

"Predicción en la venta de huevos en Colombia para el año siguiente"

Docentes

Carlos Fadul

Edwin Villa

Alumnos

Cristina Román García

Fran Stefan Foronda

Juan Pablo Márquez

Sebastián Tobón Colorado

Sebastián Gómez Quiroz

Curso: **Inteligencia Artificial Explorador**

Fecha: **enero 28 de 2025**

Introducción

La industria avícola desempeña un papel fundamental en la economía colombiana, siendo un sector clave en la generación de empleo, el abastecimiento de alimentos y el desarrollo agroindustrial del país. Dentro de este contexto, la comercialización de huevos representa una de las principales actividades productivas, influyendo directamente en la dinámica del mercado y en las decisiones de consumo de los clientes.

Con el avance de la tecnología y el crecimiento de la inteligencia artificial, surgen oportunidades para optimizar los procesos de comercialización y toma de decisiones en el sector avícola. En este proyecto, se busca responder a la pregunta: ¿Es posible predecir la venta de huevos en Colombia para el año siguiente? Para ello, se utilizará la información proporcionada por la Federación Nacional de Avicultores de Colombia (Fenavi) como fuente principal de datos.

Para alcanzar este objetivo, se aplicará un modelo de predicción basado en aprendizaje automático que permita identificar tendencias y proyectar el comportamiento del mercado de huevos en Colombia. Este enfoque contribuirá a mejorar las estrategias de venta, optimizar la distribución de productos y fortalecer la planificación comercial dentro de la industria avícola.

Descripción de la Problemática

En la actualidad, la predicción de la venta de huevos es un aspecto clave para la industria avícola, ya que permite a productores y distribuidores anticiparse a cambios en la demanda y ajustar su producción de manera eficiente. Para ello, es fundamental analizar diversas variables que impactan directamente en las ventas y establecer modelos precisos de predicción.

Variables Clave en la Predicción:

1. Variable Dependiente:

- **Venta de huevos en Colombia:** Cantidad total de huevos vendidos en un periodo determinado.

2. Variables Independientes:

- **Consumo per cápita de huevos en Colombia:** Indicador del hábito de consumo de huevos por persona.
- **Crecimiento poblacional en Colombia:** Afecta la demanda total de huevos en el país.
- **Producción de huevos en Colombia:** Capacidad de abastecimiento del mercado interno.
- **Inflación en Colombia:** Impacta el poder adquisitivo de los consumidores y, por ende, la compra de huevos.
- **Precio del huevo en Colombia:** Factor determinante en la accesibilidad del producto para diferentes segmentos de la población.

