

Proyecto Inteligencia Artificial

Camilo Andres Barbosa Molina
Facultad de Ingeniería de Sistemas
Universidad Jorge Tadeo Lozano - Bogotá -Colombia
camiloa.barbosam@utadeo.edu.co

Resumen

El presente documento presentara la primera entrega del proyecto de inteligencia artificial que tiene como fin obtener información importante que sirva como base para procesos de toma de decisiones sobre un contexto específico, todo esto por medio de la experimentación con modelos de Machine learning. Para este proyecto es necesario el uso del lenguaje de programación de Python junto a herramientas de manipulación y análisis de datos como Pandas, Matplotlib u otras como Seaborn, todas estas se han visto durante el transcurso del curso y en cursos externos en [Kaggle](#)

Palabras Claves

Machine learning, Python, Pandas, Matplotlib ,Análisis de datos

I. MARCO TEÓRICO

PARA la realización del proyecto se hará uso de un [dataset](#) que proporciona la página de [Kaggle](#), el cual permite reconocer diferentes estadísticas sobre la venta de aguacates dentro del país Estadounidense, mediante el uso de datos que ofrece el sitio web [Hass Avocado Board](#) . Este análisis se realizó por medio de datos de minoristas sobre sus ventas, los cuales se extrajeron semanalmente. Este proyecto se guiará de acuerdo a los datos proporcionados, por ende, se selecciono el siguiente problema o caso de estudio principal:

- Qué región del país Estadounidense en los próximos años será más sencillo y quizás más rentable conseguir este tipo de alimento y proporcionar así a las personas interesadas en ello la respectiva información para su uso, todo esto guiado por medio de datos existente entre los años 2015 y 2018, que proporciona el propio dataset

II. ENTENDIENDO LOS DATOS

II-A. *Comprendiendo y Recolectando los Datos*

Como se mencionó anteriormente se hará uso de un Dataset , el cual en su mayoría ya tiene datos pre-procesados los cuales no necesitan de mucha modificación a nivel de la data que lo conforma. Aunque al realizar la revisión de datos con un Notebook de Jupyter se pudo evidenciar que se hizo necesario cambiar unos nombres de columnas que no permitían el uso correctamente de la data.

De las 13 columnas que el ofrece el Dataset se piensan usar la mayoría de ellas , de acuerdo al modelo de Machine learning que se escoja en la entrega siguiente, aunque se puede tener el supuesto de que 4 columnas relacionadas a la cantidad de bolsas vendidas de este alimento serán excluidas. Se debe tener en cuenta que quizás algunas de las columnas escogidas en este punto pueden ser excluidas con el desarrollo del proyecto ya que quizás no aporten de una manera positiva a nivel de profundidad o de lo que se quiere del proyecto, aunque se tiene el supuesto de que la mayoría serán de utilidad.

Además de ello se puede presenciar que estos datos provienen de un sitio web confiable y que hoy en día sigue actualizando la página, por ende, se puede considerar que son confiables y que permitirán el desarrollo adecuado del proyecto. Otra cosa importante de resaltar es que en dentro del Notebook ya se realizaron unas operaciones básicas sobre este Dataframe, generando resultados coherentes que se pueden estudiar a más profundidad con otros comandos que no se presentan dentro del Notebook.

II-B. *Descripción de los Datos*

Para poder continuar se hace necesario reconocer las características de las diferentes columnas y filas que se tiene por ende en necesario realizar operaciones básicas con Pandas para este reconocimiento.

	Date	AveragePrice	TotalVolume	PLU4046	PLU4225	PLU4770	TotalBags	SmallBags	LargeBags	XLargeBags	type	year	region
0	2015-12-27	1.33	64236.62	1036.74	54454.85	48.16	8696.87	8603.62	93.25	0.0	conventional	2015	Albany
1	2015-12-20	1.35	54876.98	674.28	44638.81	58.33	9505.56	9408.07	97.49	0.0	conventional	2015	Albany
2	2015-12-13	0.93	118220.22	794.70	109149.67	130.50	8145.35	8042.21	103.14	0.0	conventional	2015	Albany
3	2015-12-06	1.08	78992.15	1132.00	71976.41	72.58	5811.16	5677.40	133.76	0.0	conventional	2015	Albany
4	2015-11-29	1.28	51039.60	941.48	43838.39	75.78	6183.95	5986.26	197.69	0.0	conventional	2015	Albany

