UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN
FACULTAD DE INGENIERÍA MECÁNICA Y ELÉCTRICA

Redes Neuronales

PIA

Nombre	Matricula	Carrera
Gaston Tejeda Villaseñor	1889295	ITS

Periodo Febrero-Junio 2021

PIA

Indice

Resumen

Yo trabajo como practicante para una empresa de software, en que mi principal responsabilidad es dar soporte a la base de datos del Centro Universitarios Contra el Cáncer. Durante mi tiempo trabajando ahí he notado que hay varios procesos o funcionamientos que no son eficientes o podrían ser optimizados. Uno de ellos es su sistema de inventarios.

En la actualidad ellos solamente cuentan con un sistema de inventarios que registra la historia de movimientos de medicinas. Para la adquisición de los medicamentos no se hace proceso ni planeación alguna, simplemente se compra una determinada cantidad (dependiendo del medicamento) cada vez que se acaba o esta por acabarse. Sobra decir que este sistema es ineficaz, y yo que manejo su base de datos me he dado cuenta de esto, es por eso que decidí hacer mi proyecto de predicciones, orientado a inventarios de mercancía.

Propongo crear un modelo de red neuronal que realice predicciones en determinados periodos de tiempo para saber la demanda de un articulo a partir de su historial.
Lamentablemente, por términos de privacidad y política de confidencialidad no me es permitido sacar información de las bases de datos del hospital, pero encontré unos datasets en kaggle de inventarios de mercancía bastante parecidos a los del hospital que estaré utilizando en este proyecto.

Mi plan es realizar este proyecto con los datasets de kaggle y una vez este sea calificado, presentar este mismo proyecto a mis superiores como propuesta para implementarlo en el hospital.

Características

Para este caso o problema, se tienen los datasets test_data y train_data. Dichos datasets contienen las columnas:

- Id: identificador en el csv (solamente el archivo test lo tiene)
- Date: Fecha en que se realizó la observación
- Product_Identifier: identificador de articulo
- Departamento_identifier: identificador de departamento
- Category: categoría del articulo
- Outlet: identificador de la tienda
- State: nombre del estado
- Sales: ventas de los artículos en la observación (solamente train lo tiene)

A simple vista uno puede notar que estos datasets no tiene un atributo sumamente importante, que seria el costo de cada atributo en las fechas que fueron vendidos, para poder tomar en cuenta si los descuentos o cambios de precio afectan a las ventas de los artículos, pero en realidad esa es justo la razón por la que elegí este dataset: en el hospital

donde trabajo no hay un registro histórico de precios, solo se conocen los actuales, por lo que prefería trabajar con algo parecido a eso, encontrando la relación entre los diferentes atributos existentes y a partir de estos obteniendo predicciones acertadas.
Según lo que puedo ver de los datasets creo yo que los factores mas influyentes en las ventas de artículos son la fecha y el identificador de tienda.

Objetivo

El alumno deberá proponer un problema de análisis de datos y resolverlo usando cualquiera de las técnicas computacionales estudiadas en el curso. Deberá entregarse PDF con las siguientes secciones: Resumen, Características del problema a estudiar y los datos, Objetivos, Metodología (con explicación de la técnica o técnicas computacionales empleadas) y Resultados.

El objetivo de esta actividad es encontrar la correlación de los diferentes atributos que tiene cada articulo con sus ventas y generar un modelo de nos ayude a predecir las ventas en el futuro, para poder estar preparados para ellas. También se planea generar una matriz de correlación de los datos de entrenamiento para identificar cuales son los factores principales que influyen en las ventas de los medicamentos y cuales no son tan importantes o no deberían ser tomados en cuenta.

Metodología

Comenzamos por importar todas las librerías que se van a utilizar

```
from sklearn.neural_network import MLPRegressor
import pandas as pd
import numpy as np
import pickle
import matplotlib.pyplot as plt
```

- MLPRegressor será utilizada para el entrenamiento de la red neuronal.
- Pandas será utilizado para la lectura de los archivos y gestión de los DataFrames.
- Numpy será utilizado para trabajar con vectores y matrices de datos.
- Pickle es utilizado a la hora de generar el modelo.
- Pyplot es utilizado para crear los gráficos que se utilizarán en este proyecto.

Se estudian los datasets. Nos podemos dar cuenta de que el archivo test tiene una columna extra, que el archivo train no tiene. Por motivos prácticos esta columna será removida durante el procedimiento. También podemos notar que el archivo train tiene una columna mas que test, saldos. Sin embargo, esto esta bien pues necesaria para entrenar a nuestro modelo.

