.). Addison-Wesley Professional.

Supervised Learning (Random Forest) Predictive Model

	Description	Code
1.A	**Data Collection**	https://github.com/jcburga/python_final_research/blob/main/concatenate_clean_agriculture_producer_employment.csv
1.B	**Data Collection**	https://github.com/jcburga/python_final_research/blob/main/ml_producer_profile_selected_features.csv
2	**Data Exploration:**	https://github.com/jcburga/python_final_research/blob/main/%232A_ml_producer_exploration.ipynb
3	**Data Statistical Analysis:**	https://github.com/jcburga/python_final_research/blob/main/%231_ml_producer__statistical.ipynb
	Data Statistical Visualization:	Pending (via Seaborn)
4	**Data Pre-Processing:**	https://github.com/jcburga/python_final_research/blob/main/%232B_ml_producer_preparation.ipynb
5	**Feature Selection & Importance**	https://github.com/jcburga/python_final_research/blob/main/%233B_ml_producer_feature_selection_importance.ipynb
6	**Model Training:**	https://github.com/jcburga/python_final_research/blob/main/%234_5_6_ml_producer_train_predict_evaluate.ipynb
7	**Decision Tree Printing**	
8	**Prediction:**	
9	**Model Evaluation:**	

References

1	Data Collection:	Kelleher, J. D., Mac Namee, B., & D'Arcy, A. (2015). Fundamentals of Machine Learning for Predictive Data Analytics: Algorithms, Worked Examples, and Case Studies. MIT Press.
2	Data Preprocessing:	Garcia, S., Luengo, J., & Herrera, F. (2015). Data Preprocessing in Data Mining. Springer.
3	Feature Selection:	Guyon, I., & Elisseeff, A. (2003). An Introduction to Variable and Feature Selection. Journal of Machine Learning Research, 3, 1157-1182.
4	Model Training:	Bishop, C. M. (2006). Pattern Recognition and Machine Learning. Springer.
5	Model Evaluation:	Sokolova, M., & Lapalme, G. (2009). A Systematic Analysis of Performance Measures for Classification Tasks. Information Processing & Management, 45(4), 427-437.
6	Prediction:	Hastie, T., Tibshirani, R., & Friedman, J. (2009). The Elements of Statistical Learning: Data Mining, Inference, and Prediction. Springer.