Proyecto Inteligencia Artificial

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

Resumén

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 especifico, todo esto por medio de la experimentación con modelos de Marchine 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

Marchine learning, Matriz, Preprocesamiento de datos , Pandas

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 vendedores minoristas sobre sus ventas, las cuales se extrajeron semanalmente. Este proyecto se guiará de acuerdo a los datos proporcionados, por ende, se seleccionó el siguiente problema o caso de estudio principal:

- De acuerdo a las formas de cultivo existentes que se plantea en los campos del dataset; cómo se puede visualizar las ventas de los aguacates de acuerdo a factores como el promedio del precio, el tipo de producto establecido por el PLU u otros factores en relación a las formas de cultivo , además de ello, el cómo se puede predecir a que forma de cultivo pertenece mediante las características que provee el dataset.

II. Objetivo

Esta primera entrega tendrá como objetivo principal explicar los resultados obtenidos del preprocesamiento de datos realizados dentro de Notebook de]upyter , junto a las métricas establecidas por la Guide CRISP-DM de IBM SPSS requeridas para la presentación del presente proyecto

III. Entendiendo los datos

III-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 Marchine learning que se escoja en la entrega siguiente,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 del objetivo que se quiere lograr con el 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 dia aun continua generando datos sobre la venta de aguacates,por ende se puede considerar que son datos confiables,que permitirán el desarrollo adecuado del proyecto. Otra cosa importante de resaltar es que en dentro del Notebook ya han se realizadp operaciones básicas sobre este Dataframe, generando resultados coherentes que dan idea de como es la estructura de los datos del dataset, promoviendo los recursos necesarios para enfocar el proyecto a algun campo especifico.

III-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'])
#Agregar columna Status para poder identificar con un numero las dos formas que se usan para cultivar este producto
aguacates_data['Status'] = np.where(aguacates_data.type=="conventional", '1', '2')
#Convertir dicha columna en un entero para poder crear el digrama de correlación
aguacates_data['Status']=aguacates_data['Status'].astype('int64')
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
Status                   int64
dtype: object
```

Figura 3: Comando .dtypes Pandas

En esta anterior imagen también se ve como dentro del codigo se encuentra el comando pd.to_datatime, el cual nos permite cambiar el tipo de la Columna Date a fecha o el comandos como astype() que permite cambiar el tipo de una columna a otros tipos.

III-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 × 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 columnmans 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 diferente valores que posee una columna especifica:

```
#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()

Finalmente, para visualizar mejor los datos se decide genera la matriz de correlación para ver mejor las características de los dos tipos de cultivo que existen, para ello se debe apreciar en la Figura 3 la creación de una nueva columna denominada Status cuya función es establecer un numero de acuerdo a las clases para poder realizar la matriz de correlación, con este proceso se garantiza una correcta creación de esta.

```
#Crear gráficos que permitan visualizar la correlación de características
cmap = cm.get_cmap('Set1')
dif_class=np.asarray(aguacates_data.Status)
aguacates_data=aguacates_data.drop(columns=["Status"])
pd.plotting.scatter_matrix(aguacates_data, c=dif_class, cmap=cmap, figsize=(15, 15));
```

Figura 8: Creación Matriz de correlación

Se puede visualizar como se hace de la columna Status para establecer parámetros dentro de las funciones necesarias para generar la matriz. El resultado del código anterior de acuerdo a actual Dataset trabajado seria :

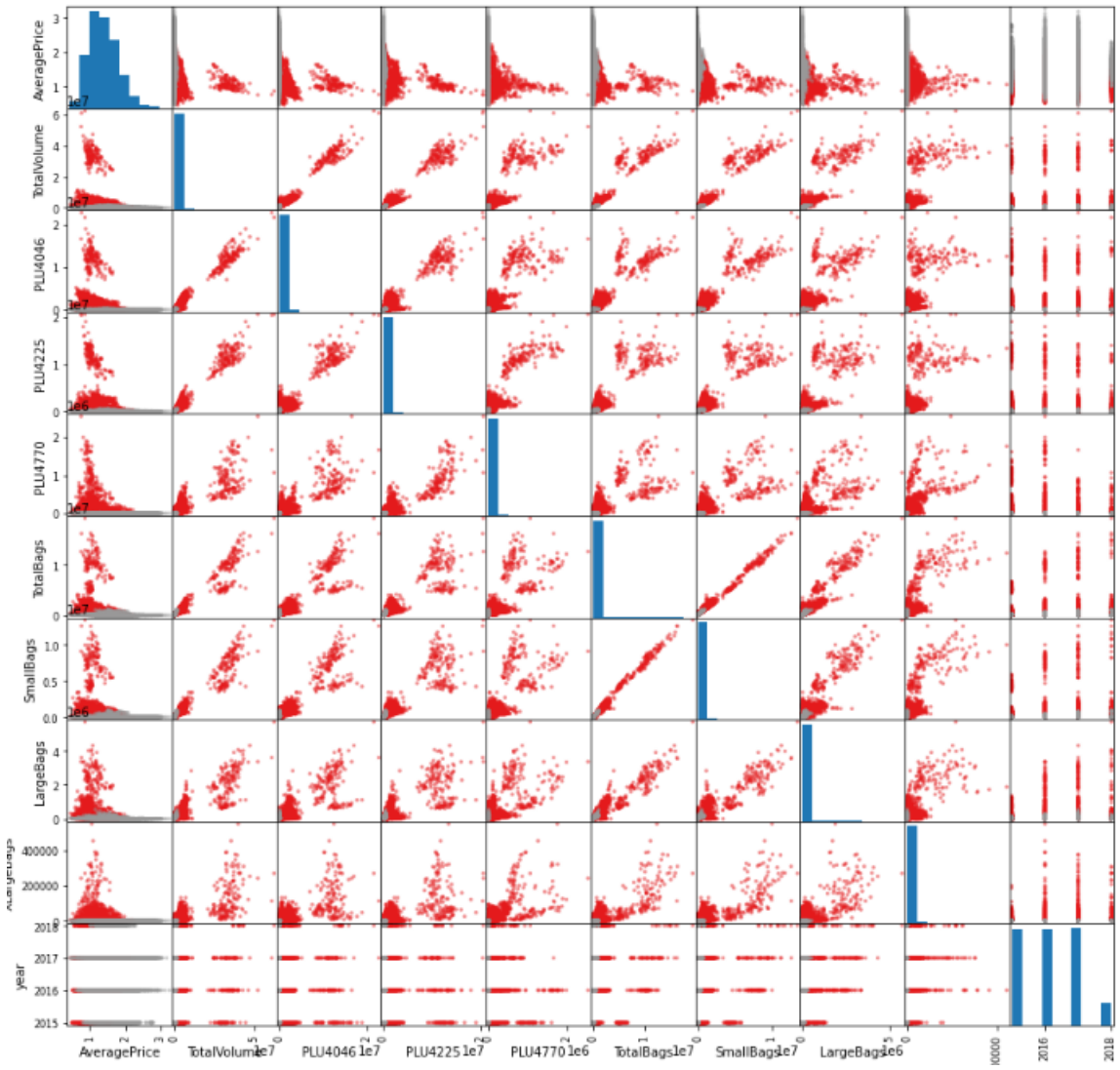

Figura 9: Visualización Matriz de correlación

IV. 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.

V. 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

VI. ENLACES

- Repositorio GitHub
- Explicación en Video

REFERENCIAS

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