Se entrenará la red neuronal con los datos de entrenamiento con el método MLPRegressor, que entrena de manera iterativa ya que en cada paso de tiempo la derivada parcial de perdida de función con respecto a los parámetros del modelo es calculada para optimizar los parámetros.

	date	product_identifier	department_identifier	category_of_product	outlet	state	sales
0	2012-01-01	74	11	others	111	Maharashtra	0
1	2012-01-01	337	11	others	111	Maharashtra	1
2	2012-01-01	423	12	others	111	Maharashtra	0
3	2012-01-01	432	12	others	111	Maharashtra	0
4	2012-01-01	581	21	fast_moving_consumer_goods	111	Maharashtra	0
...	...	...	...	...	...	...	...
394995	2014-02-28	2932	33	drinks_and_food	333	Kerala	2
394996	2014-02-28	2935	33	drinks_and_food	333	Kerala	8
394997	2014-02-28	3004	33	drinks_and_food	333	Kerala	0
394998	2014-02-28	3008	33	drinks_and_food	333	Kerala	0
394999	2014-02-28	3021	33	drinks_and_food	333	Kerala	0

Gráfica con los datos del archivo .csv Train

	id	date	product_identifier	department_identifier	category_of_product	outlet	state
0	1	2014-03-01	74	11	others	111	Maharashtra
1	2	2014-03-01	337	11	others	111	Maharashtra
2	3	2014-03-01	423	12	others	111	Maharashtra
3	4	2014-03-01	432	12	others	111	Maharashtra
4	5	2014-03-01	581	21	fast_moving_consumer_goods	111	Maharashtra
...	...	...	...	...	...	...	...
15495	15496	2014-03-31	2932	33	drinks_and_food	333	Kerala
15496	15497	2014-03-31	2935	33	drinks_and_food	333	Kerala
15497	15498	2014-03-31	3004	33	drinks_and_food	333	Kerala
15498	15499	2014-03-31	3008	33	drinks_and_food	333	Kerala
15499	15500	2014-03-31	3021	33	drinks_and_food	333	Kerala

Gráfica de los datos del archivo .csv Test

Ambos archivos cuentan con varias columnas para datos de tipo string que deberán ser optimizadas para el algoritmo de aprendizaje.

A partir del archivo de entrenamiento se crea un plot de correlación para poder examinar los pesos de la influencia de los datos entre sí.

```
dTest = pd.concat([dTest.drop('date', axis = 1),
          (dTest.date.str.split("-").str[:3].apply(pd.Series)
          .rename(columns={0:'year', 1:'month', 2:'day'}))], axis = 1)
    dTrain = pd.concat([dTrain.drop('date', axis = 1),
          (dTrain.date.str.split("-").str[:3].apply(pd.Series)
          .rename(columns={0:'year', 1:'month', 2:'day'}))], axis = 1)

    dTest = pd.concat([dTest, pd.get_dummies(dTest['category_of_product'], pre
fix='dt')], axis=1)
    dTest.drop(['category_of_product'], axis=1, inplace=True)
    dTrain = pd.concat([dTrain, pd.get_dummies(dTrain['category_of_product'],
prefix='dt')], axis=1)
    dTrain.drop(['category_of_product'], axis=1, inplace=True)
    yTrain = dTrain['sales']
    xTrain = dTrain.copy()
    xTrain.drop(['sales'], axis=1, inplace=True)
```

Los campos 'category_of_product' y `'category_of_product'` se separan en clases dependiendo de los datos que almacenan, mientras que los campos Date se separan en 3 categorías: Año, Mes y Día.

Se genera Diagrama de Relación.

```
corrPlot(dTrain)
```

```
def corrPlot(dt):
    corr = dt.corr()
    plt.figure(num=None, figsize=(13, 13), dpi=80, facecolor='w', edgecolor='k
')
    corrMat = plt.matshow(corr, fignum = 1)
    plt.xticks(range(len(corr.columns)), corr.columns, rotation=90)
    plt.yticks(range(len(corr.columns)), corr.columns)
    plt.colorbar(corrMat)
    plt.title(f'Correlation Matrix for PIA', fontsize=15)
    plt.show()
```

