Astronomy & Astrophysics manuscript no. output
April 1, 2022

Prediccion de posibles impactos de asteroides con la tierra usando modelos de machine learning

Simulacion de sistemas

Daniel Sierra Mejia

Universidad de Antioquia Medellín, Colombia
e-mail: daniel.sierram@udea.edu.co

sent 31 march 2022

ABSTRACT

Context. Investigar la posibilidad de impactos de asteroides con la tierra en un periodo futuro estimado.
Aims. Debido a la cantidad de datos reunidos de diferentes asteroides se ha podido calcular factores de riesgo que indiquen una posible colision con la tierra.
Methods. Se utilizaron metodos de aprendizaje de maquina , modelos clasicos como regresion polinomial multiple , redes neuronales feed forward , arboles de desicion y maquinas de vectores de soporte .Se refinaron los datos iniciales para seleccionar y extraer caracteristicas mediante Sequential forward selection .
Results. Se lograron predicciones similares con el tratamiento de las variables pero reduciendo la dimensionalidad significativamente..

Key words. support vector machine SVR – Polinomial regression – feed forward neural network RNN – desition tree regression .

1. Introduccion

Como se sabe , los cuerpos celestes orbitan elipses ,la organización Minor Planet Center está constantemente monitoreando por la búsqueda de nuevos asteroides que orbitan en elipses alrededor del sol .Cuando se descubre un nuevo planeta , a este se le toman varias características para definir su órbita ;luego se comparan estas características que teorizan una órbita y se compara con futuras observaciones de dicho planeta .Esto quiere decir que se compara la órbita teórica de donde se cree que el asteroide aparecerá y se compara con la observación real por donde el asteroide de hecho apareció en un momento dado;de esta forma se ajustan los valores de las características de la órbita teorizada para que concuerde con la real.

Sentry es un sistema de monitoreo que evalúa los riesgos de un impacto con la tierra para cualquier asteroide del que tenga conocimiento el Minor Planet Center . Este sistema monitorea la base de datos de asteroides constantemente para evaluar el riesgo.Si es determinado como lo suficientemente feasible,añade el asteroide a una lista de posibles impactos computando variables de riesgo para el mismo ;basándose en la base de datos del Minor Planet Center que tiene las órbitas conocidas de cada asteroide. Si el conocimiento de la órbita es lo suficientemente preciso y se determina que no hay riesgo de impacto con la tierra, el sistema no hará la evaluación de riesgo . Las variables que se utilizaron ,que cada muestra tendrá son :

1. Object Name
2. Period Start
3. Period End
4. Cumulative Impact Probability
5. Asteroid velocity
6. Asteroid Magnitude
7. Asteroid Diameter (km)
8. Cumulative Palermo Scale
9. Maximum Palermo scale
10. Maximum Torino scale
11. Object classification
12. Epoch(TDB)
13. Orbit axis(AU)
14. Orbit eccentricity
15. Orbit inclination (deg)
16. Perihelion argument(deg)
17. Node longitude (deg)
18. Mean Anomaly(deg)
19. Perihelion distance(AU)
20. Aphelion distance (AU)
21. Orbital period(yr)
22. Minimum orbit intersection distance(AU)
23. Orbital reference
24. Asteroid magnitude

De estas características se extrajo una que nos servirá como la Y teórica llamada “Possible impacts” , es la variable que dice cuantos impactos posibles va a tener un asteroide con la tierra dentro del periodo fijado entre Period Start y Period End; además de basandose en las otras características de riesgo y de su órbita como tal .Debido a esto tendremos un problema de regresión supervisado ya que tendremos la variable de salida real para el entrenamiento en cada muestra.Queremos que cuando llegue una nueva muestra ;el sistema entrenado procese sus características de órbita y de riesgo para determinar cuántos posibles impactos tendrá el nuevo asteroide con la tierra dada toda su información.

