RAPPORT DATAVIZ

Challenge-Edition 2023

RESUME

Comprendre les facteurs qui peuvent améliorer la production agricole potentielle par hectare ou par tête d'animal, en dehors de toute aide au pays de Basque.

Ferol TATANG FOMEKON

Élève ingénieure en Big Data et IA.

Table des matières

Introduction

L'histoire de l'agriculture dans le Pays Basque entre 1970 et 2020 est marquée par une évolution de l'agriculture traditionnelle vers des exploitations plus modernes et spécialisées. Depuis lors, le Pays Basque a subi d'importants recensements sur son agriculture par l'INSEE (Institut National de la Statistique et des Études Économiques). De ce fait, le Pays Basque a créé son portail de données ouvertes sur l'agriculture nommé "ZABAL Open Data Agriculture". Ainsi, le Pays Basque pourrait désormais se poser des questions sur ces données et y trouver une solution afin de les comparer à celles de l'INSEE. Partant de cette optique, il nous a été soumis des données très fiables issues de ZABAL auxquelles nous nous sommes posé des questions très pertinentes, à savoir : quels sont les facteurs qui expliquent au mieux la production agricole ? Peut-on prédire la production agricole au fil du temps ? Ainsi, tout au long de ce document, nous allons y apporter des réponses aux différentes questions.

I. Analyse Global

Une analyse globale des données consiste à observer le comportement des données de façon brute, sans y apporter de changement. Ainsi, l'observation globale des données nous montre que le jeu de données soumis à notre étude ne possède aucune ligne dupliquée et compte exactement 2561 enregistrements et 23 colonnes, dont il semblerait que nous ayons 7 colonnes de types catégoriels et le reste de types numériques (figure 1). Par ailleurs, l'observation de la description de ces données semble nous révéler qu'entre 1972 et 2020, le Pays Basque a compté exactement 486 productions agricoles (pbs) avec un effectif à temps plein de 640 (etp). Cependant, il semblerait aussi que nous ayons 882 unités de gros bétail (ugb) pour une superficie agricole utilisée de 563 hectares (sau_ha) (figure 2). Toutes ces remarques nous poussent à penser à l'existence de valeurs aberrantes dans nos données.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2561 entries, 0 to 2560
Data columns (total 23 columns):
 #   Column                  Non-Null Count  Dtype
---  ------                  --------------  -----
 0   echelle                 2561 non-null   object
 1   categorie               2561 non-null   object
 2   type                    2355 non-null   object
 3   annee                   2463 non-null   float64
 4   valeur                  2223 non-null   float64
 5   sau_tot_ha              66 non-null     float64
 6   sau_moy_ha              66 non-null     float64
 7   sau_ha                  563 non-null    float64
 8   ugb                     882 non-null    float64
 9   etp                     640 non-null    float64
 10  pbs                     486 non-null    float64
 11  age_5                   262 non-null    object
 12  tetes                   396 non-null    float64
 13  bio                     154 non-null    float64
 14  surface                 308 non-null    float64
 15  irrigation              297 non-null    float64
 16  nombre_exploitation_bio 154 non-null    float64
 17  surface_bio             154 non-null    float64
 18  otefdd_coef17           626 non-null    float64
 19  code_insee              316 non-null    float64
 20  commune                 316 non-null    object
 21  Geo Shape               326 non-null    object
 22  geo_point_2d            316 non-null    object
dtypes: float64(16), object(7)
memory usage: 460.3+ KB
```

```
la taille des données est :(2561, 23)
Le nombre de lignes dupliquées est : 0
```

Figure 1

1 to 8 of 8 entries Filter

index	annee	valeur	sau_tot_ha	sau_moy_ha	sau_ha	ugb	etp	pbs	tetes
count	2463.0	2223.0	66.0	66.0	563.0	882.0	640.0	486.0	396.0
mean	2015.2525375558262	110.35942420152946	22334.180606060603	19.669190650375054	2918.6426110124335	5458.289913832199	96.9858140284747	6626.0942648041155	21255.141414141413
std	6.603021046275816	742.9047211867107	32913.50332910186	9.651133332479212	9047.400761301307	17546.19597043908	693.2046922213145	17764.688862764666	71492.67551111033
min	1970.0	-999.0	267.54	4.965420560747663	-999.0	-999.0	-999.0	-999.0	-999.0
25%	2010.0	6.0	7784.1575	11.771747921349684	62.900000000000006	19.2705	9.2079694	188.8612725	173.75
50%	2020.0	29.0	12892.755000000001	17.414570045021037	456.05	533.5775	38.503570800000006	1373.960915	999.0
75%	2020.0	126.0	21890.6325	27.583906233766232	2281.165	3568.9845	152.10620774999998	5720.79082	7285.5
max	2020.0	12176.0	137034.36	39.74463203463203	105513.28	225911.05	5832.8264268	161277.958068	716237.0

Figure 2

II. Nettoyage des données

Au vu du paragraphe précédent, nous avons constaté que nos données contiennent des valeurs manquantes. Raison de plus pour les nettoyer afin d'apporter une meilleure analyse sur la production agricole au pays basque. De ce fait, nous avons estimé que nous ne pouvons pas nous permettre de supprimer les valeurs manquantes, sinon nous allons perdre suffisamment d'informations. C'est pourquoi nous avons remplacé certaines valeurs manquantes numériques par leur moyenne, leur maximum et leur médiane. La moyenne est une mesure statistique qui représente la valeur centrale ou typique d'un ensemble de données, tandis que le quantile est une mesure statistique utilisée pour diviser un ensemble de données en sous-groupes égaux ou inégaux en termes de taille. La figure 3 nous montre le choix de la médiane pour deux variables. Cependant, pour la variable "âge", nous nous sommes intéressés à la moyenne des valeurs extrêmes, puis nous avons remplacé ces valeurs manquantes par la moyenne globale. Par ailleurs, concernant les coordonnées géographiques, après avoir observé la position exacte de ces points sur le globe terrestre (figure 4 et 5), nous avons estimé qu'il serait judicieux de s'intéresser à la longitude et à la latitude. Ensuite, nous avons remplacé ces valeurs manquantes par leurs moyennes.

