Segunda entrega proyecto final

Carrera de Data Scientist – Mayo de 2023

Autor: Emilse Bover

Contenido

Abstract 2
Objetivo 2
Contexto comercial 2
Contexto analítico 2
Data acquisition 3
Data wrangling 3
Base de datos productivos 3
Base de datos climáticos de Concepción del Uruguay, Entre Ríos 4
Transformaciones de las bases de datos para el análisis 6
Creación de subset 6
Información tipos de datos 8
Análisis exploratorio de datos 9
Gráfico 1 9
Gráfico 2 10
Gráfico 3 11
Gráfico 4 12
Gráfico 5 13
Gráfico 6 13
Gráfico 7 14
Gráfico 8 14
Gráfico 9 15
Gráfico 10 15
Correlaciones 16
Gráfico 11 16
Machine learning aproximación 17
Modelo SVR 17

Abstract

Objetivo

El objetivo del presente trabajo es detectar el impacto de los factores que afectan al crecimiento de los pollos parrilleros para tomar decisiones que mejoren los resultados productivos.

Contexto comercial

El retorno económico de la empresa productora de pollos parrilleros depende de la obtención de mejores resultados productivos. A mayor ganancia de peso diaria, por ejemplo, el consumo total de alimento será menor disminuyendo el costo. Otro impacto positivo del aumento de la ganancia de peso diaria es que se reduce la edad a faena de los animales ya que se alcanza el peso deseado en menor cantidad de días. Esta reducción de días permite: liberación de superficie de producción (metros cuadrados de galpón, rotación), menor propensión a sufrir enfermedades y accidentes (por ejemplo cortes de luz). También el dueño de la granja tiene menores costos de luz y gas y por lo tanto mayores ganancias finales también.

Contexto analítico

Se presentan las librerías utilizadas:

```
import pandas as pd
import xlsxwriter
import matplotlib as mpl
import matplotlib.pyplot as plt
import seaborn as sns
import scipy as sp
import numpy as np
import datetime
import pingouin as pg
import statsmodels.api as sm
from scipy import stats
import requests
import json
import plotly.express as px
import ydata_profiling
from sklearn.model_selection import train_test_split
from sklearn.svm import SVR
from sklearn.metrics import mean_squared_error, r2_score
from sklearn import preprocessing
```

```
! pip install -U scikit-learn
! pip install scikit-learn
! pip install pyppeteer
```

Data acquisition

La empresa ha provisto un archivo excel con información de resultados y se utilizaron datos climáticos históricos obtenidos a través de una API de Open Meteo.

Ingresos semanales compilado registra resultados finales de la producción en la pestaña BASE. Contiene además datos de genética, nutrición y sanidad.

```
df = pd.read_excel(r"C:\Users\ebover\OneDrive - FRIGORIFICO DE AVES SOYCHU S.A.I.C.F.I.A\ingresos semanales compilado.xlsx",sheet_name='BASE', skiprows=1 , usecols=range(1,67))
```

```
Se hizo una solicitud HTTP a la API de Open-Meteo
response = requests.get('https://archive-api.open-meteo.com/v1/archive?latitude=-32.48&longitude=-58.23&start_date=2020-01-01&end_date=2022-12-31&daily=temperature_2m_max,temperature_2m_min,precipitation_sum&timezone=America%2FSao_Paulo')
```

Data wrangling

Primero se eliminaron valores erróneos de la base de datos que se conocían antes de iniciar el análisis:

Base de datos productivos

```
df.drop(df[(df['A.D.'] >100)].index, inplace=True) Se eliminan valores erróneos
df.drop(df[(df['Edad'] >70)].index, inplace=True) Se eliminan edades fuera de estándar
```

Los resultados se agrupan por crianza (índice) de cada granja y se registran los diferentes resultados e indicadores de la producción

La base de datos contiene resultados productivos de 15.571 crianzas desde el año 2015 hasta la primera quincena del mes de abril de 2023.

```
df.isna().sum()
```

```
granja                0
Nombre                0
Primer BB             0
Cantid. BB            0
A Faena               0
                   ...
conv aj AVIAGEN       0
hepatitis         14848
consumo total         0
Pes conv 2,7          0
```

```
zona prod              0
Length: 66, dtype: int64
```

Los valores en blanco hallados pertenecen a una columna que no será utilizada en este análisis

```
df.duplicated()
```

```
0        False
1        False
2        False
3        False
4        False
         ...
15643    False
15644    False
15645    False
15646    False
15647    False
```

La base no posee valores de crianzas duplicados.

Base de datos climáticos de Concepción del Uruguay, Entre Ríos

Se convirtieron los datos JSON en un diccionario de Python

```
data = response.json()['daily']
```

Se convirtió el diccionario en un DataFrame de pandas

```
dfc = pd.DataFrame(data)
```

	time	temperature_2m_max	temperature_2m_min	precipitation_sum
0	2020-01-01	26.1	20.8	0.0
1	2020-01-02	26.2	17.9	0.0
2	2020-01-03	26.8	18.7	0.0
3	2020-01-04	27.8	18.1	0.0
4	2020-01-05	30.3	19.6	0.0

```
dfc.shape
```

```
(1096, 4)
```

```
dfc.info()
```

```
   Column              Non-Null Count  Dtype
---  ------              --------------  -----
 0   time                1096 non-null   object
 1   temperature_2m_max  1096 non-null   float64
 2   temperature_2m_min  1096 non-null   float64
 3   precipitation_sum   1096 non-null   float64
```

```
dfc.isna().sum()
```

```
time                  0
temperature_2m_max    0
temperature_2m_min    0
precipitation_sum     0
```