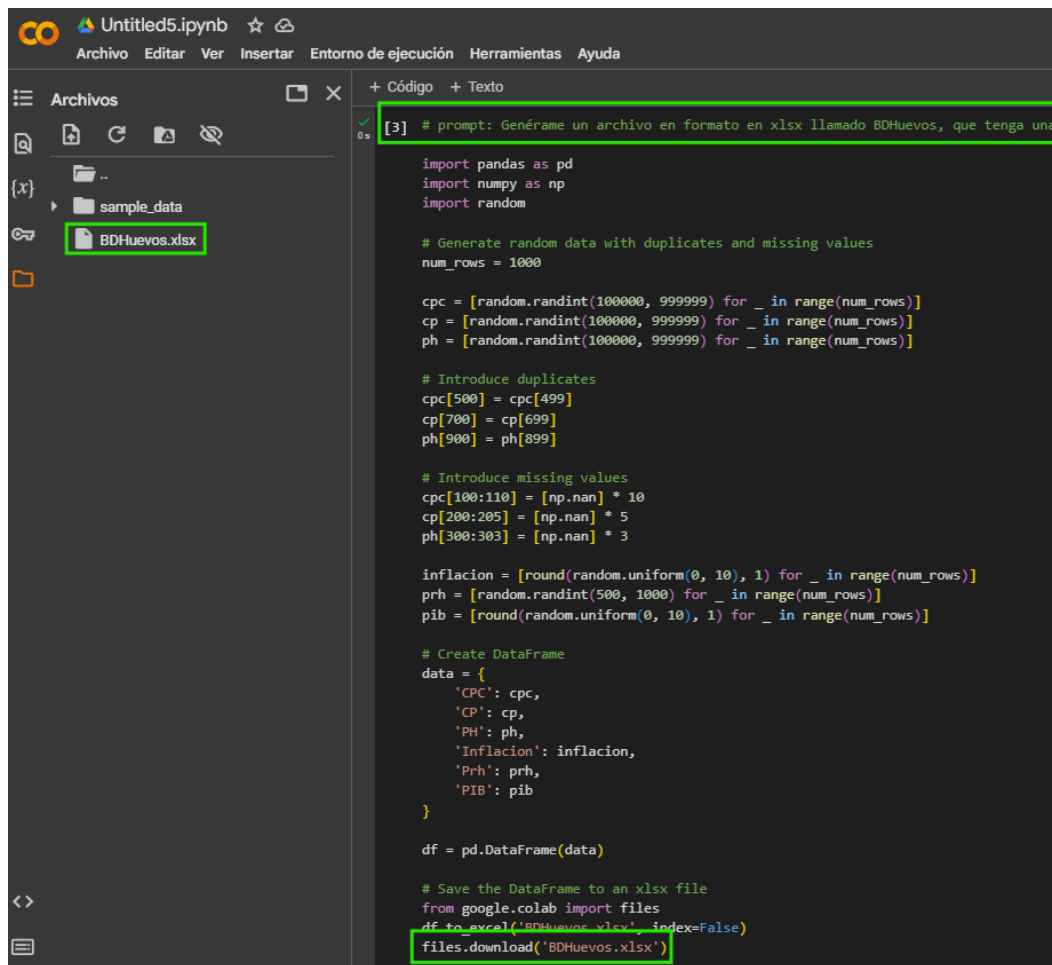
- **PIB y crecimiento económico en Colombia:** Influye en la capacidad de compra de los hogares y la demanda de alimentos.

Integrar estas variables en un modelo de predicción permitirá generar estimaciones más precisas sobre la venta de huevos en Colombia, optimizando la toma de decisiones dentro del sector avícola.