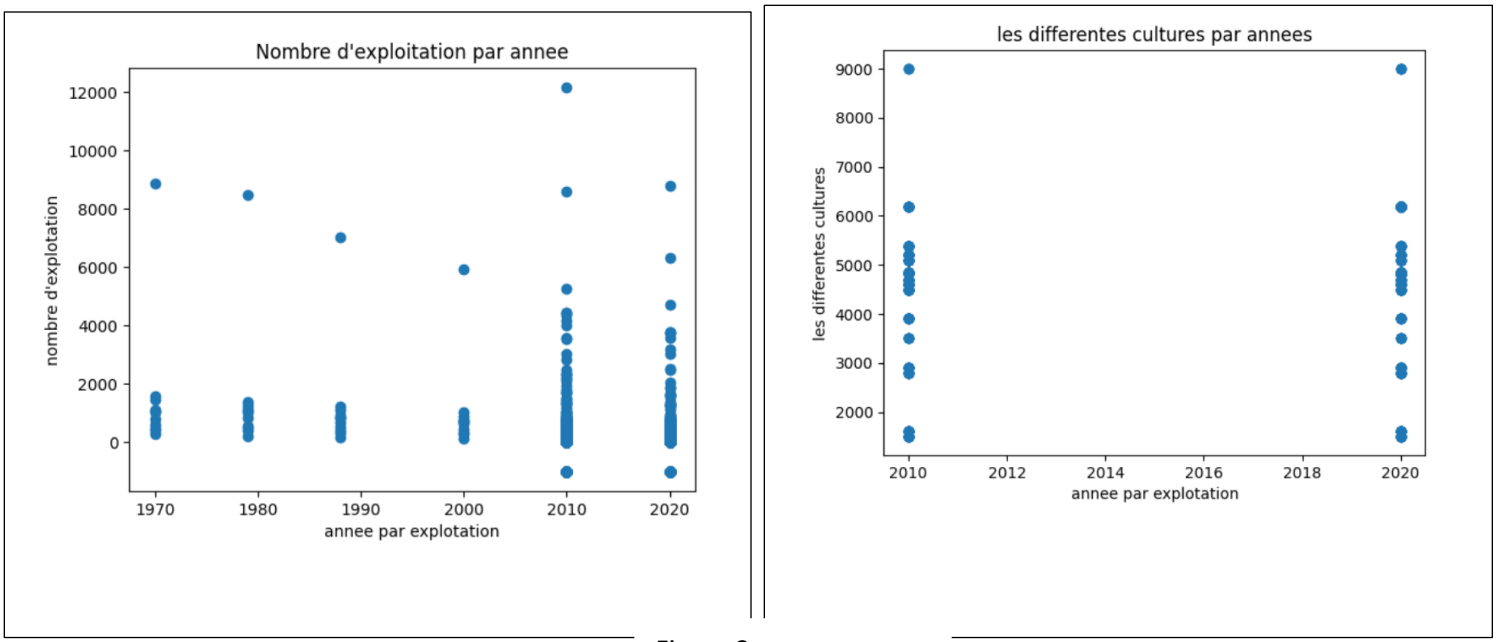


Figure 3

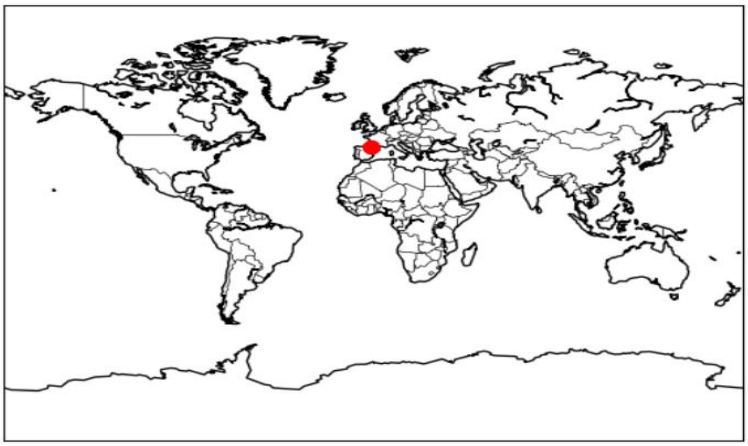


Figure 4

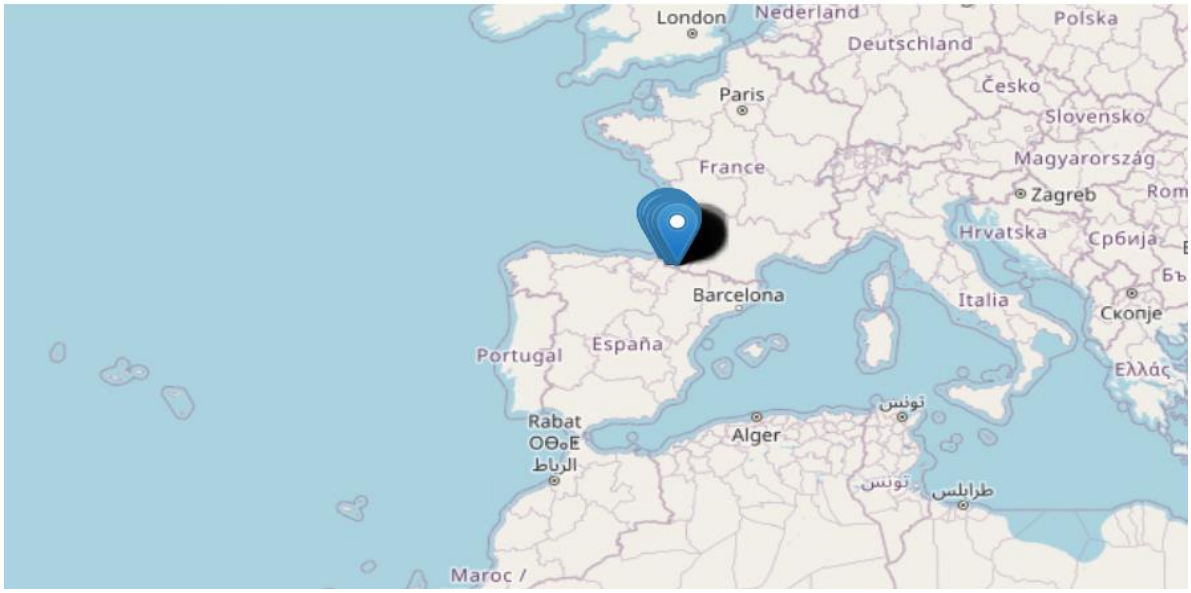


Figure 5

III. Exploration des données

Après nettoyage des données, nous pouvons faire une première description et constater que nous avons bien 2561 productions agricoles avec une moyenne de 6626,09 productions agricoles au total entre 1970 et 2020 au Pays de Basque (figure 6). Par ailleurs, une observation globale montre que l'agriculteur le plus jeune aurait presque 13 ans et le plus âgé 85 ans. La figure 6 montre que les données nous parlent désormais de manière plus cohérente et commencent à expliquer la production agricole au Pays Basque (pbs).

```
6]  df_rga.describe()
```

1 to 8 of 8 entries Filter

index	annee	valeur	sau_tot_ha	sau_moy_ha	sau_ha	ugb	etp	pbs	age_5
count	2561.0	2561.0	2561.0	2561.0	2561.0	2561.0	2561.0	2561.0	2561.0
mean	2015.43420538852	99.62163217493166	22334.180606060603	19.669190650375054	2918.6426110124335	5458.289913832199	96.98581402847472	6626.0942648041155	47.99618320610687
std	6.5391597919275215	692.6738151567022	5244.581116743484	1.5378536622061643	4239.083812653973	10293.212699663569	346.3314571703993	7732.296378248583	6.14395969197104
min	1970.0	-999.0	267.54	4.965420560747663	-999.0	-999.0	-999.0	-999.0	12.5
25%	2010.0	8.0	22334.180606060603	19.669190650375054	2918.6426110124335	2736.066	96.9858140284747	6626.0942648041155	47.99618320610687
50%	2020.0	29.0	22334.180606060603	19.669190650375054	2918.6426110124335	5458.289913832199	96.9858140284747	6626.0942648041155	47.99618320610687
75%	2020.0	101.0	22334.180606060603	19.669190650375054	2918.6426110124335	5458.289913832199	96.9858140284747	6626.0942648041155	47.99618320610687
max	2020.0	12176.0	137034.36	39.74463203463203	105513.28	225911.05	5832.8264268	161277.958068	85.0

Show 25 per page

Like what you see? Visit the data table notebook to learn more about interactive tables.