La base no posee valores en blanco

```
dfc.duplicated()
```

```
0       False
1       False
2       False
3       False
4       False
        ...
1091    False
1092    False
1093    False
1094    False
1095    False
```

La base no posee fechas duplicadas

Transformaciones de las bases de datos para el análisis

De la base de datos provista por la empresa se seleccionarán las siguientes columnas para el análisis:

Nombre	Descripción
granja	Unidad productiva compuesta por galpones
Primer BB	Fecha inicio crianza de aves
A Faena	Cantidad de aves al final de la crianza
% Mortan.	Mortalidad animales
P. Prom.	Peso promedio a faena
Ulto. Levan.	Fecha fin de crianza de aves
Edad	Edad al final de la crianza
A.D.	Ganancia media diaria de peso
año levante	año de fin de crianza
mes levante	mes de fin de crianza
Estación del año	Momento del año en que ocurrió el fin de la crianza
Consumo medio diario (g)	Consumo de alimento por ave
Destino	Destino de venta: mercado interno o exportación
Densidad	Cantidad de aves por metro cuadrado de galpón (de cada granja)
kg pollo/m2	Cantidad de kilos logrados por unidad de superficie de granja
Ventilación forzada	Tipo de sistema de ventilación de granja
Ambiente controlado	Subtipo de sistema de ventilación
localidad	Localidad geográfica de las granjas
% desvío consumo std	Desvío del consumo de alimento respecto del estándar
Índice	Número de crianza (único)
zona climática	Zona de producción según clima imperante
Pes conv 2,7	Indicador de eficiencia productivo
zona prod	Zona de producción de la granja

Creación de subset

```
df['Ulto. Levan.'] = pd.to_datetime(df['Ulto. Levan.']) se define columna de fecha
dfc['time'] = pd.to_datetime(dfc['time']) se define columna de fecha

subset para año

df_1=df[['A.D.','año levante']]

df_1=df_1.groupby('año levante').mean().reset_index().round(2)
df_1.drop(df_1[(df_1['año levante'] ==2023)].index, inplace=True)

subset para estación del año
```

```
df_2=df[['A.D.','Estación del año']]

df_2=df_2.groupby('Estación del año').mean().reset_index().round(2)

df_3=df[['A.D.','mes levante']]

df_3=df_3.groupby('mes levante').mean().reset_index().round(2)
```

subset para sistema de ventilación

```
dfv=df[['Índice','granja','Ulto. Levan.','A Faena','% desvío consumo
std','% Mortan.','P. Prom.','Edad','kg pollo/m2','Estación del
año','Ventilación forzada','Ambiente controlado','A.D.','Pes conv
2,7','zona prod','mes levante','año levante']]
dfv.set_index('Índice')
```

Datos para gráfico 5

```
serie1 =  dfv.loc[dfv['Ventilación forzada'] == 'N' ,'A.D.']
serie2 =  dfv.loc[dfv['Ventilación forzada'] == 'S' ,'A.D.']
serie3 =  dfv.loc[dfv['Ventilación forzada'] == 'MI' ,'A.D.']
```

subset para sistema de ventilación 2

```
dfv_a= dfv.groupby('Ventilación forzada')['A.D.'].mean().reset_index()
dfv_a.set_index('Ventilación forzada')
```

Nuevo dataframe con los datos filtrados para la localidad de Concepción del Uruguay y fechas para los que se tienen valores de temperaturas

```
dfCDU_2=df[['Índice','granja','Ulto. Levan.','A Faena','% desvío
consumo std','% Mortan.','P. Prom.','Edad','kg pollo/m2','Estación del
año','Ventilación forzada','Ambiente controlado','A.D.','Pes conv
2,7','zona prod','mes levante','año levante','zona climática']]
dfCDU_3=dfCDU_2[(dfCDU_2['Ulto. Levan.']>'2019-12-31' ) &
(dfCDU_2['Ulto. Levan.']<'2023-04-01')]
```

Se unifican los dos dataframes generados: datos climáticos y datos productivos de la zona de Concepción del Uruguay en Entre Ríos

```
df_U=pd.merge(dfc, dfCDU_3, left_on='time', right_on='Ulto. Levan.')
```

subset para análisis nuevo dataframe

```
df_U2=df_U[['año levante','mes
levante','temperature_2m_max','A.D.','Ventilación forzada']]

df_U2=df_U2.groupby(['año levante','mes levante','Ventilación
forzada']).mean().reset_index()

df_U3=df_U[['año levante','mes
levante','temperature_2m_min','A.D.','Ventilación forzada']]
```

```
df_U3=df_U3.groupby(['año levante','mes levante','Ventilación
forzada']).mean().reset_index()

df_4=df[['A.D.','zona prod']]
df_4=df_4.groupby(['zona prod']).mean().reset_index().round(2)
```

subset para machine learning

```
df_5= df[['A.D.', 'año levante','mes levante','Estación del año',
'Ventilación forzada','zona prod']]

label_encoder = preprocessing.LabelEncoder()
df_5['Estación del año']= label_encoder.fit_transform(df_5['Estación
del año'])
df_5['Estación del año'].unique()


label_encoder = preprocessing.LabelEncoder()
df_5['Ventilación forzada']=
label_encoder.fit_transform(df_5['Ventilación forzada'])
df_5['Ventilación forzada'].unique()


label_encoder = preprocessing.LabelEncoder()
df_5['zona prod']= label_encoder.fit_transform(df_5['zona prod'])
df_5['zona prod'].unique()
```