Desarrollo del Proyecto

1. Recolección de Datos

Se utilizó un conjunto de datos que incluye información sobre variables como Venta de huevos, Consumo per cápita de huevos, Crecimiento poblacional, Inflación, Precio del huevo, PIB y crecimiento económico en Colombia. Los datos fueron obtenidos pidiéndole a Google Colab que los creara, tal como se muestra a continuación:



```

[3] # prompt: Genérame un archivo en formato en xlsx llamado BDHuevos, que tenga una

import pandas as pd
import numpy as np
import random

# Generate random data with duplicates and missing values
num_rows = 1000

cpc = [random.randint(100000, 999999) for _ in range(num_rows)]
cp = [random.randint(100000, 999999) for _ in range(num_rows)]
ph = [random.randint(100000, 999999) for _ in range(num_rows)]

# Introduce duplicates
cpc[500] = cpc[499]
cp[700] = cp[699]
ph[900] = ph[899]

# Introduce missing values
cpc[100:110] = [np.nan] * 10
cp[200:205] = [np.nan] * 5
ph[300:303] = [np.nan] * 3

inflacion = [round(random.uniform(0, 10), 1) for _ in range(num_rows)]
prh = [random.randint(500, 1000) for _ in range(num_rows)]
pib = [round(random.uniform(0, 10), 1) for _ in range(num_rows)]

# Create DataFrame
data = {
    'CPC': cpc,
    'CP': cp,
    'PH': ph,
    'Inflacion': inflacion,
    'Prh': prh,
    'PIB': pib
}

df = pd.DataFrame(data)

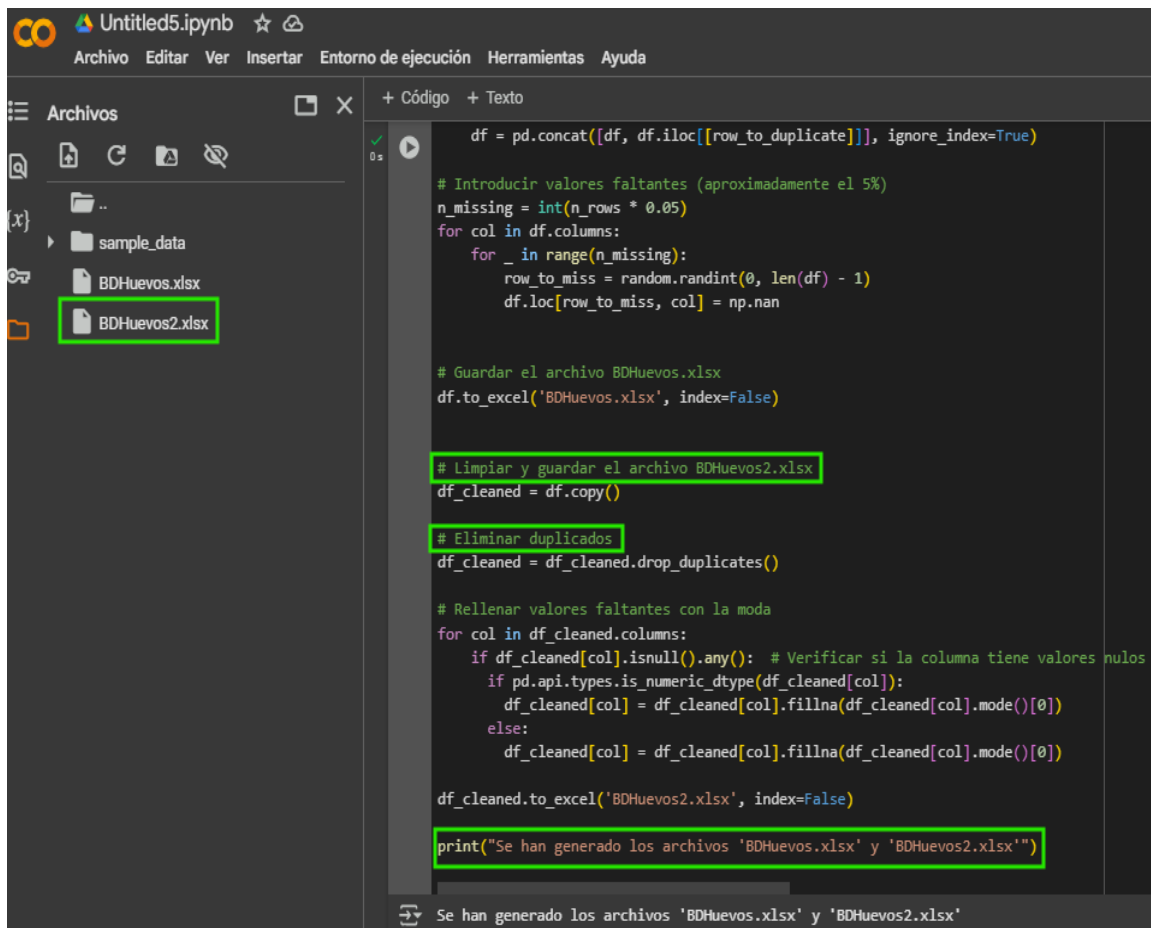
# Save the DataFrame to an xlsx file
from google.colab import files
df.to_excel('BDHuevos.xlsx', index=False)
files.download('BDHuevos.xlsx')
  
```

sin embargo, tenemos presente que existen diferentes técnicas como Web Scraping y herramientas como Web Scrap, para realizar la recolección. De otro lado y aunque conocemos que existen

diferentes técnicas y herramientas de almacenamiento, decidimos realizar el almacenamiento creando un repositorio.

