



INSTITUTO POLITÉCNICO NACIONAL



"ESCUELA SUPERIOR DE CÓMPUTO"

Integrantes:

- **Velasco Huerta Ángel Eduardo**
- **Hernández Clemente Samantha**

Práctica No. 7

Aplicación de tareas de aprendizaje supervisado.

Materia:

Data Mining

Profesor:

Zagal Flores Roberto Eswart

Grupo:

3CV18

Introducción:

En la presente práctica, se aplicarán conocimientos previos vistos en el curso. Referentes a la introducción al aprendizaje de máquina, mediante técnicas de clasificación y predicción referentes al análisis de datos.

Para ello, se probará un modelo de predicción de datos en un dataset ligeramente modificado y seccionado, para poder entrenar con el 80% de los datos el modelo de Support Vector Regression, y posteriormente, usar el 20% restante para testing y ajuste. Esto se logrará gracias a las librerías de Python que nos permiten implementar estos modelos sin un análisis matemático tan profundo.

Finalmente, realizaremos una breve exploración de los datos de testing y los de prueba para poder obtener conclusiones más precisas relevantes al desempeño del modelo implementado.

1- Elija un problema de clasificación o predicción aplicado a alguna de las bases de datos mencionadas, indique en un párrafo de texto el problema elegido y que desea obtener.

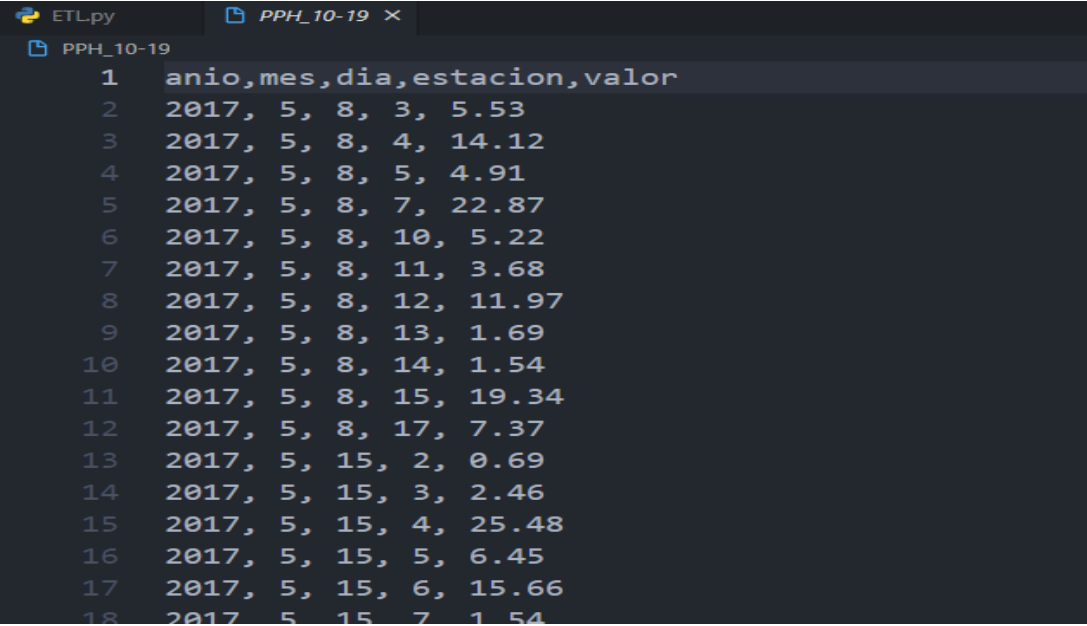
Problema de predicción:

Se pronostica el nivel de precipitación pluvial por mes, semana, día y estaciones de depósito atmosférico de PP de la CDMX, utilizando 3 años de datos (2017-2019), no se pudieron considerar los años 2020 y 2021 pues los datasets no se encuentran disponibles para su descarga en el portal de datos abiertos:

<http://www.aire.cdmx.gob.mx/default.php?opc=%27aKBk%27>

2-Diseñe dos tablas de hechos, o datasets, para la fase de entrenamiento y de pruebas. Documente y explique por qué eligió las dimensiones o columnas y el nivel de granularidad de datos.