Información tipos de datos

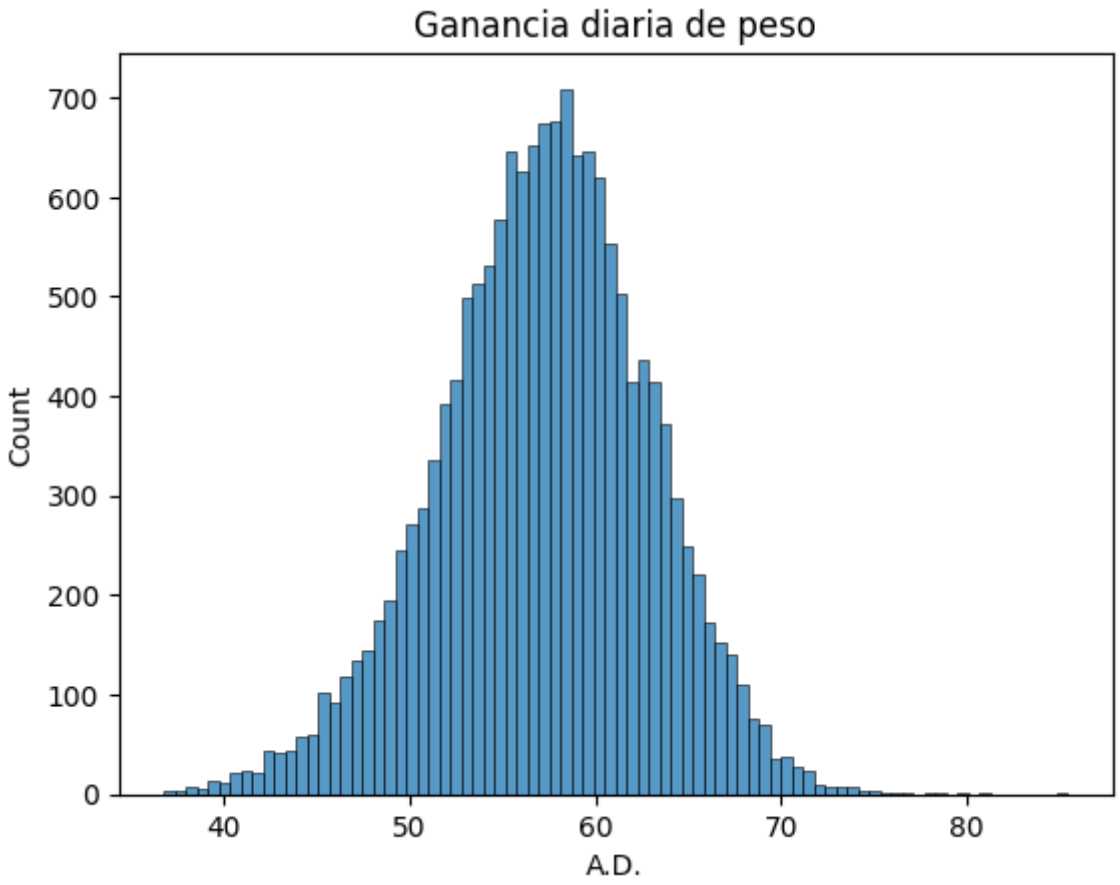
```
df2_num = df.select_dtypes(exclude=object)
df2_num.info()


Data columns (total 16 columns):
    Column                    Non-Null Count  Dtype
---  ------                    --------------  -----
 0   granja                    15645 non-null  int64
 1   Primer BB                 15645 non-null  datetime64[ns]
 2   A Faena                   15645 non-null  int64
 3   % Mortan.                 15645 non-null  float64
 4   P. Prom.                  15645 non-null  float64
 5   Ulto. Levan.              15645 non-null  datetime64[ns]
 6   Edad                      15645 non-null  float64
 7   A.D.                      15645 non-null  float64
 8   año levante               15645 non-null  int64
 9   mes levante               15645 non-null  int64
 10  Consumo medio diario (g)  15645 non-null  float64
 11  Densidad                  15645 non-null  float64
 12  kg pollo/m2               15645 non-null  float64
 13  % desvío consumo std      15645 non-null  float64
 14  Índice                    15645 non-null  int64
 15  Pes conv 2,7              15645 non-null  float64

df2_object= df.select_dtypes(include=object)
df2_object.info()

<class 'pandas.core.frame.DataFrame'>
```

```
Int64Index: 15645 entries, 0 to 15647
Data columns (total 8 columns):
    Column                Non-Null Count  Dtype
---  ------                --------------  -----
 0   Tipo de granja        15645 non-null  object
 1   Estación del año      15645 non-null  object
 2   Destino               15645 non-null  object
 3   Ventilación forzada   15645 non-null  object
 4   Ambiente controlado   15645 non-null  object
 5   localidad             15645 non-null  object
 6   zona climática        15645 non-null  object
 7   zona prod             15645 non-null  object
dtypes: object(8)
memory usage: 1.1+ MB
```