Figure 6

A. Analyse des variables de types numériques

La corrélation est un indicateur de la force et de la direction de la relation linéaire entre les variables. Mais pour effectuer une corrélation, nous devons nous assurer que les variables suivent une distribution normale. Ainsi, en observant la figure 7, il semblerait que les données suivent une distribution normale. De ce fait, nous avons donc effectué un test de Shapiro-Wilk sur les valeurs numériques. Nous obtenons une statistique de test de 0.33 et une p-valeur de 0.0. La valeur p obtenue est très faible (inférieure à 0,05), ce qui suggère un rejet de l'hypothèse nulle selon laquelle les données suivent une distribution normale. Cependant, nos données sont assez grandes, donc nous ne pouvons pas nous focaliser uniquement sur le test de Shapiro-Wilk pour rejeter l'hypothèse de la distribution normale.

Par la suite, nous avons effectué une matrice de corrélation sur nos données (figure 8), ce qui nous a permis de visualiser la corrélation linéaire entre plusieurs variables et la production agricole (pbs) au Pays Basque. Ainsi, la figure 9 nous montre davantage cette corrélation. Cependant, d'autres variables semblent ne pas expliquer significativement la production agricole par hectare ou par tête d'animal au Pays Basque, comme nous pouvons l'observer dans la figure 10.

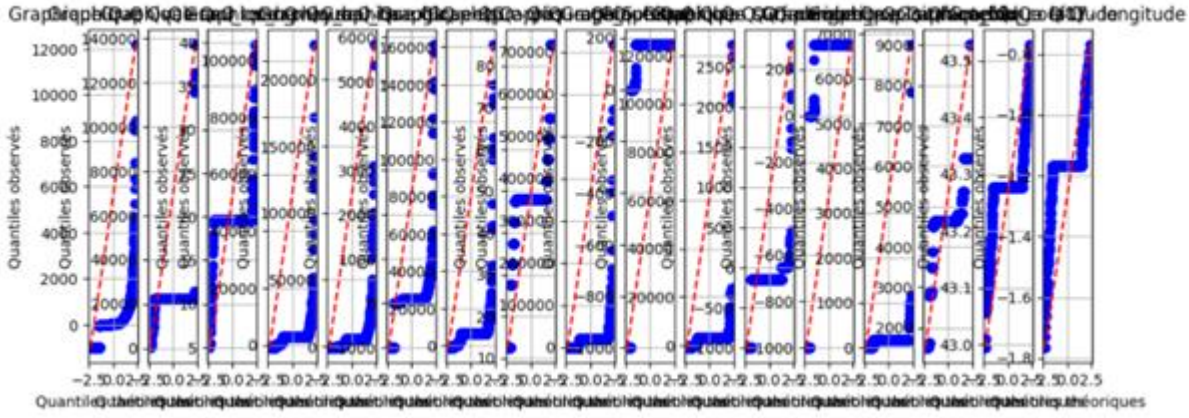
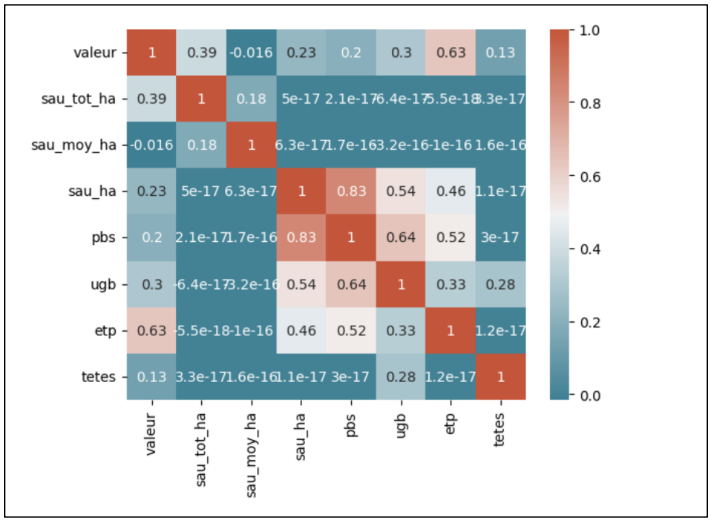
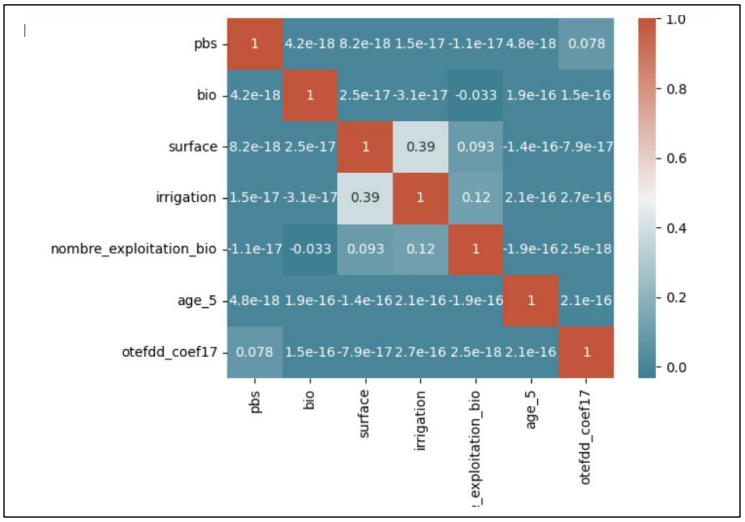


Figure 7





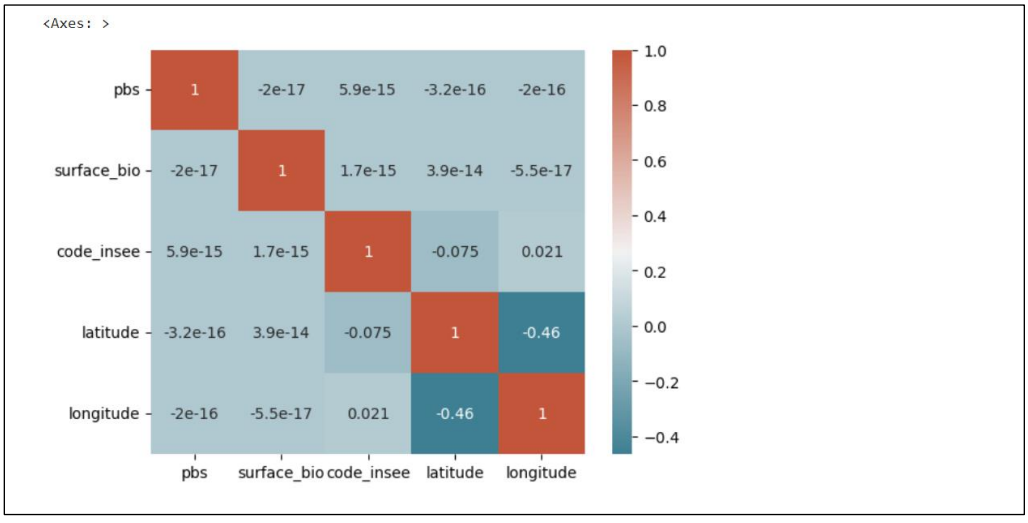
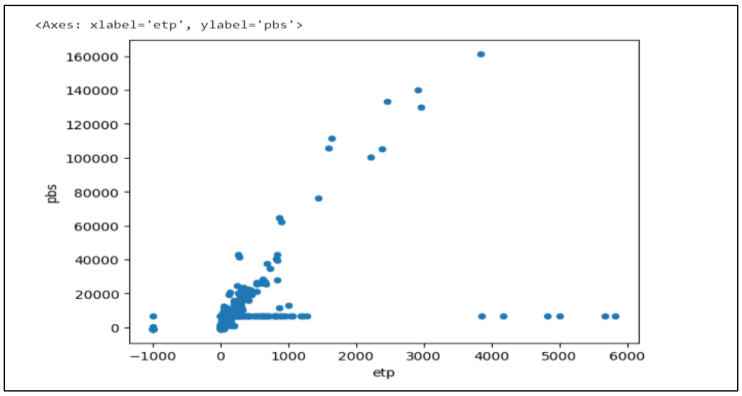
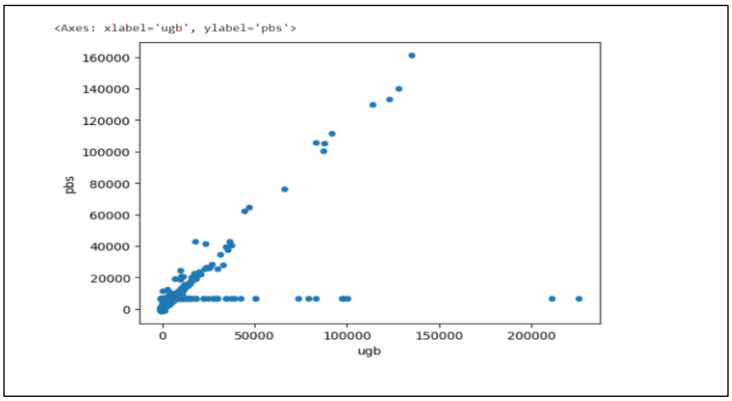
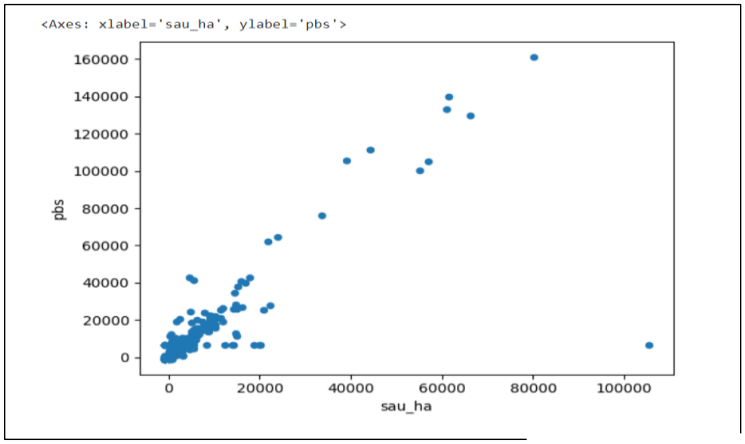


Figure 8







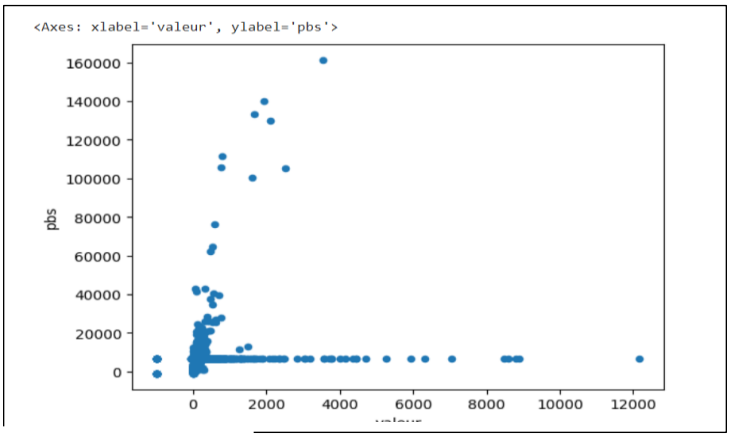


Figure 9

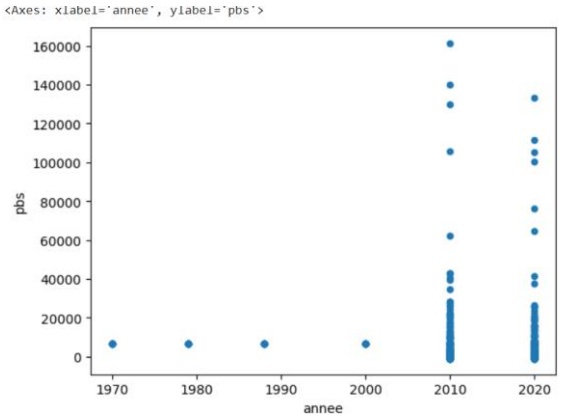


Figure 10

Enfin, cette section montre que l'équivalent temps plein (ETP), l'unité de référence utilisée pour calculer les besoins nutritionnels ou alimentaires de chaque type d'animal d'élevage (UGB), la superficie agricole utilisée en hectares, les valeurs et les années expliquent le mieux la production agricole standard (PBS) au Pays Basque entre 1970 et 2020.

B. Analyse des variables de types catégoriels