Primero, utilizamos la herramienta de ETL creada previamente, para modificarla y solo utilizar la extracción y transformación de 3 años, quedándonos una estructura con los siguientes valores:



	anio	mes	dia	estacion	valor
1	2017	5	8	3	5.53
2	2017	5	8	4	14.12
3	2017	5	8	5	4.91
4	2017	5	8	7	22.87
5	2017	5	8	10	5.22
6	2017	5	8	11	3.68
7	2017	5	8	12	11.97
8	2017	5	8	13	1.69
9	2017	5	8	14	1.54
10	2017	5	8	15	19.34
11	2017	5	8	17	7.37
12	2017	5	15	2	0.69
13	2017	5	15	3	2.46
14	2017	5	15	4	25.48
15	2017	5	15	5	6.45
16	2017	5	15	6	15.66
17	2017	5	15	7	1.54

De esta forma, tenemos dimensiones de tiempo, que cambian con mes y día, y una geográfica (estación) pues los valores si cambian dependiendo la estación. Ahora, dividiremos esto en dos datasets, uno de training, que contendrá 1013 registros (80% del dataset original) y uno de testing que contendrá el resto.

3-Utilice el código adjunto con nombre “svmPredictorBetaParaModificarParaClasificador.zip” el cual consiste en un ejemplo en Python para predecir valores usando SVM

Entre los principales cambios realizados al documento, fueron las columnas trabajadas, primero, se realizó el proceso con las siguientes columnas:

```
X_train = dfTraining[["año", "mes", "dia", "estacion"]]
y_train = dfTraining.valor

X_testing = dfTesting[["año", "mes", "dia", "estacion"]]
y_testing = dfTesting.valor
```

Y el resultado nos arrojó errores muy grandes al momento de aplicar el algoritmo de predicción:

Predicted Value:	-102.45098930489883	Real value:	35.8	% Error:	386.175947779047
Predicted Value:	-101.70656729225487	Real value:	30.4	% Error:	434.56107661925955
Predicted Value:	-101.47567245163452	Real value:	40.0	% Error:	353.6891811290863
Predicted Value:	-481.36883059089087	Real value:	122.6	% Error:	492.63363017201544

Por lo que realizamos una modificación para solo tomar en cuenta mes, día y estación, pues el año no aportaba mucho a la predicción de datos, específicamente por la estructura del dataset.

Así que modificamos también el valor de epsilon, que identifica que tan permisible es el margen de error dentro del vector.

```
clf = SVR(C=1.0, epsilon=0.01)
```

Finalmente, corrimos el programa con estas columnas para el entrenamiento y la prueba:

```
X_train = dfTraining[["mes", "dia", "estacion"]]
y_train = dfTraining.valor

X_testing = dfTesting[["mes", "dia", "estacion"]]
y_testing = dfTesting.valor
```

Y obtuvimos valores mucho más acertados, si bien, seguimos sin acercarnos tanto a los valores, esta vez ya no son tan disparados, y nos dan una aproximación más cercana:

Predicted Value:	20.379151060717696	Real value:	27.9	% Error:	26.956447811047678
Predicted Value:	20.480740206753307	Real value:	65.5	% Error:	68.73169434083464
Predicted Value:	20.600117063483758	Real value:	18.3	% Error:	12.568945702097034

Ahora, corremos el algoritmo con otra prueba, para seguir verificando distintos valores de prueba, y obtenemos lo siguiente:

Predicted Value:	30.58820947439449	Real value:	30.0	% Error:	1.96069824798163
Predicted Value:	28.74189992679328	Real value:	58.6	% Error:	50.952389203424445
Predicted Value:	27.264446706265403	Real value:	56.4	% Error:	51.658782435699635

De la misma forma, son valores mucho que en ciertas ocasiones, se acercan más que como estaba antes, pues obteníamos errores de hasta 1000%

Ahora, procedemos a realizar 15 pruebas individuales para revisar el algoritmo implementado:

1-

Predicted Value:	4.6049377712599835	Real value:	0.6	% Error:	667.4896285433307
------------------	--------------------	-------------	-----	----------	-------------------

2-

Predicted Value:	8.02486211146902	Real value:	11.7	% Error:	31.41143494470923
------------------	------------------	-------------	------	----------	-------------------