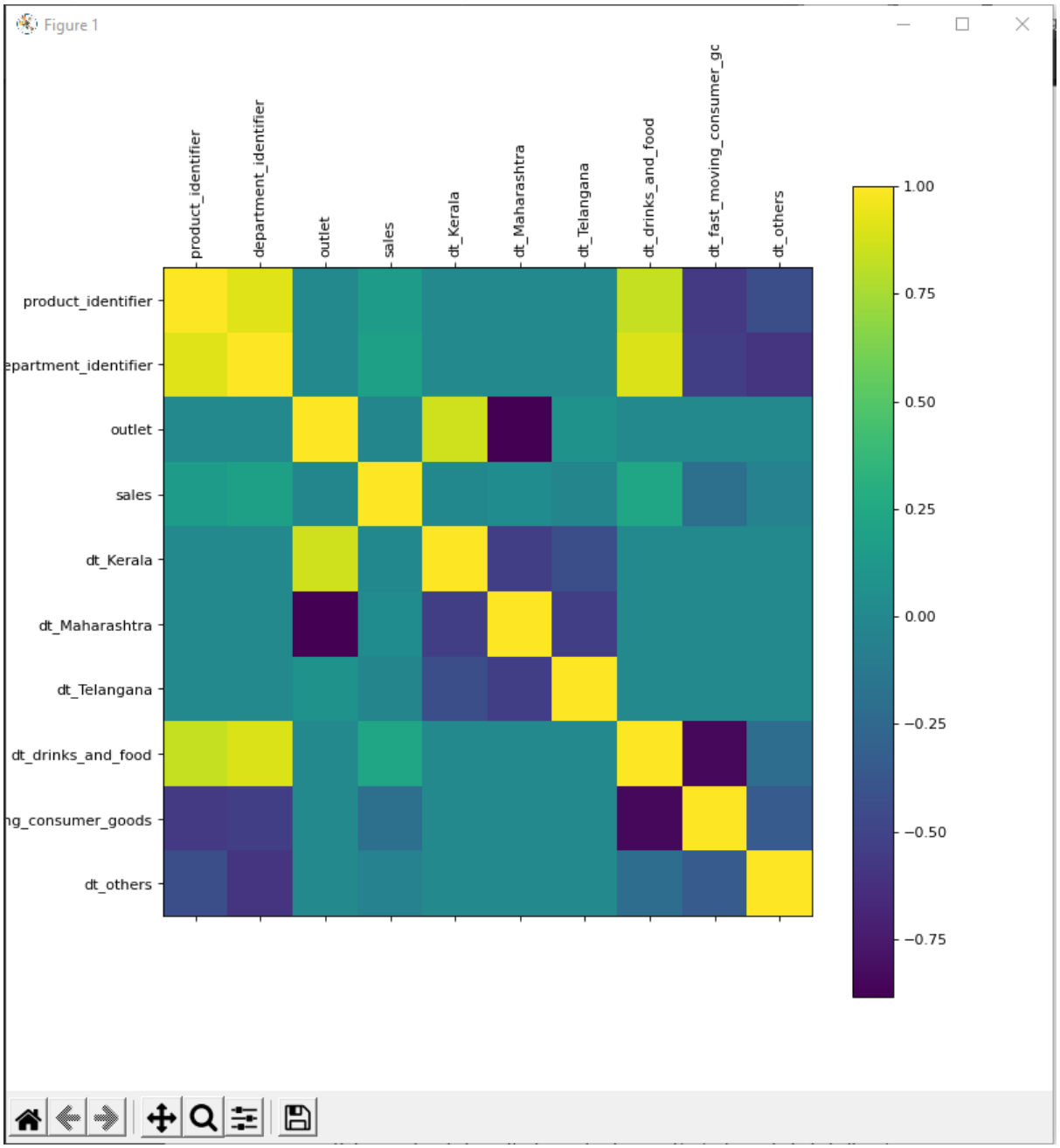
Se entrena el clasificador con los datos de entrenamiento.

```
clf = MLPRegressor(solver='lbfgs', tol=1e-
100, hidden_layer_sizes=(3,), max_iter=500)
clf.fit(xTrain, yTrain)
```

Por medio del clasificador se genera el modelo del proyecto.

```
#se usa el modelo entrenado para
    predicted = clf.predict(dTest)
    dTest = pd.concat([dTest, pd.DataFrame(np.rint(predicted))], axis=1)
```

Resultados



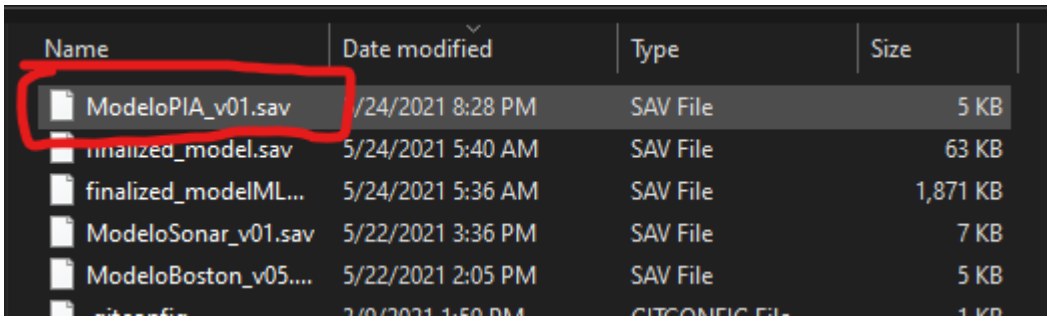
Mediante este plot de correlación podemos examinar las relaciones de los valores de los datos entre sí, por ejemplo, podemos notar que las ventas de artículos están correlacionadas positivamente principalmente a datos como el identificador de producto y el identificador de departamento. Por otro lado, diente su correlación mas negativa con los artículos pertenecientes a la categoría de consumer goods.