2. Articulos cientificos

2.1. A transfer learning approach to space debris classification using observational light curve data

En esta primera publicación trata sobre "simulation training transfer" .Lo que quiere decir es que se ha comprobado que las redes neuronales convolucionales brindan unos muy buenos resultados sobre datos preprocesados en comparación a un dataset sin procesamiento.En este artículo el procesamiento se definió como el uso de un dataset de muestras de asteroides basadas en datos de curvas de luz (medidas discretas lumínicas sobre el objeto o muestra en cuestión y cómo evolucionan en magnitud en el tiempo) .El procesamiento dado fue que el dataset de entrada ,fue generado entrenando una red neuronal que refino las características de un dataset muy grande el cual fue generado por un simulador que ellos crearon .Luego los pesos optimizados de este modelo inicial ,que es una red neuronal convolucional, se usan para inicializar otra red neuronal convolucional separada que se entrenara con un dataset mas pequeño(ya que son muy difíciles de medir las propiedades lumínicas de los cuerpos celestes) de muestras de curvas de luz en objetos celestes reales; para darle algo así como un ajuste final.

Para testear el modelo se utilizó un dataset del MMT(Multi Channel Monitoring telescope)que se tuvo que balancear.Cabe aclarar que para este dataset y para el que fue simulado, debido a que había demasiadas muestras, se utilizó batches de tamaño muy grande .Finalmente como técnica de validación se uso cross validation con k=5 con una relación 80/20.En los resultados se compararon estas RNN con una red completamente conectada y con un SVM, dando como resultado que las redes convolucionales que se trataron en este trabajo dieron una precisión 20% mayor en comparación a los demas ,resultando en 80%,84%,90%,75% .(1)

2.2. An automated bolide detection pipeline for GOES GLM

Existen en los satellites GOES 16 y 17 un mapeador geoestacionario de relámpagos (GLM).Estos dispositivos se han encontrado ser eficaces para la detección de "bólidos" o asteroides luminiscentes cerca de la atmósfera de la tierra de poco tamaño ,entre 0.1 a 3 metros ,teniendo mayor capacidad que los detectores de luz que están en la tierra.El proyecto se desarrolla con un "pipeline" en el cual se creó primero un prototipo para demostrar la fiabilidad de la detección mediante GLM.Posteriormente se programaron algoritmos prototipo de clasificación con clustering jerárquico y luego sequencial para ser agendados para su ejecución en supercomputadoras de la nasa;los resultados son manualmente etiquetados para generar el dataset de aprendizaje supervisado debido a que aún no se cuenta con un detector lo suficientemente confiable ya que los relámpagos resultan, en muchos casos, ser falsos positivos .Para la preparación de las variables se utilizó la normalización de scikit-learn y se añadieron variables provenientes de las propiedades de los clusters anteriormente mencionados a las muestras.Según el documento debido a las características del modelo Random forest ,que puede exaltar características teniendo en cuenta una gran varianza,se escogió este por encima de los SVM .Para los hiperparametros se uso grid search con una validación cruzada de k=3 .El dataset de validación fue el 33% del dataset original con 67% para entrenamiento .Finalmente para cada satelite la precisión fue 45.9% y 41.2% respectivamente para GOES 16 y 17 ;esto fue suficiente para un etiquetado manual aunque se espera que cuando se haga automáticamente estas cifras aumenten .(2)

2.3. Solar-sail trajectory design for multiple near-Earth asteroid exploration based on deep neural networks

El problema de intercambio de órbitas entre N cuerpos celestes es un problema que se ataca mediante optimización .Este artículo propone identificar los tiempos de transferencia óptimos entre 2 orbitas mediante redes neuronales profundas para luego pasar a la secuencia óptima de transferencias entre N órbitas mediante Monte Carlo Tree Search(MCTS).En la red neuronal profunda se le alimenta mediante información orbital de los objetos en cuestión renderizados en un plano representado por ecuaciones predefinidas ;con tiempos de iniciación de la nave que orbitaba los cuerpos celestes en cuestión(la cual es una nave que navega con viento solar) ,para encontrar el tiempo óptimo de transferencia orbital.En la red neuronal se utiliza inicialmente un set de hasta 40000 objetos orbitales generados mediante un algoritmo que promete entrenar la red para que brinde los menores tiempos de transferencia posibles que luego se puedan evaluar en un set real .Estos 40000 objetos se dividieron en batches generalmente de a 200 para entrenar la red y se dividió todo el conjunto inicial haciendo bootstrapping con 90% para entrenamiento y 10% para validación con un resultado entre los dos de hasta 98.352%.Finalmente esta red entrenada se utilizó sobre otro set de muestras aparte de las 40000 iniciales para constatar los resultados ;que brindaron una precisión de hasta 98.934%.(3)