3-

Predicted Value:	14.747283162334611	Real value:	13.1	% Error:	12.574680628508483
------------------	--------------------	-------------	------	----------	--------------------

4-

Predicted Value:	24.09057346648308	Real value:	18.7	% Error:	28.826596077449622
------------------	-------------------	-------------	------	----------	--------------------

5-

Predicted Value:	32.12008050490351	Real value:	30.2	% Error:	6.357882466567926
------------------	-------------------	-------------	------	----------	-------------------

6-

Predicted Value:	23.684994109928667	Real value:	23.9	% Error:	0.8996062346080832
------------------	--------------------	-------------	------	----------	--------------------

7-

Predicted Value:	25.230070319234244	Real value:	26.1	% Error:	3.333063910979914
------------------	--------------------	-------------	------	----------	-------------------

8-

Predicted Value:	23.34568682036981	Real value:	21.6	% Error:	8.081883427638003
------------------	-------------------	-------------	------	----------	-------------------

9-

Predicted Value:	25.129913938422618	Real value:	9.9	% Error:	153.8375145295214
------------------	--------------------	-------------	-----	----------	-------------------

10-

Predicted Value:	29.124047555206594	Real value:	2.6	% Error:	1020.1556752002535
------------------	--------------------	-------------	-----	----------	--------------------

11-

Predicted Value:	30.15023769456912	Real value:	13.9	% Error:	116.90818485301526
------------------	-------------------	-------------	------	----------	--------------------

12-

Predicted Value:	29.542908841557622	Real value:	21.6	% Error:	36.77272611832232
------------------	--------------------	-------------	------	----------	-------------------

13-

Predicted Value:	24.61774578688564	Real value:	28.8	% Error:	14.521716017758202
------------------	-------------------	-------------	------	----------	--------------------