Análisis exploratorio de datos

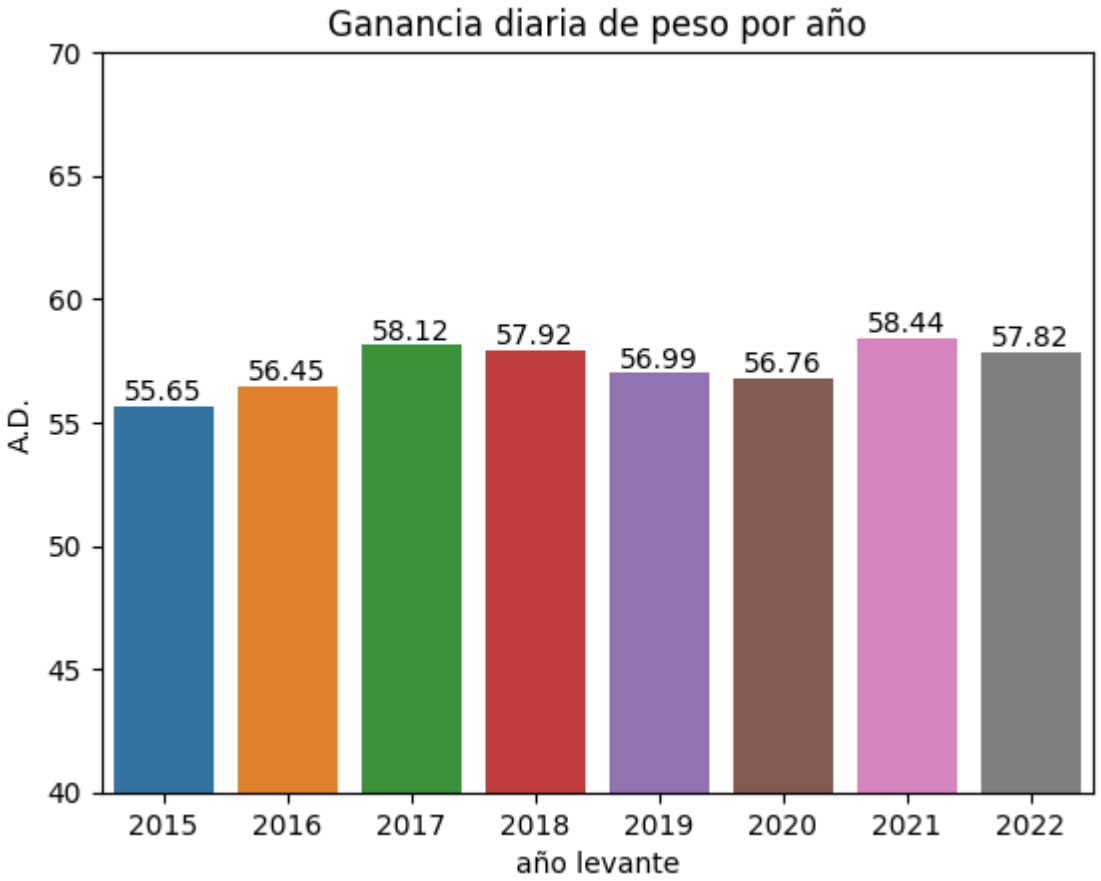
Gráfico 1

Histograma de Ganancia media diaria de peso de las 15.571 crianzas de la base



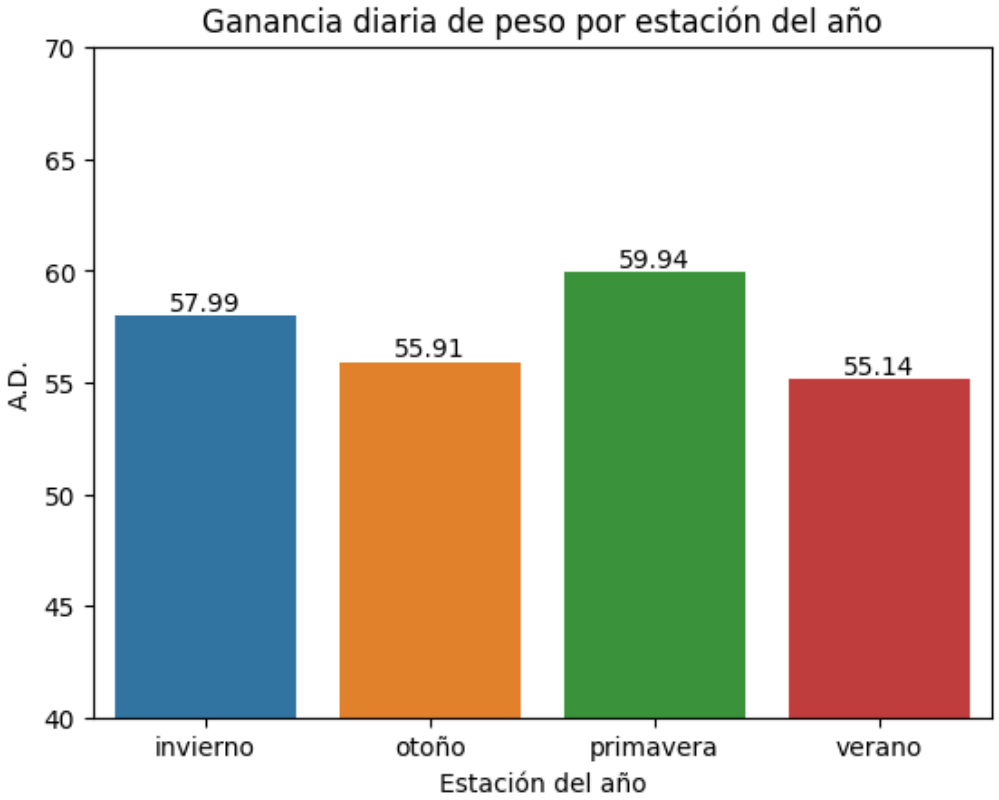
La **ganancia media diaria de peso** es una medida que permite comparar resultados entre granjas y además permite ver la evolución de los resultados globales de la empresa a lo largo del tiempo. A mayor valor, mejor será el resultado. El valor medio es de **57.18** gramos de peso ganados promedio por día.