2.4. Modeling irregular small bodies gravity field via extreme learning machines and Bayesian optimization

En este artículo trata el tema de la navegación de cualquier nave alrededor de asteroides ;particularmente irregulares . Esto se debe a que al ser irregulares y relativamente pequeños es difícil calcular su campo gravitacional ;sin embargo saber esto es crucial tanto para navegar sobre ellos como para saber sus órbitas en relación a cualquier otra perturbación gravitacional de cualquier otro cuerpo cercano.Esto nos servirá para completar la información de su órbita y poder predecir posibles impactos aun mejor.La idea es, basado en un dataset , un calculador de la aceleración que un asteroide ejerce ,según su posición ,a una nave espacial en función de la distancia .Para este fin se usan técnicas de una rama llamada Extreme Machine Learning (es una técnica que trata de no especializar el aprendizaje sobre los parámetros de una neurona en particular,brindando más poder de generalización a un menor costo computacional que los modelos tradicionales y sin necesidad de intervención humana).Esta técnica es usada implementando una red de una sola capa embebida llamada Single Layer Feedforward Network (SLFN) la cual es muy útil para las relaciones no lineales entre las entradas y salidas . Para la optimización de hiperparametros y otras cualidades ,como número de neuronas por ejemplo,se utilizó optimización bayesiana .En cuanto al conjunto de datos , la base de datos original fue generada mediante los llamados modelos polyhedron ;y se partió aleatoriamente en 90% para entrenamiento y 10% para validación .En el entrenamiento para mejorar la eficiencia en tiempo ,se partió el set de entrenamiento en diferentes círculos concéntricos de órbitas que fueron entrenando el modelo .Luego para probar el modelo se evaluó sobre los asteroides conocidos 25143 Itokawa y el cometa 67/P Churyumov-Gerasimenko,dando en las medidas de error utilizadas (como absolute prediction error y las componentes gravitacionales en varios ejes ,entre otros) valores lo suficientemente precisos para concluir que es un proyecto feasible para instalar en una nave espacial calculando en tiempo real.(4)

3. Experimentos

Durante todo el proyecto se utilizó la validación cruzada con k=10 ; esto se pensó como la mejor solución para evitar el sobreajuste y que fuera de 10 como lo suficiente para el tamaño de la base de datos final .

Para la preparación de variables se utilizó dos bases de datos : la de las órbitas de los asteroides y la de posibles impactos .La base de datos de órbitas contiene todo lo relacionado con las órbitas de los asteroides ;la base de datos de impactos son dichos asteroides que representan un riesgo dentro en los próximos 100 años aproximadamente de impactar la tierra y contiene por consiguiente información derivada de la primera base de datos utilizada para calcular métricas de riesgo por cada asteroide.

De esta forma se concatenan ambas bases de datos para que cada muestra tenga su propia descripción tanto de su órbita como de su riesgo .Esto da como resultado una intersección de las bases de datos ,en la cual solo queda la información concatenada de un asteroide si este está en la base de datos de posibles impactos ;debido a que si no esta ,no presenta riesgo alguno con la tierra aunque se conozca su órbita .

En la preparación de los datos se eliminó Maximum torino scale debido a que sólo contiene ceros y no brinda mucha información al proyecto ,Adicionalmente se borró la columna redundante donde se repite el nombre del asteroide durante la concatenación de las dos bases de datos.Se tuvo tambien 3 muestras que no pudieron quedar en la base de datos final debido a que no tenían una correspondencia clara entre las bases de datos originales o simplemente en alguna no estaban.El resultado fue una base de datos de 680 muestras en total basada en: una base de datos de órbitas de asteroides con 15635 muestras y una de asteroides de riesgo con 683 muestras,ambas tomadas de www.kaggle.com de un mismo proyecto.

Para la característica Object classification se utilizó el 1-hot encoder donde solo queda un 1 en la etiqueta correspondiente de la base de datos;esto fue para no darle valores numéricos a las etiquetas que luego el modelo pueda mal interpretar ;tomando valores más grandes como más importantes y menores como menos .Las demás variables numéricas se normalizaron con la función Min Max Scaler por la misma razón .En todos los datos originales solo existía un valor de magnitud de un asteroide como NaN, el cual fue cambiado por la media de los valores de esa columna .Una vez preparado todo , solo basto quitar el nombre del asteroide como tal para solo tener números e inicializar el entrenamiento.