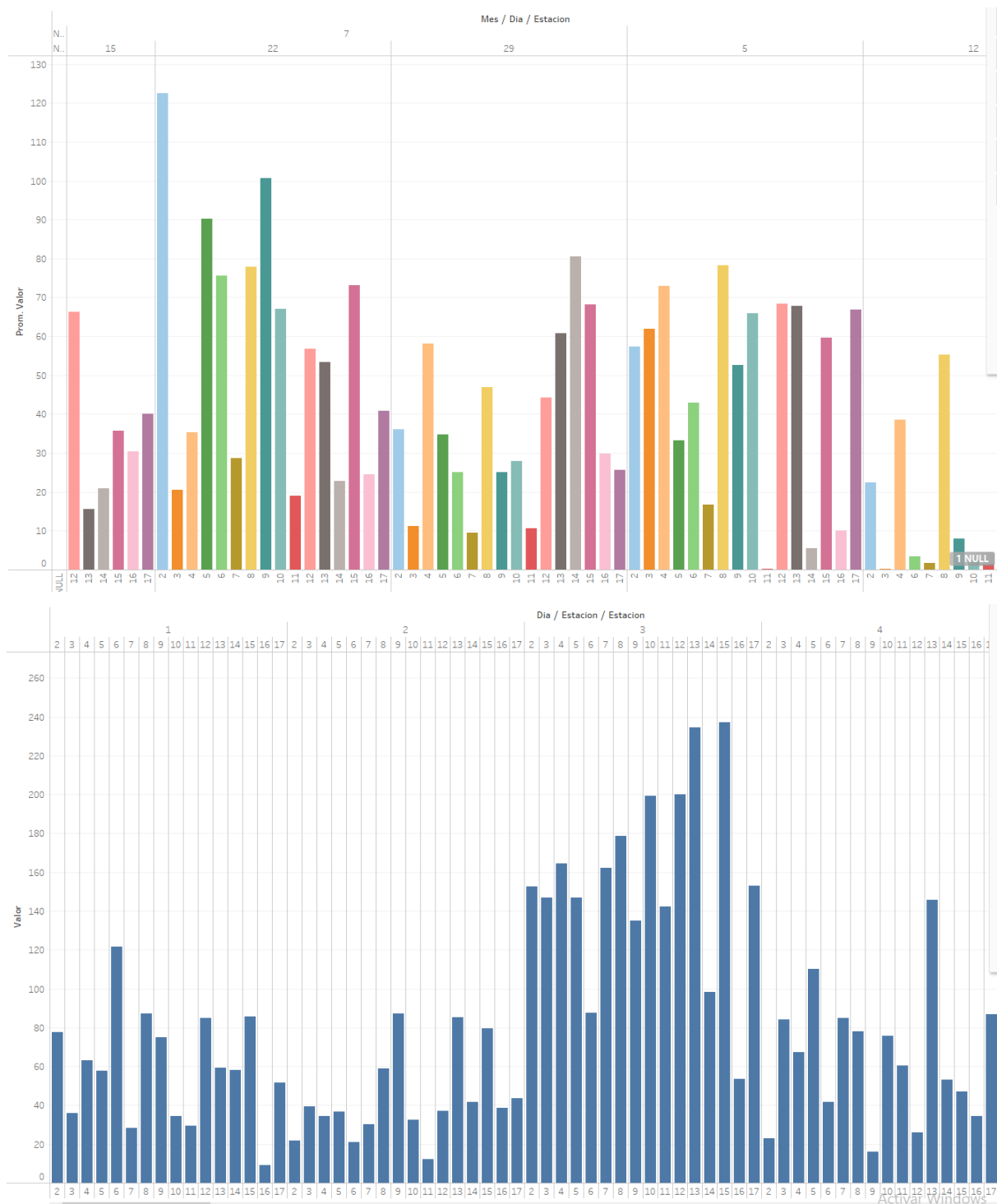
14-

Predicted Value:	23.668961598176985	Real value:	21.8	% Error:	8.573218340261395
------------------	--------------------	-------------	------	----------	-------------------

15-

Predicted Value:	23.622011939743818	Real value:	21.1	% Error:	11.952663221534673
------------------	--------------------	-------------	------	----------	--------------------

Análisis exploratorio del dataset de prueba y entrenamiento:



Dentro del análisis exploratorio, podemos ver un detalle que me pareció importante dentro del algoritmo, y es que, no parece haber una relación clara entre los datos, estos se disparan de manera aleatoria, pues recordemos estamos hablando de un fenómeno climatológico, por lo que predecir algo como esto, puede ser bastante complicado, y necesitar de información más precisa, pues podemos ver que si bien las tendencias en el mes 3, suben, en años anteriores, el mes 2 era el que presentaba mas lluvias, por lo que no es tan facil determinar que valores puede tomar, por lo que solo en los meses 9 y 10, que es

cuando menos llueve, el algoritmo registro valores un poco menores, que si bien no acertaban al 100%, seguían la tendencia de bajada en la precipitación pluvial.

Conclusiones:

Velasco Huerta Ángel Eduardo: En esta práctica, pudimos poner a práctica algo de lo visto en la unidad 4, que es lo relacionado con aprendizaje de máquina y breves conceptos para la predicción y categorización de datos, aplicamos un algoritmo de SVR (support vector regression, que básicamente divide de un lado de un vector, los valores que acepta el algoritmo, y del otro aquellos que no coinciden con el entrenamiento.

Al momento de trabajar con esto, teníamos bastante incertidumbre de los resultados, pues primero obtuvimos algunos bastantes alocados, y que se desviaban bastante del objetivo de predicción, pero al momento de modificar algunos valores, logramos que estos, se acercaran un poco más a las tendencias que mostraba el dataset original, que si bien no son bastante claras, y tienden a ser impredecibles, si se mostraba que en los meses con menos registro de PP, el algoritmo brindaba valores menores.

Hernández Clemente Samantha: Con el desarrollo de esta práctica logramos comprender mejor los temas vistos en la unidad 4 de este curso, que es principalmente aprendizaje de máquina como la predicción y categorización de datos aplicando un algoritmo SVR.

Al principio de realizar la práctica no estábamos muy seguros de cómo llevarla a cabo pero a través de ver el material de clase brindado poco a poco nos fuimos dando una idea para desarrollar la práctica. En el transcurso del desarrollo tuvimos algunas dudas porque obtuvimos resultados dispares con el objetivo de predicción, pero logramos que se acercaran a la tendencia del dataset original.