La production agricole standard au Pays Basque dépend également de facteurs non quantitatifs tels que le type d'exploitation, l'échelle, la catégorie et la commune. Une première observation de la production selon la catégorie, figure 11, semble nous donner peu d'informations, nous avons donc poussé l'analyse plus loin en observant la surface agricole utilisée selon l'échelle entre 1970 et 2020, figure 12, nous montre que de 1970 à 2020, le pôle Bidache a presque toujours eu une surface agricole utilisée plus importante que les autres pôles, avec 39,74 ha, suivi par les pôles Amikuze (38,65 ha) et Soule Xiberoa (38,44 ha).

Par ailleurs, la figure 13 montre que le pôle Garazi Biagorri a eu la production la plus élevée de l'histoire du Pays Basque en 2010, avec un potentiel de production par hectare ou par tête d'animal de plus de 40 000 unités. On observe également une croissance dans les autres pôles, qui atteindront plus de 20 000 unités de production d'ici 2020. Cependant, si l'on regroupe tous les pôles du Pays Basque, on constate qu'ils avaient une production de plus de 166 000 unités en 2010 et une production en baisse de plus de 133 000 unités en 2020.

La figure 14, qui présente la production agricole par type, montre que, dans l'ensemble, les types de production ont connu un déclin au fil des ans, la production la plus élevée étant de plus de 161 000 en 2010, contre plus de 105 000 en 2020. Toutefois, les types d'exploitation tels que les entreprises individuelles, les petites et moyennes exploitations et les élevages de chèvres et de moutons ont une valeur de production potentielle par hectare ou par tête d'animal plus élevée que les autres types. En revanche, il serait judicieux d'investir dans des élevages caprins individuels, salariés ou de petite et moyenne taille, malgré une baisse très importante de la production sur 10 ans.

Comme pour les analyses précédentes de la figure 15, nous pouvons constater que la production a diminué au fil des ans. Cependant, nous pouvons dire que des catégories telles que le statut de l'exploitation, l'orientation technico-économique de l'exploitation (Otex) et la taille de l'exploitation expliquent de manière significative la production agricole.

Si l'on examine la figure 16, il semblerait que, dans l'ensemble, les communes n'expliquent pas de manière significative la production agricole au fil du temps.

En conclusion, nous pouvons affirmer avec certitude que les variables telles que l'échelle, le type, la catégorie et la commune expliquent la production agricole standard (PBS).