A continuación, se presenta el código empleado para analizar los datasets, generar la matriz de correlación y entrenar y generar el modelo para hacer las predicciones de ventas de artículos. Lamentablemente en los datasets proporcionados por kaggle no venían los datos de ventas por articulo para test, simplemente el archivo de test que permitía generar predicciones, por lo que estas no pueden ser verificadas.

```
from sklearn.neural_network import MLPRegressor
from sklearn.metrics import mean_squared_error
import pandas as pd
import numpy as np
import pickle
import matplotlib.pyplot as plt

def corrPlot(dt):
    corr = dt.corr()
    plt.figure(num=None, figsize=(13, 13), dpi=80, facecolor='w', edgecolor='k
')
    corrMat = plt.matshow(corr, fignum = 1)
    plt.xticks(range(len(corr.columns)), corr.columns, rotation=90)
    plt.yticks(range(len(corr.columns)), corr.columns)
    plt.colorbar(corrMat)
    plt.title(f'Correlation Matrix for PIA', fontsize=15)
    plt.show()



if __name__ == '__main__':
    #Leer csv's
    dTrain = pd.read_csv(r'C:\Users\gasto\Documents\UANL\7mo_Semestre_Strong\R
edesNeuronales\PIA\train_data.csv')
    dTest = pd.read_csv(r'C:\Users\gasto\Documents\UANL\7mo_Semestre_Strong\Re
desNeuronales\PIA\test_data.csv')

    #Separar Valores Binarios
    dTest = pd.concat([dTest, pd.get_dummies(dTest['state'], prefix='dt')], ax
is=1)
    dTest.drop(['state'], axis=1, inplace=True)
    dTest.drop(['id'], axis=1, inplace=True)
    dTrain = pd.concat([dTrain, pd.get_dummies(dTrain['state'], prefix='dt')],
 axis=1)
    dTrain.drop(['state'], axis=1, inplace=True)

    dTest = pd.concat([dTest.drop('date', axis = 1),
          (dTest.date.str.split("-").str[:3].apply(pd.Series)
          .rename(columns={0:'year', 1:'month', 2:'day'}))], axis = 1)
    dTrain = pd.concat([dTrain.drop('date', axis = 1),
          (dTrain.date.str.split("-").str[:3].apply(pd.Series)
          .rename(columns={0:'year', 1:'month', 2:'day'}))], axis = 1)

    dTest = pd.concat([dTest, pd.get_dummies(dTest['category_of_product'], pre
fix='dt')], axis=1)
    dTest.drop(['category_of_product'], axis=1, inplace=True)
```

```
    dTrain = pd.concat([dTrain, pd.get_dummies(dTrain['category_of_product'],
prefix='dt')], axis=1)
    dTrain.drop(['category_of_product'], axis=1, inplace=True)

    corrPlot(dTrain)

    yTrain = dTrain['sales']
    xTrain = dTrain.copy()
    xTrain.drop(['sales'], axis=1, inplace=True)

    #Se entrena el clasificador
    clf = MLPRegressor(solver='lbfgs', tol=1e-
100, hidden_layer_sizes=(3,), max_iter=500)
    clf.fit(xTrain, yTrain)

    #se usa el modelo entrenado para
    predicted = clf.predict(dTest)
    dTest = pd.concat([dTest, pd.DataFrame(np.rint(predicted))], axis=1)
    pickle.dump(clf, open("ModeloPIA_v01.sav", 'wb'))
```

Se generó un modelo entrenado para generar predicciones de ventas de artículos para la base de datos de inventarios y ventas proporcionadas por kaggle.



Conclusión

Aunque no pudimos verificar la exactitud de la predicción de nuestro modelo, se debe recordar que el propósito principal de este proyecto es establecer una base o precedente para poder ofrecer esta herramienta a los ejecutivos del hospital. Incluso sin una verificación de la predicción me parece que la matriz de correlación de los datos y su análisis son lo suficientemente atractivos para que se acepte una propuesta y poner en marcha su implementación. Me parece los modelos predictivos y los plots de matrices de correlación son herramientas bastante útiles, pues permiten examinar grandes cantidades de datos y sus relaciones sin necesidad de interactuar directamente con las agobiantes cantidades de información, volviendo cualquier proceso mucho más eficiente y presentando una herramienta de planeación y administración bastante efectiva.