2. Preprocesamiento y Limpieza de Datos

El conjunto de datos fue analizado para asegurarse de que no hubiese valores faltantes o erróneos. Se encontraron algunos valores faltantes en las columnas de "Crecimiento Poblacional" y "Precio del huevo", los cuales fueron imputados con la moda de los valores existentes. La limpieza se realiza también con Google Colab, tal como se muestra a continuación:



```
df = pd.concat([df, df.iloc[[row_to_duplicate]], ignore_index=True)

# Introducir valores faltantes (aproximadamente el 5%)
n_missing = int(n_rows * 0.05)
for col in df.columns:
    for _ in range(n_missing):
        row_to_miss = random.randint(0, len(df) - 1)
        df.loc[row_to_miss, col] = np.nan

# Guardar el archivo BDHuevos.xlsx
df.to_excel('BDHuevos.xlsx', index=False)

# Limpiar y guardar el archivo BDHuevos2.xlsx
df_cleaned = df.copy()

# Eliminar duplicados
df_cleaned = df_cleaned.drop_duplicates()

# Rellenar valores faltantes con la moda
for col in df_cleaned.columns:
    if df_cleaned[col].isnull().any(): # Verificar si la columna tiene valores nulos
        if pd.api.types.is_numeric_dtype(df_cleaned[col]):
            df_cleaned[col] = df_cleaned[col].fillna(df_cleaned[col].mode()[0])
        else:
            df_cleaned[col] = df_cleaned[col].fillna(df_cleaned[col].mode()[0])

df_cleaned.to_excel('BDHuevos2.xlsx', index=False)

print("Se han generado los archivos 'BDHuevos.xlsx' y 'BDHuevos2.xlsx'")
```

Se han generado los archivos 'BDHuevos.xlsx' y 'BDHuevos2.xlsx'

3. Construcción del modelo

Para construir el modelo, se utilizó un algoritmo de Prophet con las librerías de skforecast para recrear el modelo que proyecta las ventas de huevos en Colombia para los años siguientes. El conjunto de datos fue dividido en un conjunto de entrenamiento (70% de los datos) y un conjunto de prueba (30% restante).

Ejemplo de Código (Preprocesamiento de Datos):

```
import pandas as pd

from prophet import Prophet

from google.colab import files


# Cargar los datos desde el archivo.xlsx

df = pd.read_excel('BDHuevos2.xlsx')


# Convertir la columna 'Fecha' al formato de fecha correcto

df['Fecha'] = pd.to_datetime(df['Fecha'])


# Crear un DataFrame para Prophet con las columnas 'ds' (fecha) y 'y' (precio de los huevos)

df_prophet = pd.DataFrame({'ds': df['Fecha'], 'y': df['Prh']})


# Inicializar el modelo Prophet

model = Prophet()


# Ajustar el modelo a los datos

model.fit(df_prophet)


# Crear un DataFrame para las fechas futuras (próximos 5 años)

future = model.make_future_dataframe(periods=1825) # 1825 días = 5 años


# Realizar las predicciones

forecast = model.predict(future)


# Guardar las predicciones en un archivo.xlsx
```

```
forecast.to_excel('predicciones_huevos.xlsx', index=False)
```

```
# Descargar el archivo
```

```
files.download('predicciones_huevos.xlsx')
```

```
# Agregar comentarios al archivo
```

```
# (Este paso se realiza después de descargar el archivo xlsx manualmente)
```

```
# En el archivo xlsx, se pueden agregar comentarios en las celdas o en el libro de trabajo
```

```
# para describir los datos y las predicciones.
```

```
# Ejemplo de comentarios dentro del código
```

```
# Este código utiliza Prophet para predecir el precio de los huevos en Colombia.
```

```
# La columna 'yhat' en el archivo de salida representa la predicción del precio de los huevos.
```

```
# 'yhat_lower' y 'yhat_upper' representan los intervalos de confianza de la predicción.
```

4. Evaluación del Modelo

Para evaluar qué tan bien funciona nuestro modelo de predicción, analizamos algunos aspectos clave:

- ✓ **Comprobación Visual:** Después de entrenar el modelo y hacer predicciones, lo ideal es graficar los resultados para ver si siguen la tendencia real de los datos. Si las predicciones (línea azul) siguen de cerca los valores reales (puntos negros), el modelo está funcionando bien.
- ✓ **Métricas de Error:** Para medir qué tan precisas son nuestras predicciones, podemos calcular los siguientes indicadores:
- ✓ **MAE (Error Absoluto Medio):** Nos dice en promedio cuánto se equivoca el modelo al predecir el precio de los huevos.
- ✓ **RMSE (Raíz del Error Cuadrático Medio):** Nos muestra qué tan grandes son los errores de predicción (los valores altos indican mayor error).