4. Regresion polinomial multiple

Para la regresión polinomial se utilizó el código de gradiente descendente , el método potencia Polinomio, el método que hace la regresión con el producto punto y el error calculado como el error cuadrático medio ECM. La aproximación que se tomó en este caso fue validación cruzada con un k=10 para hacer 10 particiones (9 de entrenamiento y 1 de validación) y diez iteraciones de la validación cruzada .Para encontrar los menores valores de error se anidaron ciclos para ir observando los cambios del error en relación a distintos parámetros .Se evaluó 1000 iteraciones en el gradiente descendente para ajustar los pesos ,con diferentes valores de aprendizaje eta que aumentan de 10 en 10 : 0.0001,0.001,0.01,0.1 .Luego para cada uno de estos valores eta , se ejecutan para cada modelo con grado 1,2,3,4,5 y 6.El error se midió partiendo del error cuadrático medio como se dijo ;por cada iteración de la validación cruzada nos brinda un ECM .Una vez tenemos todos los ECM de todas las iteraciones de la validación cruzada ,promediamos esos errores para obtener uno solo que corresponderá a un error único de la misma sobre cada valor de eta y de grado del polinomio(modelo)asi como sus intervalos de confianza.Los resultados son :

	ETA	
grado	0.0001	0.001
1	6979+/- 15208	6849 +/- 14175
2	6911 +/- 15006	6817 +/- 14150
3	6891 +/- 14896	6799 +/- 14149
4	6886 +/- 14826	6787 +/- 14153
5	6888 +/- 14777	6775 +/- 14154
6	6893 +/- 14743	6763 +/- 14152

	ETA	
grado	0.01	0.1
1	6369 +/- 13571	5673 +/- 11563
2	6257 +/- 13377	5606 +/- 11367
3	6218 +/- 13272	5572 +/- 11300
4	6207 +/- 13195	5561 +/- 11268
5	6207 +/- 13136	5557 +/- 11250
6	6209 +/- 13092	5557 +/- 11237

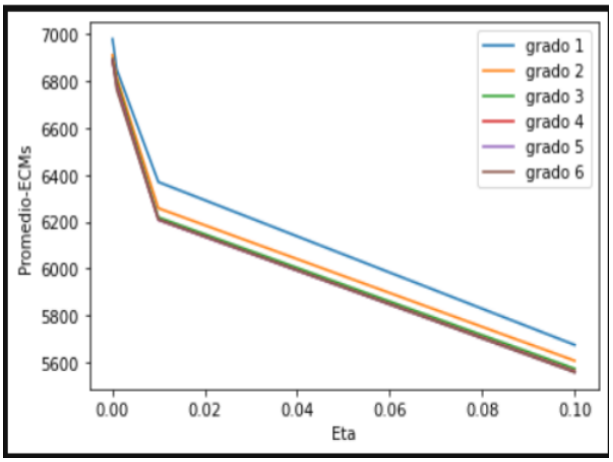

Fig. 1. Relacion eta versus promedio del error cuadratico medio

A pesar de variar los parámetros como los eta en un factor de 10 y complejizar el modelo en su grado ,los datos muestran que se alcanza un mínimo global fácilmente con el grado 3 y la tasa de aprendizaje en 0.1.Con los demás valores superiores se tiende a llegar al mismo error mínimo,solo que ligeramente menor.Como son valores promediados podemos decir con certeza que en ese promedio hay valores aún más bajos alcanzados.Con la mejor combinación de parámetros tenemos:

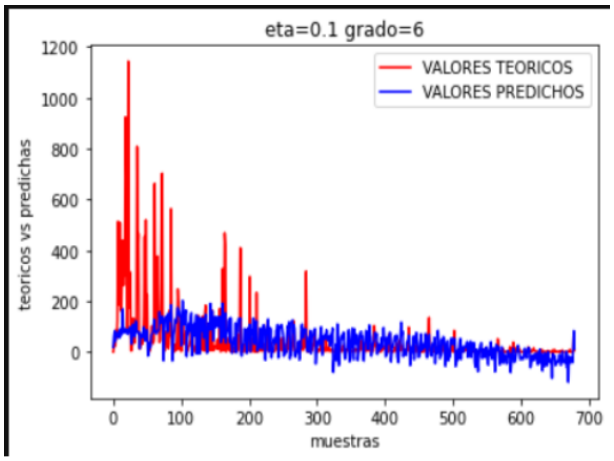

Fig. 2. Valores predichos versus reales

5. Ventana de parzen

Para el modelo ventana de parzen se ensayaron con diferentes hiperparametros de h los cuales no dieran error, división por cero por ejemplo, sin embargo los resultados no son muy prometedores el error al ser cuadrático tiende a volverse muy grande y las gráficas demuestran la diferencia entre lo predicho y lo real. A consecuencia no es un buen modelo para este tipo de base de datos.Ni siquiera la validación cruzada pudo hacer diferencia alguna .

h	ECM
1	361798544 +/- 93245754
10	361798544 +/- 93245754
100	361798544 +/- 93245754
1000	361798544 +/- 93245754

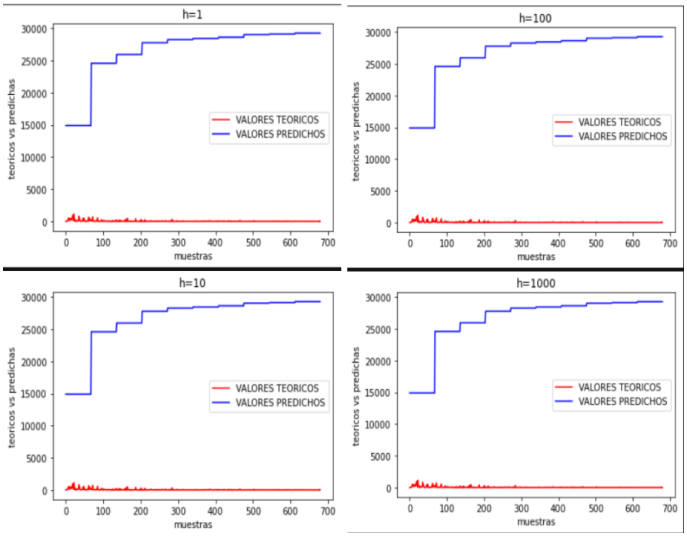

Fig. 3. Valores predichos versus reales

6. Red neuronal Feed forward

En las redes neuronales artificiales feedforward se utilizó validación Cruzada con k = 10 .Inicialmente se probaron 2 tipos diferentes de redes neuronales una con una capa oculta de 10 neuronas y la otra con dos capas ocultas de 20 y 10 neuronas respectivamente. La capa inicial de entrada tenía 29 neuronas correspondiente al número de características de cada muestra y finalmente una neurona para la capa de salida para predecir el número de posibles impactos. Para el entrenamiento se pretendió utilizar secuencias diferentes de epochs pero esto era computacionalmente excesivo para la máquina por lo que se dejó en 100 (número que se considera suficiente). En la compilación del modelo se utilizó un optimizador RMSprop el cual es utilizado para problemas de regresión, se dejaron todos sus valores por defecto cómo se recomienda excepto la tasa de aprendizaje que varió en los valores 0.001,0.01,0.1. Para el error se utilizó el error cuadrático medio de regresión el cual se promedió con los valores de cada iteración de la validación.Debido a que el modelo inicialmente comete errores muy grandes ,al ser inicializado aleatoriamente,se optó por graficar el menor error qué obtuviera durante esa validación cruzada en vez del promedio de todos.Para la compilación del modelo también se utilizó el error cuadrático medio como referencia para los ajustes de los pesos de la red además, en el entrenamiento, se le dijo qué se fuera entrenando con conjuntos de a 68 muestras; los cuáles son un décimo de las totales acorde a la validación Cruzada. Con una red de una capa oculta de 10 neuronas tenemos errores ECM de:

Tasa de aprendizaje		
0.001	0.01	0.1
926953+/- 1939833	886842 +/- 1618475	1055028 +/- 1774356

Debido a que el error, como son redes neuronales, inicialmente es muy grande con la inicializacion de pesos; el promedio tambien lo es ,por lo que se procedio a evaluar el error minimo alcanzado:

Tasa de aprendizaje		
0.001	0.01	0.1
36357	18931	20295

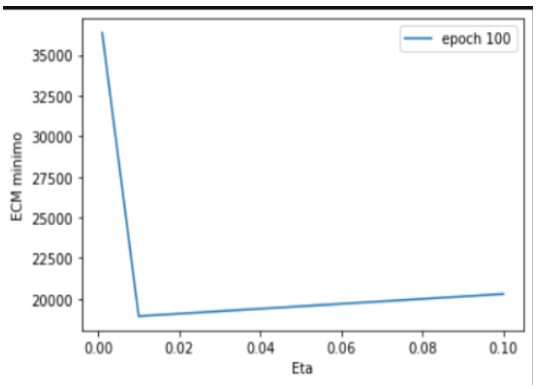

Fig. 4. Relacion entre el error y la tasa de aprendizaje

Con una red neuronal de 2 capas ocultas respectivamente de 20 y 10 neuronas:

Tasa de aprendizaje		
0.001	0.01	0.1
6843	13812	7535

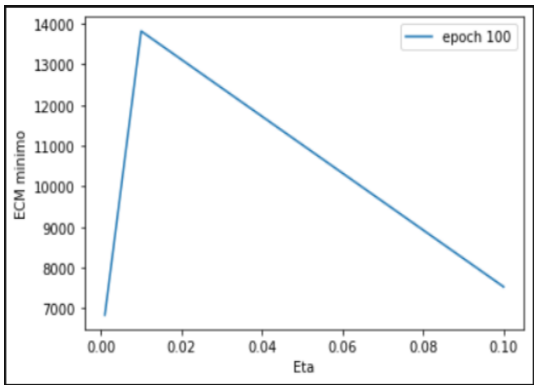

Fig. 5. Relacion entre el error y la tasa de aprendizaje

Cómo podemos observar el menor error dio con el modelo más complejo de dos capas ocultas con una tasa de aprendizaje de 0.001 donde el error cuadrático medio alcanzó un mínimo de 6843 al error mínimo qué habría dado en el modelo menos complejo con una tasa de aprendizaje de 0.01 con un mínimo de 18931.Con la mejor configuracion de parámetros , la del menor error tenemos:

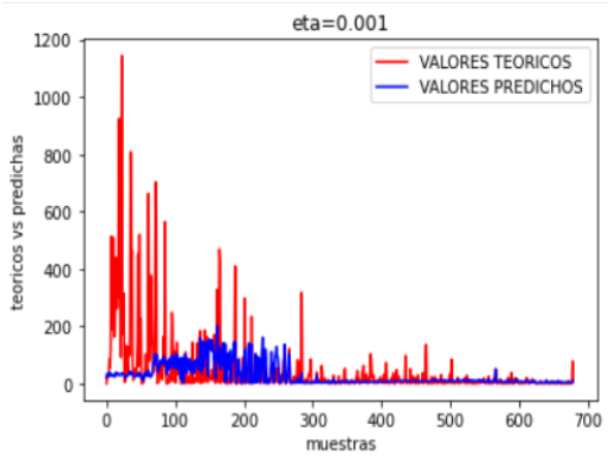

Fig. 6. Relacion entre el error y la tasa de aprendizaje

7. Árbol de decisión para regresión

Para evitar el sobreajuste se usó validación cruzada con k = 10.Para la configuración del árbol como tal ,se tomó el criterio del error cuadrático medio de friedmann y la estrategia para partir el nodo llamada best.En cuánto la partición del árbol se quiso usar una aproximación en la que se permitiese el mayor número de hojas con el menor número posible de muestras, o sea uno, y la mayor profundidad posible .El mínimo de muestras para partir un nodo se tomó como 2 .Para ir escogiendo las características se cambió el estado Random para ver si este conlleva a alguna mejoría en el error.Debido a la no limitación en el árbol ,luego se jugó con El parámetro qué dictamina la impureza mínima qué se ganaría al partir un nodo; finalmente, se ajustó los parámetros para hacer el algoritmo de podado con diferentes valores para ver si resulta va en un menor error. Utilizamos un algoritmo general que prueba entre muchos valores diferentes de los tres parámetros anteriormente expuestos para encontrar la mejor combinación de menor error .Se identificó buenos valores de random entre 0 y 20, buenos valores de decremento de impureza entre 0 y 2 así como para parámetro de podado (Cost complexity pruning). Los resultados más relevantes son:

ccp	random state	impureza	ECM
0.8	13	0.6	11319 +/- 18352
0	8	0.7	10881 +/- 18271
0	17	0.7	10407 +/- 18780
0.1	17	0.7	10393 +/- 18731
0.9	17	0.7	10389 +/- 18784