Gráfico 2



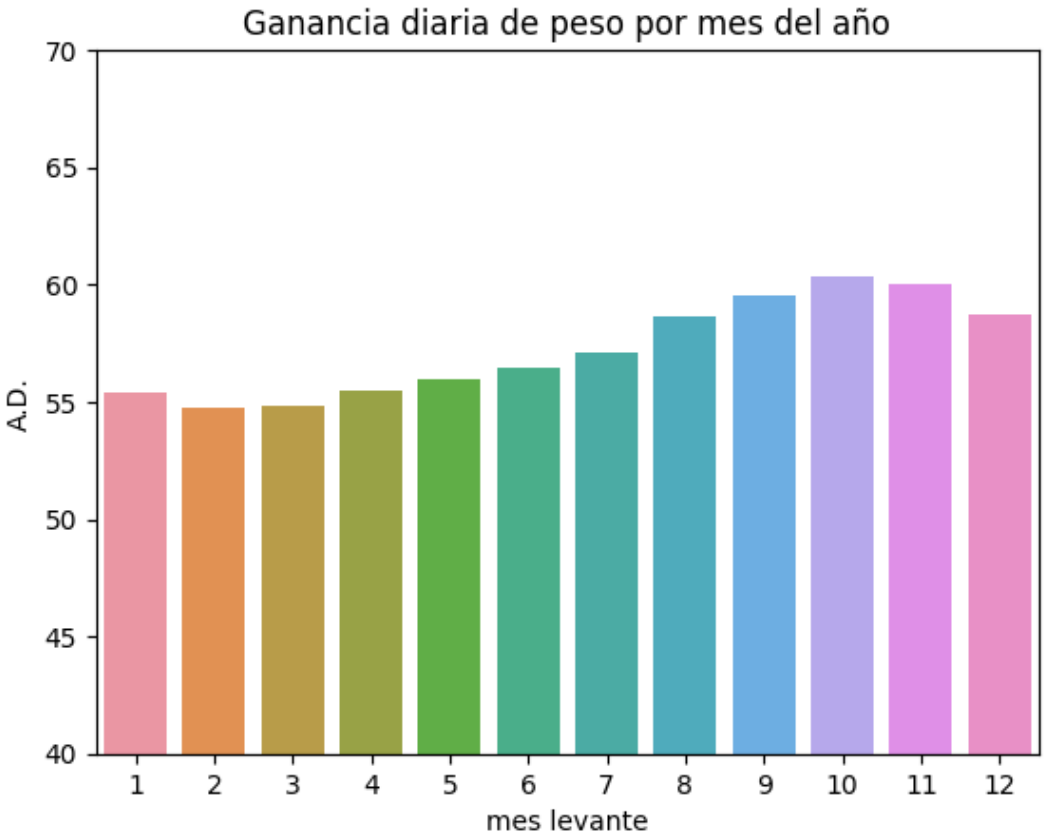
Se observa que las ganancias de peso de los años 2017 y 2021 fueron superiores al resto de los años analizados. El año 2023 fue excluído por estar incompleto.

Gráfico 3



Se observa que la ganancia de peso es variable según la estación del año en la que se desarrolle la crianza, las mayores ganancias de peso ocurren en primavera y las peores en verano dando indicios de que el ambiente puede estar afectando el resultado productivo.

Gráfico 4



Se observan también diferencias en los valores de ganancia media diaria de peso en los meses del año. Debido a ello a continuación se realizará un análisis según los tipos de ventilación que poseen las granjas.

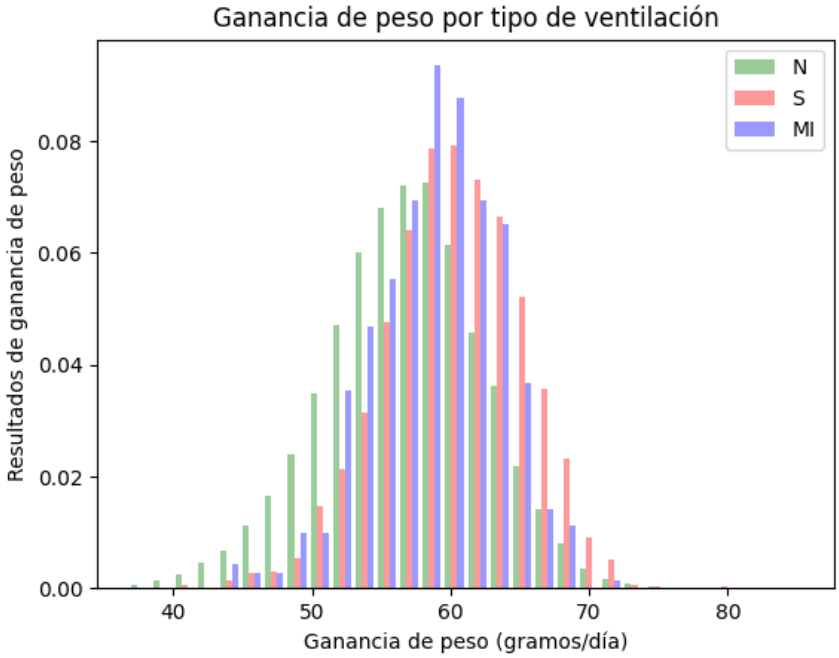
Para ello se han clasificado las granjas en tres categorías:

S: Ventilación forzada: La ventilación del galpón se realiza a través de la entrada del aire por inlets o aberturas en la zona superior de los galpones y su posterior expulsión por extractores.