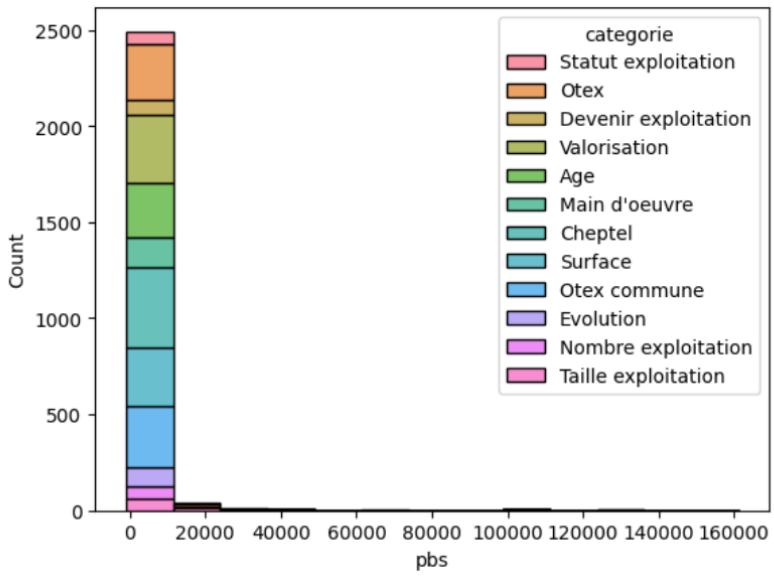


Figure 11

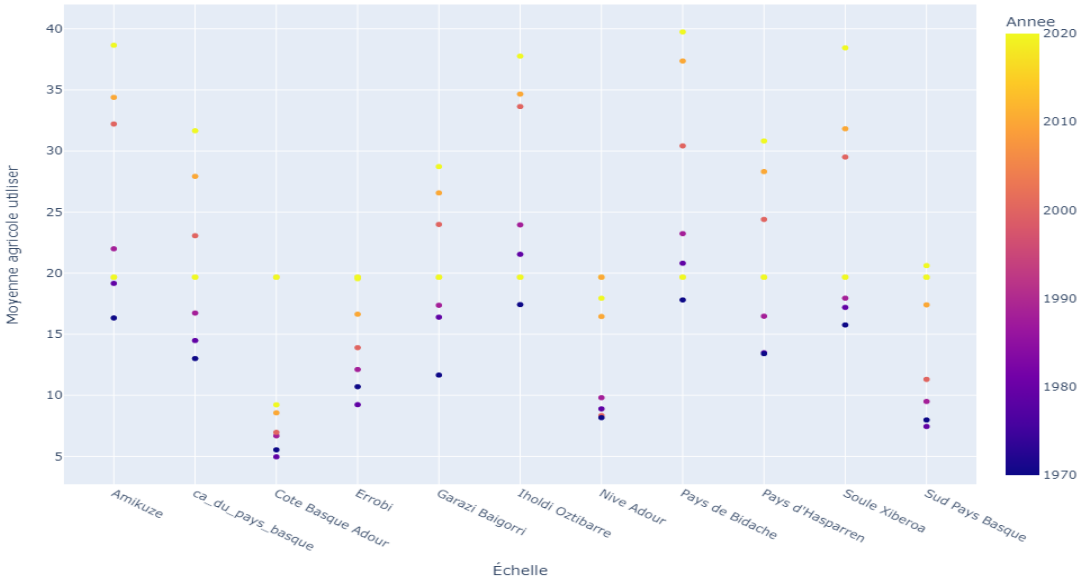


Figure 12

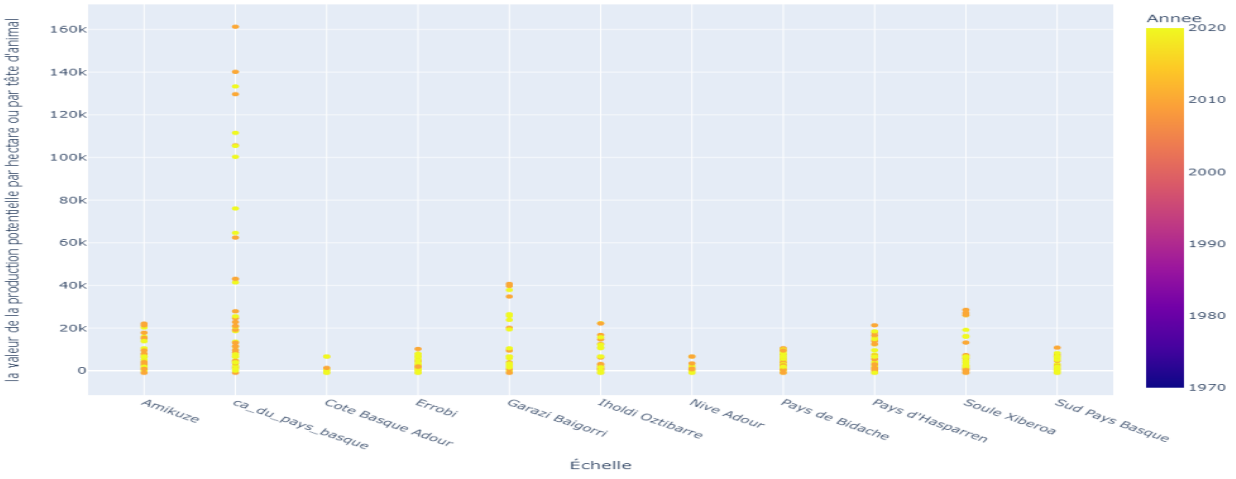


Figure 13

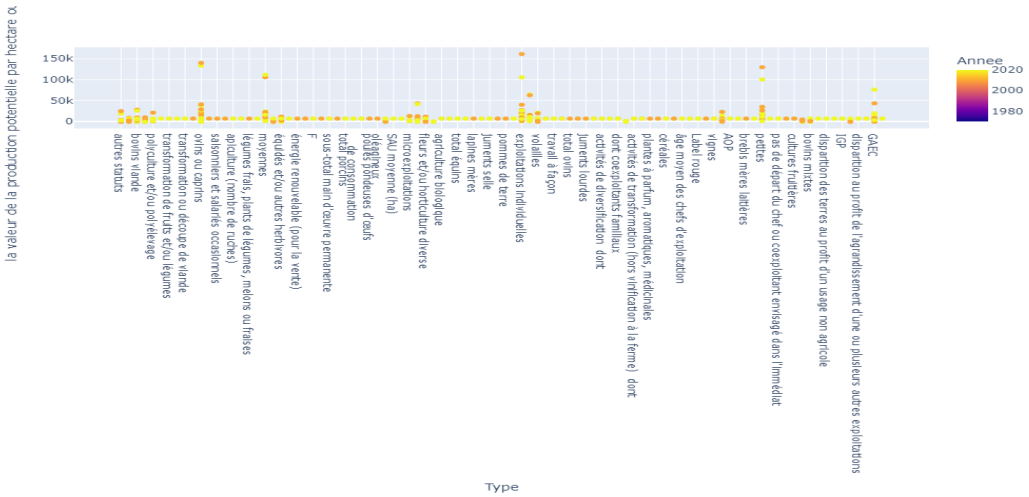


Figure 14

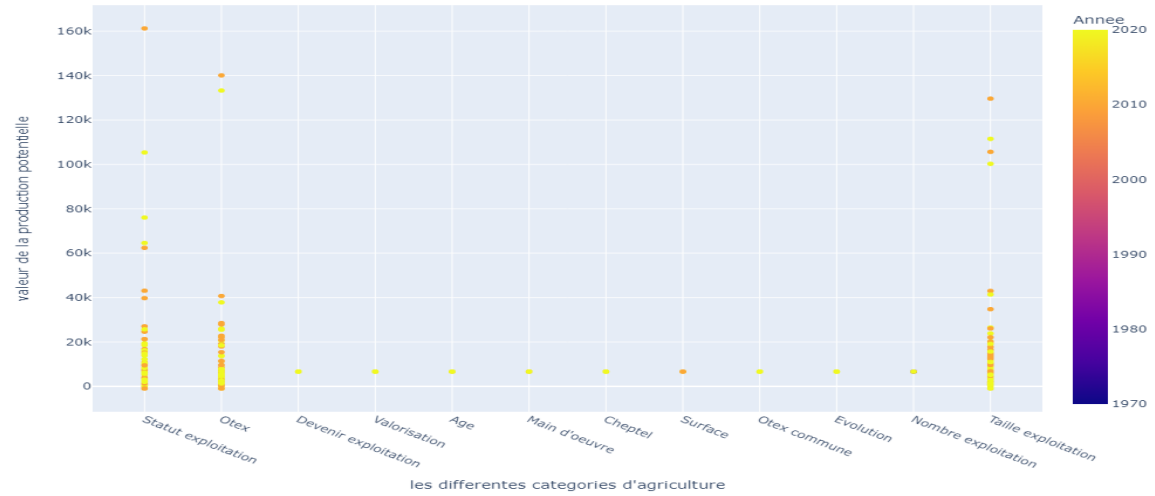


Figure 15

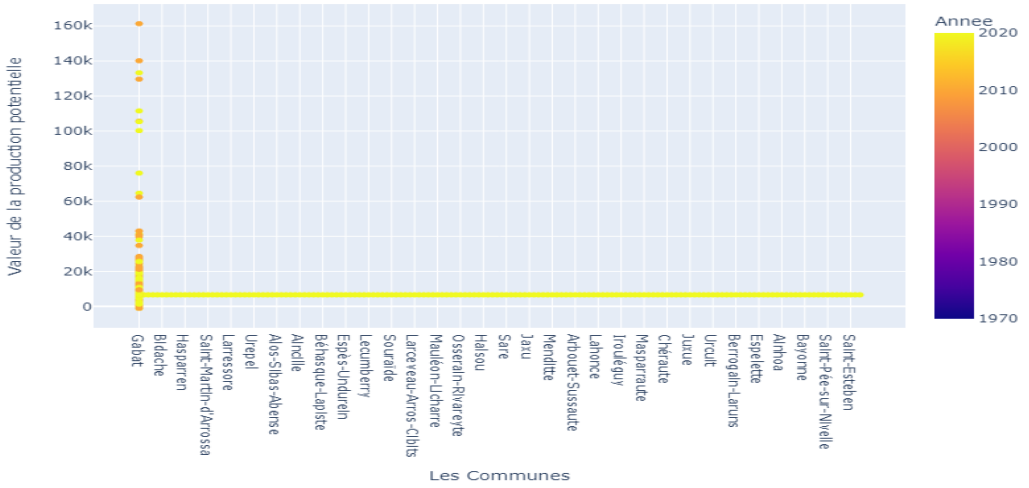


Figure 16

Nous prévoyons ensuite la production agricole en fonction de toutes les autres variables.

IV. Prédiction de la production agricole

A. Définition du pipeline

Un pipeline est une séquence d'opérations encodées par une concaténation de paires (clé, valeur). Ainsi, notre pipeline est utilisé pour prétraiter et entraîner un modèle de régression forestière utilisant des caractéristiques numériques et catégorielles (figure 17).

Voici un résumé du pipeline :

1. La première étape du pipeline est integer_transformer. Elle utilise deux étapes :
    a. imputer : Cela remplace les valeurs manquantes des caractéristiques numériques par la valeur la plus fréquente de chaque colonne.
    b. scaler : Il standardise les caractéristiques numériques en les centrant sur zéro et en les mettant à l'échelle pour avoir une variance unitaire.
2. La deuxième étape du pipeline est categorical_transformer. Elle utilise deux étapes :
    a. imputer : Cela remplace les valeurs manquantes des caractéristiques catégorielles par la valeur la plus fréquente de chaque colonne.
    b. onehot : Il transforme les caractéristiques catégorielles en variables binaires (one-hot encoding), en créant de nouvelles colonnes pour chaque catégorie unique.