Con este ultimo de menor error tenemos:

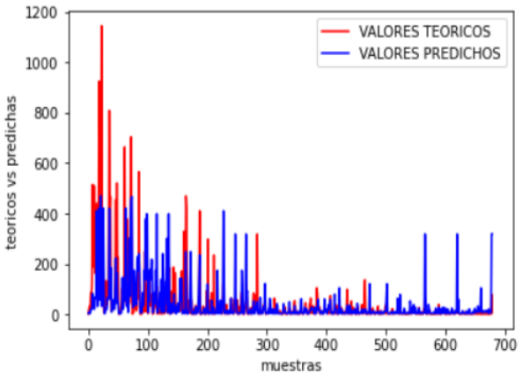

Fig. 7. Relacion entre el error y la tasa de aprendizaje

8. Maquina de Vector de soporte

Para la máquina de vector de soporte se utilizó la librería sklearn el SV Regresión .Se utilizó el kernel gaussiano para aumentar la dimensionalidad y los parámetros C para penalizar la función de costo por muestras que invadieran el espacio de los vectores de soporte(regularización) ;además el parámetro gamma como la inversa de la influencia de un vector de soporte determinado .Se itero en varios valores de los parámetros ,gama con 0.0001,0.001,0.01,0.1 y C con 1,10,100,100.Los mejores resultados fueron:

C	gamma	ECM
10	0.1	13394 +/- 29455
100	0.01	13318 +/- 29334
100	0.1	12545 +/- 27998

Con la muestra de menor error dio como resultado:

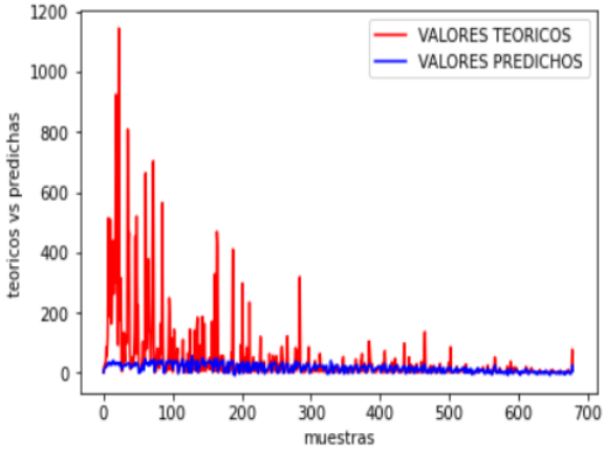

Fig. 8. Relacion entre el error y la tasa de aprendizaje

9. Seleccion de caracteristicas

La selección de características se efectuó y comparó en cada modelo ; en los mejores de menor ECM :regresión , redes neuronales y árboles . Se pensó utilizar una función wrapper para validar la selección que fuera una máquina de vector de soporte para regresión.Este modelo puede hacer un hiperplano que se ajusta a los datos en N dimensiones que puede ser regularizado con los parámetros C y gamma como parametro del kernel gausiano usado.La tecnica de seleccion usada fue la sequential forward selection debido a que se quiso comenzar con el conjunto vacío para ir probando desde lo mínimo de características , las mejores combinaciones.

9.1. Regresion polinomial multiple

Con la seleccion de caracterisaticas se seleccionaron d=10 de d=29 .La mejor combinación de variables con los menores errores fueron comparados con los mejores anteriores dando como resultado:

Grado=6

eta	ECM d=29	ECM d=10	mejora %
0.0001	6893 +/- 14743	6995 +/- 15216	-1.48
0.001	6763 +/- 14152	6869 +/- 14359	-1.57
0.01	6209 +/- 13092	6322 +/- 13438	-1.82
0.1	5557 +/- 11237	5668 +/- 11403	-2