N: Granjas con sistema de ventilación convencional con encendido manual de ventiladores y/o apertura y cierre de las cortinas laterales del galpón.

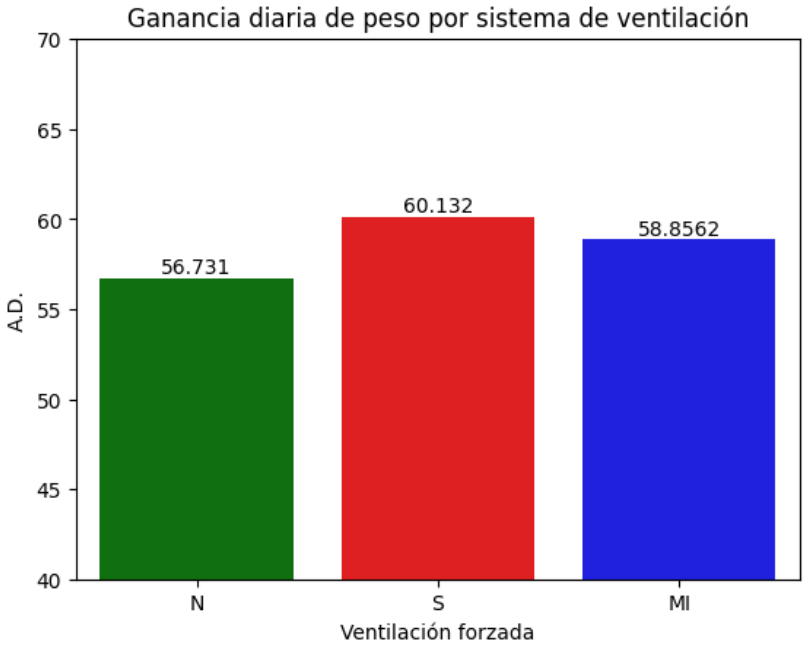
MI: Mixtas: Granjas que tienen ambos tipos de galpones.

Gráfico 5



En el gráfico se observa que los valores de ganancia de peso de los tres sistemas tienen una distribución normal y que el sistema convencional logra menores ganancias de peso que los sistemas de ventilación forzada y mixto.La distribución de los datos de las granjas convencionales se encuentra desplazada hacia la izquierda, es decir hacia valores menores de ganancia. Su promedio se ubica alrededor de los 56 gramos/días mientras que el promedio de las granjas de ventilación forzada está alrededor de los 60 gramos/día.

Gráfico 6



Habría indicios de que el sistema de ventilación tiene impacto sobre la ganancia de peso de los animales.