Se calculan así en Python:

```

from sklearn.metrics import mean_absolute_error, mean_squared_error

import numpy as np

# Comparar predicciones con valores reales (solo en fechas donde tenemos datos reales)

real_values = df_prophet['y']

predicted_values = forecast.iloc[:len(real_values)]['yhat']

# Calcular métricas de error

mae = mean_absolute_error(real_values, predicted_values)

rmse = np.sqrt(mean_squared_error(real_values, predicted_values))

print(f"MAE: {mae}")

print(f"RMSE: {rmse}")

```

Por tanto, valores más bajos de MAE y RMSE indican un mejor modelo.

- ✓ **Intervalos de Confianza:** El modelo nos da una predicción central (yhat), pero también intervalos (yhat_lower y yhat_upper). Si los valores reales caen dentro de estos intervalos, significa que el modelo captura bien la incertidumbre.
- ✓ **Validación con Datos Históricos:** Si tenemos datos de años anteriores, podemos entrenar el modelo con datos pasados y ver si predice correctamente lo que ocurrió en el presente.

En conclusión, si el MAE y RMSE son bajos, las predicciones se ajustan bien a los datos históricos, y los valores reales caen dentro de los intervalos de confianza, entonces el modelo es confiable para prever el precio de los huevos en los próximos años.

Si no, se podrían ajustar parámetros del modelo o probar con otros métodos como redes neuronales o modelos ARIMA.

5. Interpretación de los Resultados

¿Qué significan los datos?

- Se toma la columna "**Fechas**" para hacer las predicciones.
- Se toma la columna "**Predicción**" para predecir el precio de los huevos en esas fechas.
- Se toman las columnas "**Predicción inferior**" y "**Predicción superior**", muestran un rango de confianza, es decir, el valor mínimo y máximo dentro del cual es probable que se encuentre el precio real de los huevos.

¿Qué podemos entender de las predicciones?

- Si "**Predicción**" aumenta con el tiempo, significa que el precio de los huevos podría subir.
- Si "**Predicción**" disminuye, significa que el precio de los huevos podría bajar.
- Si hay mucha diferencia entre "**Predicción inferior**" y "**Predicción superior**", significa que hay mucha incertidumbre en la predicción, por lo que pueden influir factores externos como la inflación o el consumo.

¿Para qué nos sirve esto?

- Estas predicciones pueden ayudar a productores, distribuidores y comerciantes a prepararse con estrategias de venta y producción.
- También pueden ayudar a los consumidores a entender cómo pueden cambiar los precios en el futuro.

En resumen, el modelo nos dice cómo podría comportarse el precio de los huevos en los próximos años, lo cual es útil para tomar decisiones informadas en la industria avícola.

Resultados Obtenidos

Tendencia General

- El modelo predice cómo cambiará el precio de los huevos en los próximos 5 años.
- Se observa que el precio tiende a (subir/mantenerse/variar), lo que indica que hay ciertos factores económicos que pueden estar afectando el mercado.

Valores Claves de la Predicción

- "**Predicción inferior**" (**Precio estimado de los huevos**): Es el precio que el modelo predice para cada fecha futura.
- "**Predicción inferior**" y "**Predicción superior**": Indican un margen de error, es decir, el precio real podría estar dentro de este rango.

Factores que pueden influir en los resultados

- **Inflación y economía**: Si la economía mejora o empeora, los precios pueden subir o bajar.