Figura 1: Ejemplo de DataFrame cargado en Jupyter

Para obtener este resultado se hace necesario el uso comandos como :

- `aguacates_data = pd.read_csv('sample_data/avocado.csv', index_col=0)`
- `aguacate_data.head()`

Como se puede presenciar en la Figura 1 solo se muestra las primeras 5 filas por el comando `.head()` pero antes de eso deberíamos tener una visión a nivel de cantidad de datos más acertada para ello se puede usar:

```
#Averiguar la cantidad de datos de DataFrame
aguacates_data.shape

(18249, 13)
```

Figura 2: Comando `.shape` Pandas

Ya con esta información podemos continuar describiendo las columnas que se escogieron en el apartado de comprendiendo y recolectando los datos de nuestro DataFrame, teniendo en cuenta el resultado de la Figura 1: [1]

- Date: la fecha de la observación
- AveragePrice: el precio medio de un solo aguacate
- type: convencional u orgánico
- year: el año
- region: la ciudad o región de la observación
- TotalVolume: número total de aguacates vendidos
- 4046: Número total de aguacates con PLU 4046 vendidos
- 4225: Número total de aguacates con PLU 4225 vendidos
- 4770: Número total de aguacates con PLU 4770 vendidos

Ya el ultimo paso para el reconocimiento de los datos será saber de que tipo son y para ello se hace uso del siguiente comando:

```
#Revisar el tipo de datos de todas las columnas y si es necesario cambiarla de tipo
aguacates_data.dtypes

#Como convertir la columna date a tipo fecha
#aguacates_data.Date = pd.to_datetime(aguacates_data['Date'])
#aguacates_data.dtypes

Date          datetime64[ns]
AveragePrice   float64
TotalVolume    float64
PLU4046        float64
PLU4225        float64
PLU4770        float64
TotalBags      float64
SmallBags      float64
LargeBags      float64
XLargeBags     float64
type           object
year           int64
region         object
dtype: object
```

Figura 3: Comando .dtypes Pandas

En esta anterior imagen también se ve como dentro de los comentarios se encuentra el comando `pd.to_datetime`, el cual nos permite cambiar el tipo de la Columna Date a fecha

II-C. Verificación de la calidad de los datos

En este apartado es importante realizar operaciones que me permitan ver que datos del Dataframe presentan situaciones como:

1. Datos faltantes
2. Errores de datos
3. Errores de medición
4. Inconsistencias de codificación

Para la primera situación podemos hacer uso del siguiente comando para poder mirar si dentro de los datos existen valores Nan o Null:

```
#Revisar si existen valores nulos
#De manera Global
aguacates_data.isnull()
|
```

	Date	AveragePrice	TotalVolume	PLU4046	PLU4225	PLU4770	TotalBags	SmallBags	LargeBags	XLargeBags	type	year	region
0	False	False	False	False	False	False	False	False	False	False	False	False	False
1	False	False	False	False	False	False	False	False	False	False	False	False	False
2	False	False	False	False	False	False	False	False	False	False	False	False	False
3	False	False	False	False	False	False	False	False	False	False	False	False	False
4	False	False	False	False	False	False	False	False	False	False	False	False	False
...
7	False	False	False	False	False	False	False	False	False	False	False	False	False
8	False	False	False	False	False	False	False	False	False	False	False	False	False
9	False	False	False	False	False	False	False	False	False	False	False	False	False
10	False	False	False	False	False	False	False	False	False	False	False	False	False
11	False	False	False	False	False	False	False	False	False	False	False	False	False

18249 rows x 13 columns

Figura 4: Comando .isnull() que permite apreciar de forma general si los valores son Nulos

```

#Revisar si existen valores nulos
#De manera Global

aguacates_data.isnull()

#Revisar valores que existen dentro de una columna especifica

#aguacates_data.Date.isnull().unique()
#aguacates_data.AveragePrice.isnull().unique()
#aguacates_data.TotalVolume.isnull().unique()
#aguacates_data.PLU4046.isnull().unique()
#aguacates_data.PLU4225.isnull().unique()
#aguacates_data.PLU4770.isnull().unique()
#aguacates_data.TotalVolume.isnull().unique()
#aguacates_data.SmallBags.isnull().unique()
#aguacates_data.LargeBags.isnull().unique()
#aguacates_data.XLargeBags.isnull().unique()
#aguacates_data.type.isnull().unique()
#aguacates_data.year.isnull().unique()
aguacates_data.region.isnull().unique()

#O de forma mas profunda

#aguacates_data.loc[aguacates_data.region.isnull()]

array([False])