Gráfico 7

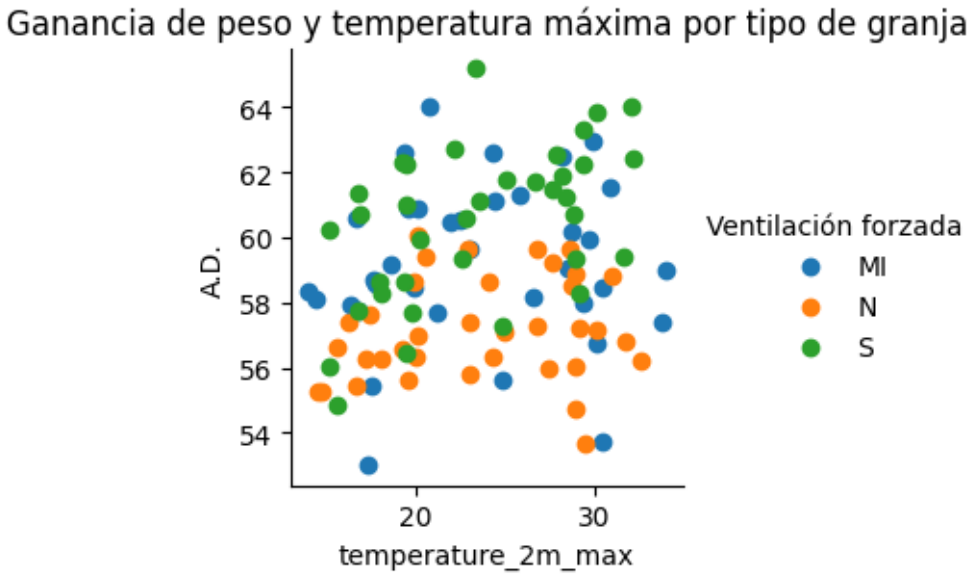
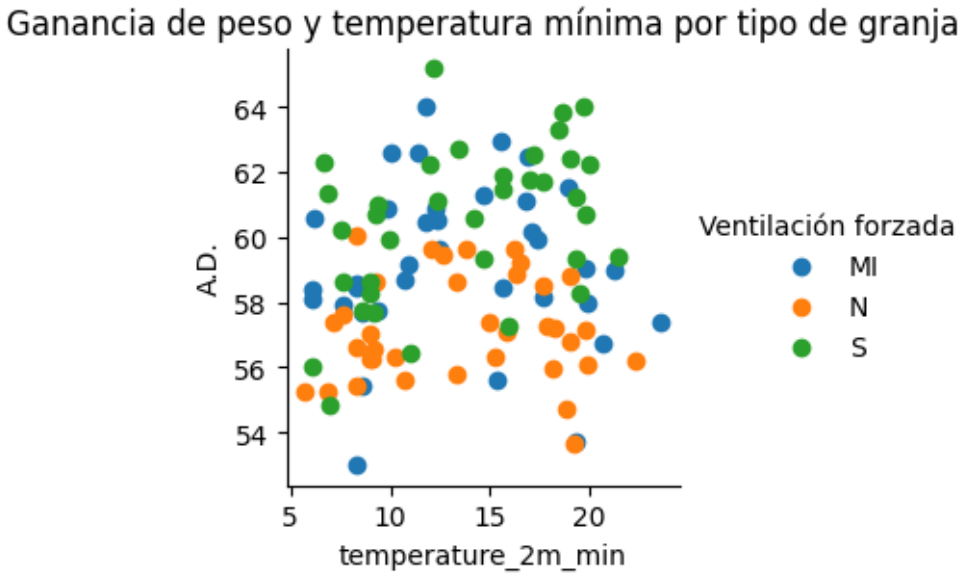
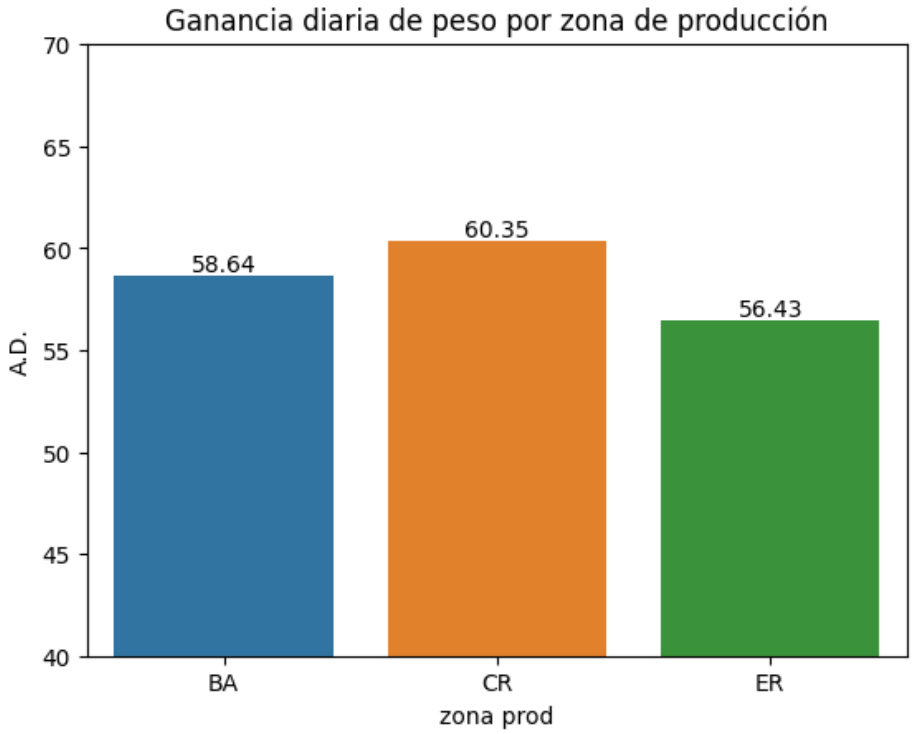


Gráfico 8



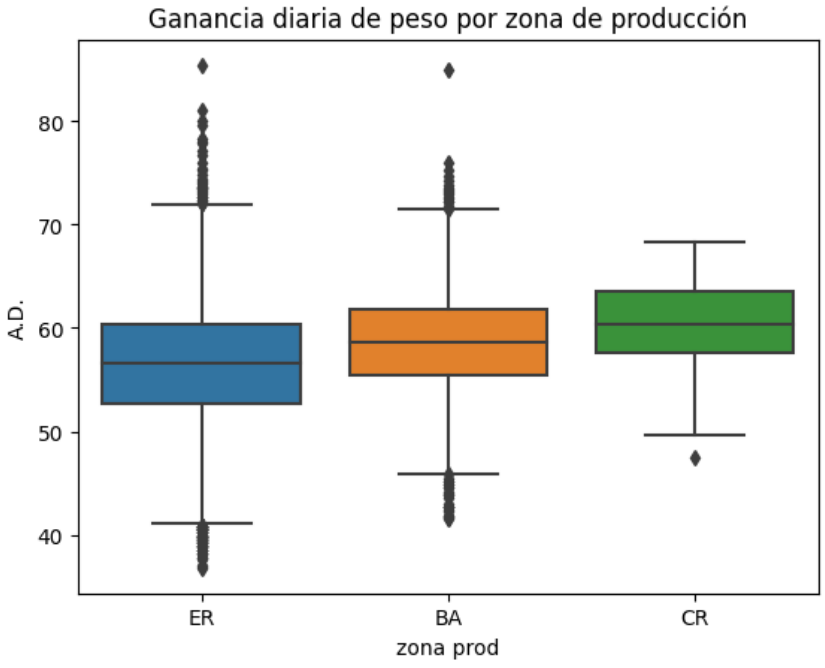
Los gráficos de correlación entre la ganancia de peso y las temperaturas máximas y mínimas diarias no muestran ninguna relación entre estas variables y tampoco con relación al sistema de ventilación.

Gráfico 9



A nivel de promedios, hay diferencias entre las zonas del país donde se encuentran ubicadas las granjas.