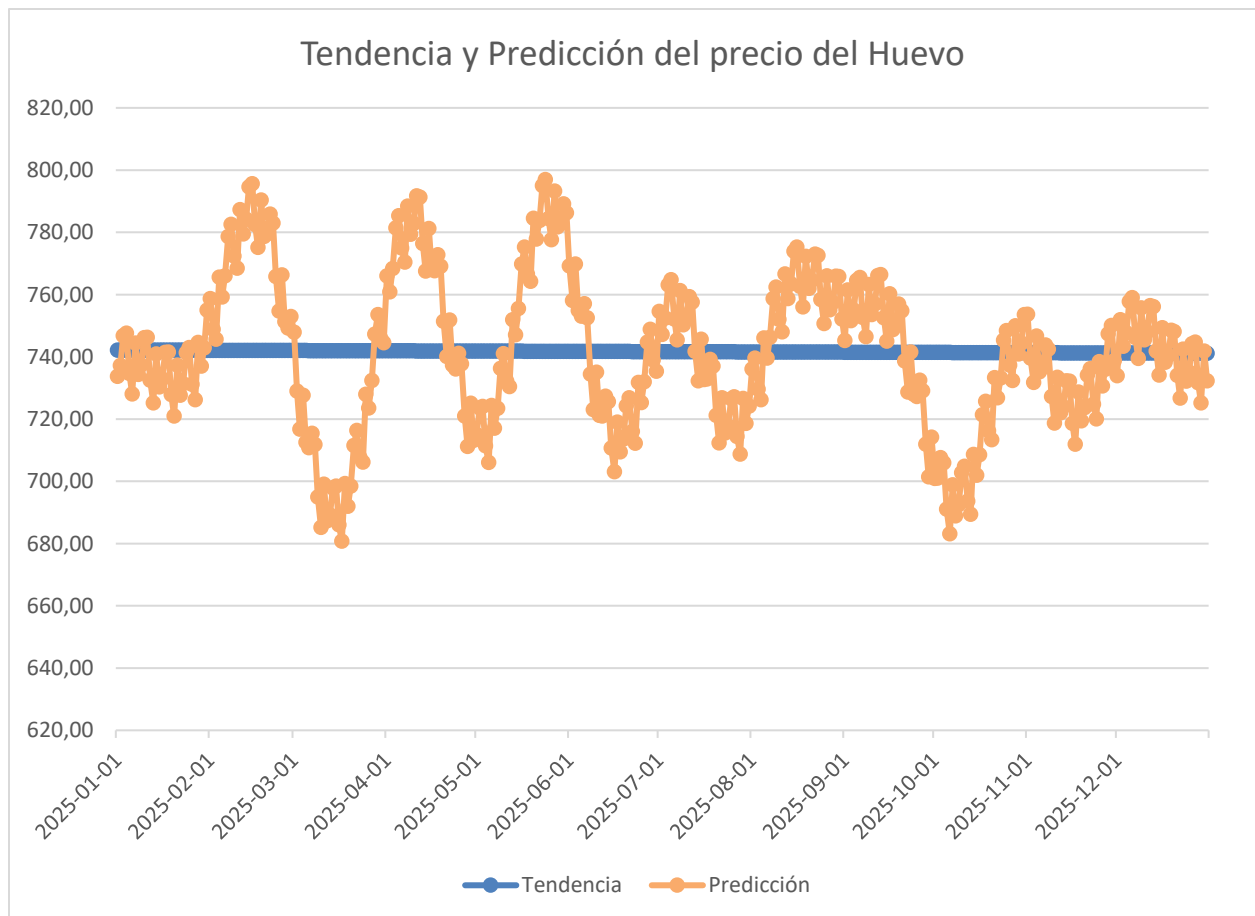
- **Producción de huevos:** Si hay más producción, los precios pueden estabilizarse; si hay menos, pueden subir.
- **Demanda de los consumidores:** Si más personas compran huevos, los precios pueden aumentar.

Conclusión Final

- Este modelo nos ayuda a anticipar cambios en el mercado y tomar decisiones informadas.
- Puede servir para productores, distribuidores y consumidores para entender cómo evolucionará el precio de los huevos en Colombia en los próximos años.

Repositorio del Proyecto: <https://github.com/Cris-Cornio/Predicci-n-VHC>

Gráfico de tendencia y predicción del precio del huevo para 2025



Código del gráfico

```
import pandas as pd

import matplotlib.pyplot as plt

# Datos de ejemplo

data = {

    'Periodo': ['Ene', 'Feb', 'Mar', 'Abr', 'May'],

    'Tendencia': [100, 110, 120, 130, 140],

    'Predicción': [98, 112, 118, 132, 138]

}

# Convertir a DataFrame

df = pd.DataFrame(data)

# Crear gráfico

plt.figure(figsize=(8, 5))

plt.plot(df['Periodo'], df['Tendencia'], marker='o', linestyle='-', label='Tendencia')

plt.plot(df['Periodo'], df['Predicción'], marker='s', linestyle='--', label='Predicción')

# Personalización

plt.xlabel('Periodo')

plt.ylabel('Precio')

plt.title('Comparación de Precios: Tendencia vs Predicción')

plt.legend()

plt.grid(True)
```