3. La troisième étape du pipeline est preprocessor. Elle utilise ColumnTransformer pour appliquer les transformations spécifiques à chaque type de caractéristique :
    a. Pour les caractéristiques numériques, elle utilise integer_transformer.
    b. Pour les caractéristiques catégorielles, elle utilise categorical_transformer.
4. La quatrième étape du pipeline est base. Elle utilise deux étapes :
    a. preprocessor : Cela applique les transformations définies précédemment aux données d'entrée.
    b. regressor : Cela utilise un modèle de régression forestière pour effectuer la prédiction finale.

En résumé, ce pipeline pré-traite les caractéristiques numériques et catégorielles, remplace les valeurs manquantes, effectue un codage unique pour les caractéristiques catégorielles, normalise les caractéristiques numériques et utilise un modèle de régression forestière pour effectuer des prédictions.

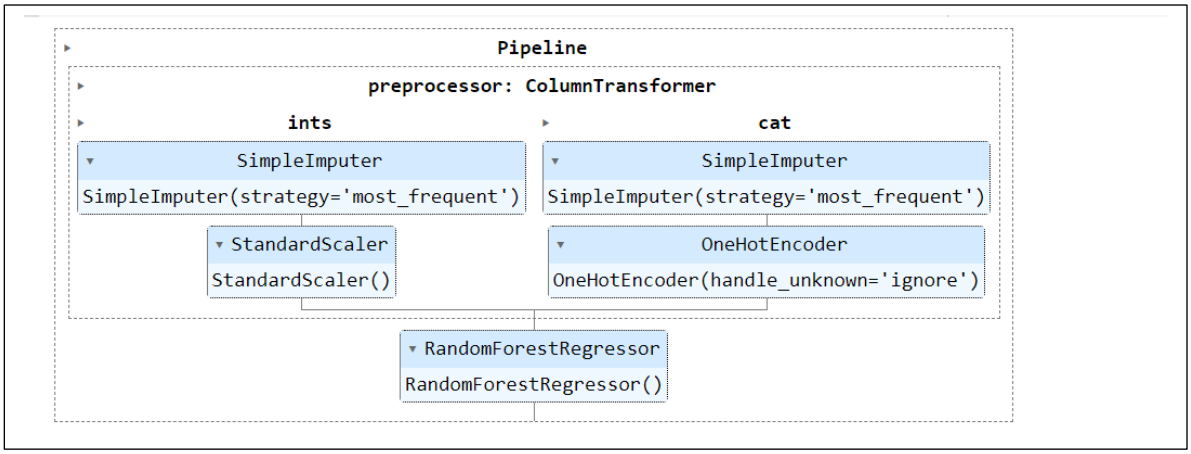


Figure 17

Nous avons ensuite sérialisé le modèle de base en le nommant modèle_rf. Cela nous a permis d'obtenir un coefficient de détermination de 89,17 %. En d'autres termes, cela signifie que le modèle de régression forestière utilisé pour faire des prédictions a une bonne capacité à expliquer la variation de la variable dépendante, puisqu'environ 89,17% de la variance de la variable dépendante est expliquée par le modèle.

Nous avons également examiné les coefficients de détermination des modèles de régression linéaire et de Ridge (Fig. 18), obtenant respectivement 54,2 % pour le modèle linéaire et 57,1 % pour le modèle de Ridge.