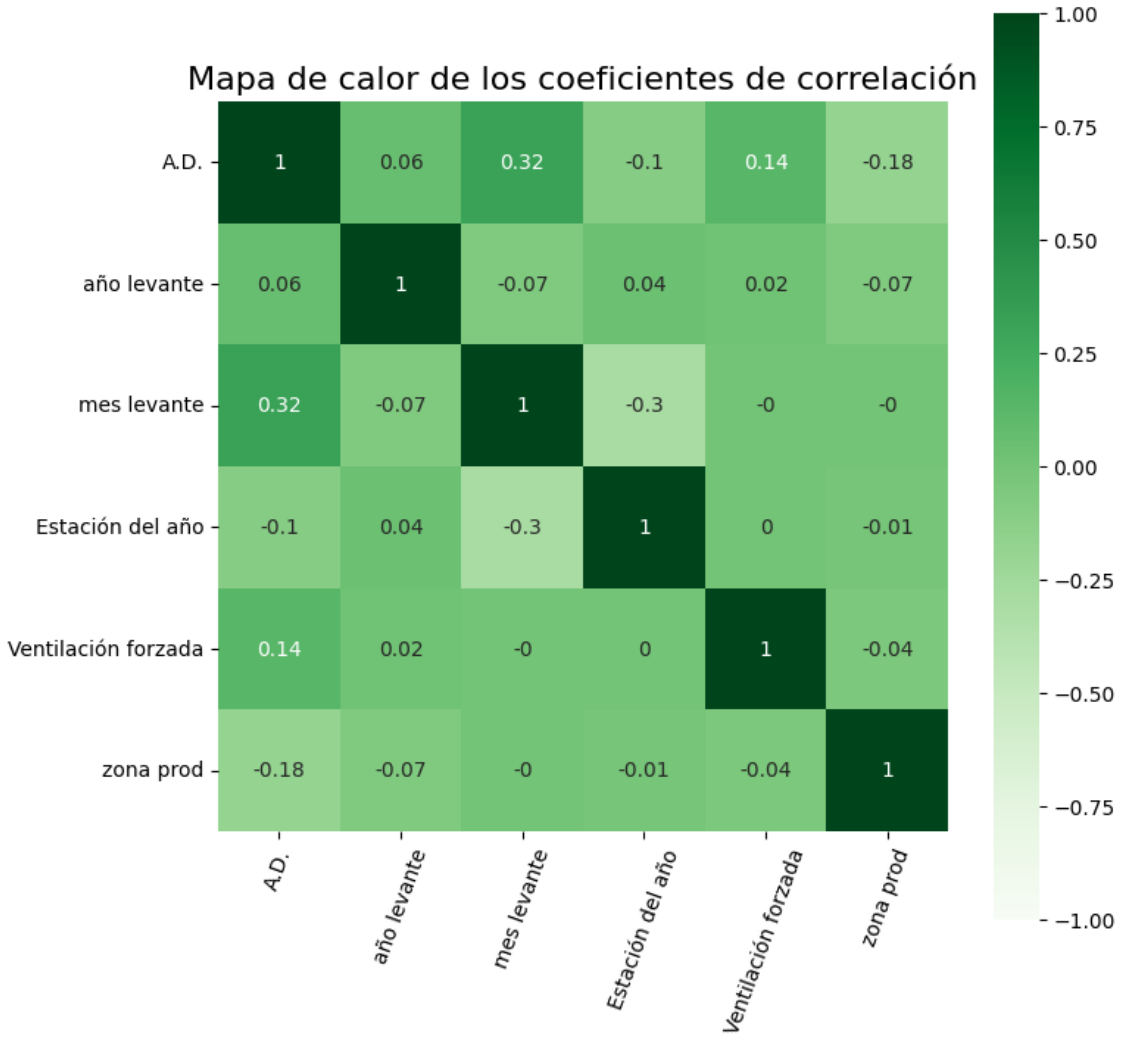
Gráfico 10



Los mejores resultados se encuentran en la zona de Crespo (CR), no solo porque tiene mayores valores de ganancia de peso sino también porque los resultados tienen menor dispersión y prácticamente no posee resultados atípicos.

Correlaciones

Gráfico 11



Del análisis del mapa de calor se desprende que la correlación entre la ganancia diaria de peso y las variables analizadas es baja en general, siendo los mayores valores un 32% para el mes de levante, un 27% de correlación positiva para la estación del año primavera y un 22% de correlación negativa para la estación verano.

Machine learning aproximación

```
 Seleccionar las variables predictoras y la variable de destino
X = df_5.drop('A.D.', 1)
y = df_5['A.D.']


 Dividir el conjunto de datos en conjunto de entrenamiento y prueba
X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.2, random_state=42)
```

Modelo SVR

```
 Crear el modelo de SVR
model = SVR(kernel='rbf', C=1e3, gamma=0.1)

model.fit(X_train, y_train)

SVR(C=1000.0, gamma=0.1)
 Hacer predicciones en el conjunto de prueba
y_pred = model.predict(X_test)

 Calcular el error cuadrático medio (MSE) y el coeficiente de
determinación (R²)
mse = mean_squared_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)


 Mostrar las métricas de evaluación de rendimiento
print("Error cuadrático medio (MSE):", mse)
print("Coeficiente de determinación (R²):", r2)
```

```
Error cuadrático medio (MSE): 24.685740309200384
Coeficiente de determinación (R²): 0.23898212288917642
```

El modelo aplicado tiene un error cuadrático medio de 24.7 con un coeficiente de determinación del 23.9%

```
raíz cuadrada del error

4.968474646126353
```

La diferencia entre los valores predichos y los valores reales del modelo es aceptable mientras que la calidad del ajuste del modelo a los datos es baja.