```

Figura 5: Comandos que permiten visualizar los diferentes valores de casillas que conforman una columna

Para las demás situaciones se puede hacer una revisión general, presenciando un conjunto de operaciones con los datos que nos permitan entender si los datos tienen errores al dar resultados incoherentes, estos se pueden realizar por medio del siguiente comando:

```

#Funciones como .describe permiten ver las operaciones como mean,count,std,min,max
#Para columnas como AveragePrice, TotalVolume y cada PLU
#aguacates_data.PLU4046.describe()
#aguacates_data.PLU4225.describe()
#aguacates_data.PLU4770.describe()
#aguacates_data.TotalVolume.describe()
aguacates_data.AveragePrice.describe()

```

count	18249.000000
mean	1.405978
std	0.402677
min	0.440000
25%	1.100000
50%	1.370000
75%	1.660000
max	3.250000
Name: AveragePrice, dtype: float64	

Figura 6: Comando .describe() que permite visualizar diferentes operaciones sobre una columna

No se puede olvidar que se tiene columnas de diferente tipo, las cuales no permiten realizar operaciones numéricas, para este caso se hará uso de comandos como .unique() que me permite conocer los diferentes valores que posee una columna:

```
#Es importante reconocer vales unicos en columnas como type, year y region
#aguacates_data.type.unique()
#aguacates_data.year.unique()
aguacates_data.region.unique()

array(['Albany', 'Atlanta', 'BaltimoreWashington', 'Boise', 'Boston',
       'BuffaloRochester', 'California', 'Charlotte', 'Chicago',
       'CincinnatiDayton', 'Columbus', 'DallasFtWorth', 'Denver',
       'Detroit', 'GrandRapids', 'GreatLakes', 'HarrisburgScranton',
       'HartfordSpringfield', 'Houston', 'Indianapolis', 'Jacksonville',
       'LasVegas', 'LosAngeles', 'Louisville', 'MiamiFtLauderdale',
       'MidSouth', 'Nashville', 'NewOrleansMobile', 'NewYork',
       'Northeast', 'NorthernNewEngland', 'Orlando', 'Philadelphia',
       'PhoenixTucson', 'Pittsburgh', 'Plains', 'Portland',
       'RaleighGreensboro', 'RichmondNorfolk', 'Roanoke', 'Sacramento',
       'SanDiego', 'SanFrancisco', 'Seattle', 'SouthCarolina',
       'SouthCentral', 'Southeast', 'Spokane', 'StLouis', 'Syracuse',
       'Tampa', 'TotalUS', 'West', 'WestTexNewMexico'], dtype=object)
```

Figura 7: Visualización comando .unique()

III. RESULTADOS

Se realiza la descripción detallada de lo que se quiere realizar con este proyecto para poder tener una visión adecuada de los datos que se seleccionaron. Con ello realizado se procede a la carga del Dataset al Notbebook de forma correcta, con el cual se realizó diferentes procedimientos por medio de comandos aprendidos en el curso para poder evidenciar y comprender los datos de mejor forma .

Además de ello se hicieron algunas modificaciones dentro de los datos para poder realizar operaciones de forma adecuada para las siguientes entregas del presente proyecto.

IV. CONCLUSIONES

- Se debe realizar de forma correcta el planteamiento del problema para poder reconocer que se quiere realizar con el proyecto
- Es necesario ir trabajando de forma práctica con los datos para revisar si estos tienen las características necesarias para cumplir con el fin del proyecto
- Es importante realizar el pre procesamiento de datos para asegurar que los datos estarán preparados para el uso de un Modelo de Marchine Learning

V. ENLACES

- [Repositorio GitHub](#)
- [Explicación en Video](#)

REFERENCIAS

- [1] Kiggins, J. (n.d.). Avocado Prices [Data set]
- [2] Teran, M. (n.d.). machinelearning.