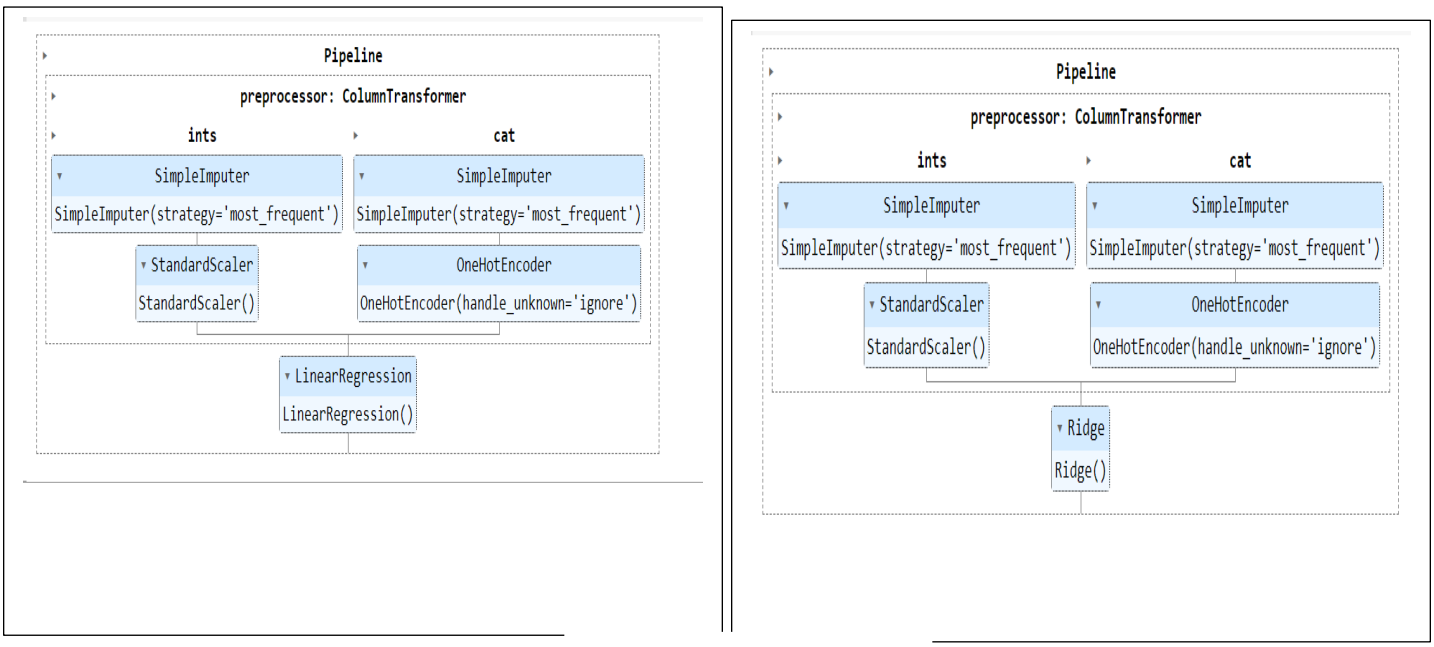


Figure 18

B. Evaluation de la prédiction

La figure 19 nous montre bien que les points qui ont été prédits sont très comparables aux données de test.

s

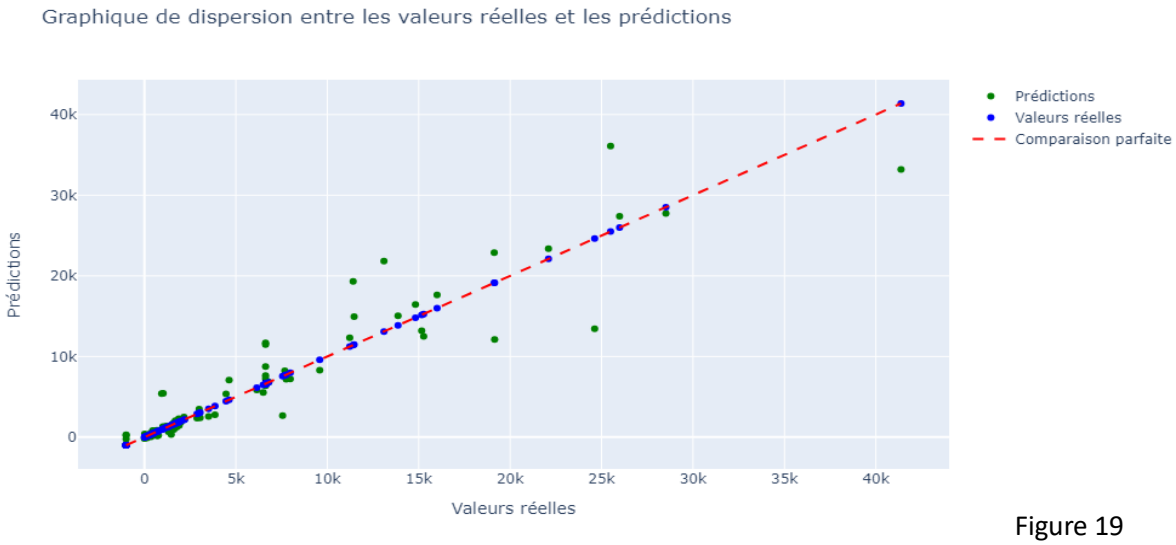


Figure 19

Utilisation de l’api FAST

Après avoir sérialisé notre modèle, nous avons créé une API nommée rgaApi.py qui importe FastAPI et Pydantic pour créer une route vers les applications. Ensuite, nous avons utilisé uvicorn pour démarrer le serveur Gunicorn qui gère les requêtes entrantes. Ensuite, nous avons créé un endpoint approprié dans l'application pour recevoir des requêtes et faire des prédictions en utilisant le modèle sérialisé. Enfin, nous avons utilisé Postman pour tester différentes entrées et nous assurer que notre API fonctionnait correctement en fournissant des réponses exactes (Figure 20).

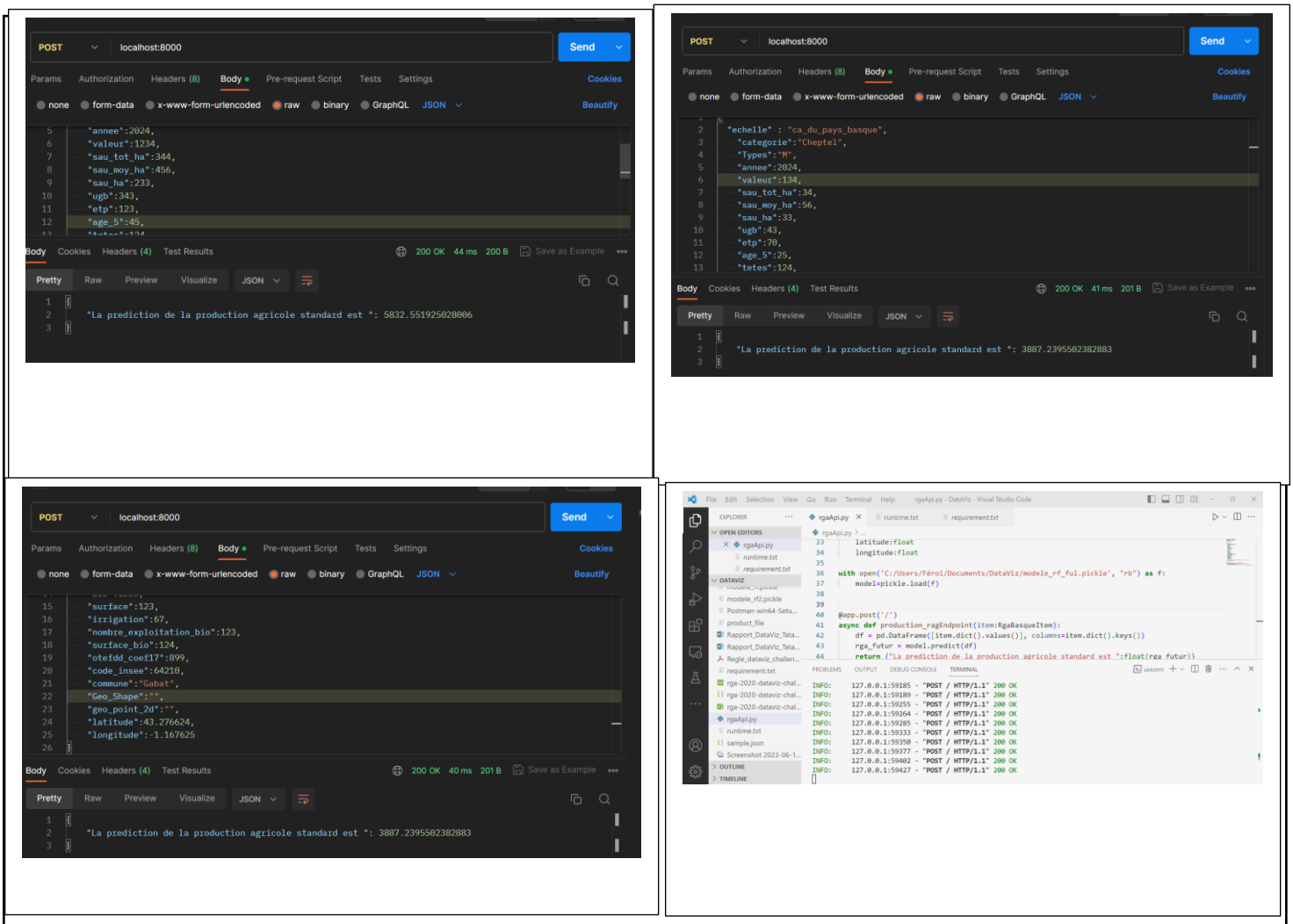


Figure 20

Conclusion

En définitive, l'objectif de cette présentation était de comprendre les facteurs qui peuvent améliorer la production agricole potentielle par hectare ou par tête de bétail, sans aucune aide (PBS), en se basant sur les données fournies par l'open data agricole ZABAL. Par conséquent, nous avons d'abord nettoyé les données au début de l'étude, puis effectué une analyse exploratoire, fait une prédiction et enfin évalué les résultats. Ainsi, nous pouvons affirmer que la production agricole par hectare ou par tête de bétail au Pays Basque peut être expliquée en fonction des variables incluses dans notre étude, et en particulier qu'elle est mieux expliquée par la superficie agricole utilisée, le temps de travail à temps plein, l'unité de bétail, 1e nombre d'exploitations et le temps. De plus, nous avons fait une prédiction et nous avons réalisé que dans les années suivantes, la production agricole serait bonne à 89,17%. Enfin, nous avons créé une api.

Licence et version

- Python == 3.10.11 (Open source)
- Postaman version 2012 (version Open Source)
- pandas version: 1.5.3 (Open source)
- numpy version: 1.22.4 (Open source)
- matplotlib version: 3.7.1 (Open source)
- seaborn version: 0.12.2 (Open source)
- plotly version: 5.13.1 (Open source)
- mpl_toolkits.basemap version: 1.3.7 (Open source)
- folium version: 0.14.0 (Open source)
- pickle version: 4.0 (Open source)
- scikit-learn version: 1.2.2 (Open source)

- anyio==3.7.0
- click==8.1.3
- colorama==0.4.6
- exceptiongroup==1.1.1
- fastapi==0.97.0
- gunicorn==20.1.0
- h11==0.14.0
- idna==3.4
- joblib==1.2.0
- numpy==1.25.0
- pandas==2.0.2
- pydantic==1.10.9
- python-dateutil==2.8.2
- pytz==2023.3
- scikit-learn==1.2.2
- scipy==1.10.1
- six==1.16.0
- sniffio==1.3.0
- starlette==0.27.0
- threadpoolctl==3.1.0
- typing_extensions==4.6.3
- tzdata==2023.3
- uvicorn==0.22.0

Utilisation de logiciels

Voici les rôles de chacun des logiciels utilisés :

a. import pandas as pd: Importe la bibliothèque Pandas qui est utilisée pour la manipulation et l'analyse des données.

b. import numpy as np: Importe la bibliothèque NumPy qui est utilisée pour effectuer des calculs numériques efficaces et pour manipuler des tableaux multidimensionnels.

c. import matplotlib.pyplot as plt: Importe la bibliothèque Matplotlib, qui est utilisée pour créer des graphiques et des visualisations en utilisant une syntaxe similaire à celle de MATLAB.

d. import seaborn as sns: Importe la bibliothèque Seaborn, qui est utilisée pour créer des graphiques statistiques attrayants et informatifs.

e. import plotly.express as px: Importe la bibliothèque Plotly Express, qui facilite la création de graphiques interactifs tels que des graphiques à dispersion, des histogrammes, des boîtes à moustaches, etc.

f. import plotly.graph_objects as go: Importe la bibliothèque Plotly Graph Objects, qui offre plus de contrôle sur la création de graphiques interactifs en utilisant une syntaxe orientée objet.

g. from mpl_toolkits.basemap import Basemap: Importe la classe Basemap de la bibliothèque mpl_toolkits.basemap, qui est utilisée pour créer des cartes et des visualisations géographiques.

h. import folium: Importe la bibliothèque Folium, qui est utilisée pour créer des cartes interactives et des visualisations géospatiales.

i. from sklearn.model_selection import train_test_split: Importe la fonction train_test_split de la bibliothèque scikit-learn (sklearn), qui est utilisée pour diviser les données en ensembles d'entraînement et de test.

j. from sklearn.compose import ColumnTransformer: Importe la classe ColumnTransformer de scikit-learn, qui est utilisée pour appliquer différentes transformations sur les colonnes spécifiées d'un tableau de données.

k. from sklearn.pipeline import Pipeline: Importe la classe Pipeline de scikit-learn, qui est utilisée pour définir des pipelines d'apprentissage automatique, qui sont des séquences ordonnées d'estimateurs (transformations et modèles) à appliquer sur les données.

l. from sklearn.impute import SimpleImputer: Importe la classe SimpleImputer de scikit-learn, qui est utilisée pour remplacer les valeurs manquantes dans un tableau de données par des valeurs spécifiées.

m. from sklearn.preprocessing import OneHotEncoder, StandardScaler: Importe les classes OneHotEncoder et StandardScaler de scikit-learn, qui sont utilisées respectivement pour encoder les variables catégorielles et standardiser les variables numériques.

n. from sklearn.ensemble import RandomForestRegressor: Importe la classe RandomForestRegressor de scikit-learn, qui est utilisée pour entraîner des modèles de régression à base d'arbres de décision aléatoires.

o. from sklearn.linear_model import LinearRegression, Ridge, Lasso, BayesianRidge: Importe les classes LinearRegression, Ridge, Lasso et BayesianRidge de scikit-learn, qui sont utilisées pour entraîner des modèles de régression linéaire avec différentes régularisations.

p. from sklearn.preprocessing import PolynomialFeatures: Importe la classe PolynomialFeatures de scikit-learn