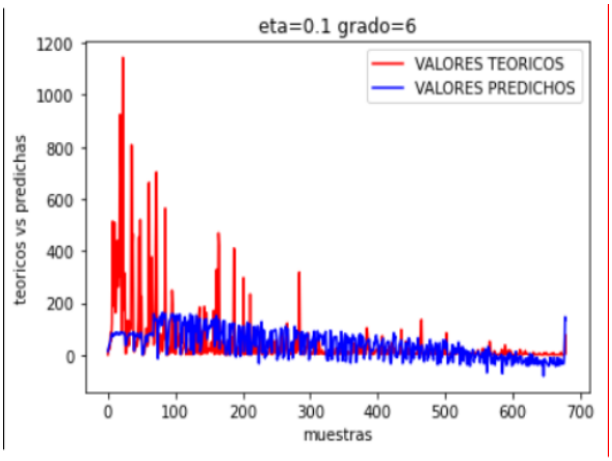

Fig. 9. Relacion entre el error y la tasa de aprendizaje

9.2. Red neuronal Feed forward

Con selección de características de d=29 a d=10 y la misma estructura de capas dio como resultado para una tasa de aprendizaje de 0.001 un error de validación mínimo de 9301.Debido a esto se procedió a cambiar las capas para acomodarse mejor a la nueva entrada;creando desde la entrada a la salida respectivamente capas de 10 ,5,y 1 neuronas.El resultado es que con una tasa de aprendizaje del 0.001 el error de validación es 914101 +/- 1985054 llegando a un mínimo de 6666.

tasa	ECM d=29	ECM d=10	mejora %
0.001	6843	6666	2.66

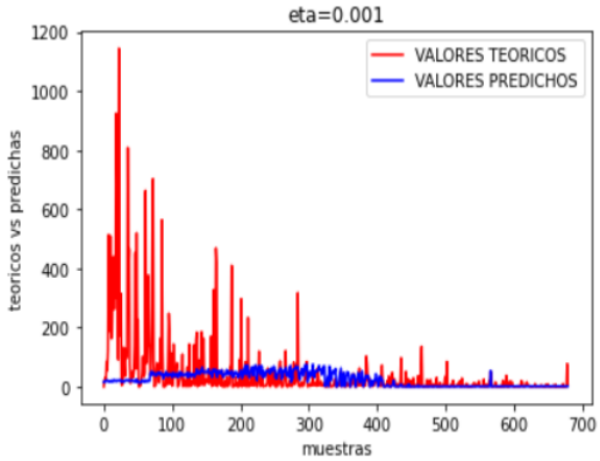

Fig. 10. Relacion entre el error y la tasa de aprendizaje

9.3. Árbol de decisión regresión

Con la seleccion de caracteristicas de d=29 a d=10 y con los mejores valores ,tenemos:

ccp	random state	impureza	ECM
0	13	0.6	9695 +/- 15362
0.9	13	0.6	9679 +/- 15287
0	13	0.7	9339 +/- 15059

En comparacion con los 3 mejores resultados que habiamos tenido con características d=29 comparado con d=10 tenemos:

ECM d=29	ECM d=10	mejora %
10407 +/- 18780	9695 +/- 15362	7.34
10393 +/- 18731	9679 +/- 15287	7.38
10389 +/- 18784	9339 +/- 15059	11.25

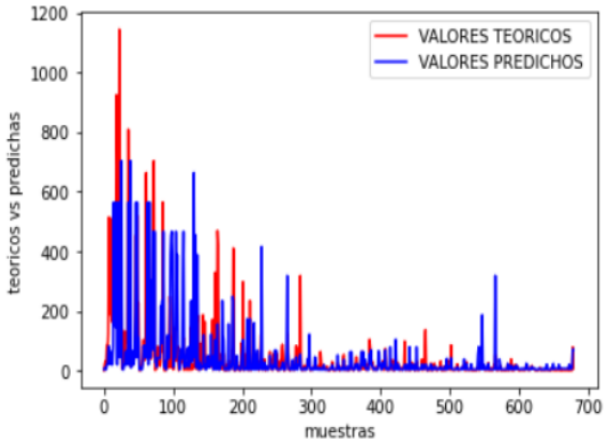

Fig. 11. Relacion entre el error y la tasa de aprendizaje

10. Extracción de caracteristicas

Para la extracción de características se evaluaron varios valores de los componentes ;se pretende ser algo más conservadores en el número extraído ya que en la selección de características se tomaron un tercio de las características con resultados medianamente buenos en algunos modelos ,que mejoran , excepto en el

de regresión multiple que empeoro.De todas maneras la reducción de la dimensionalidad manteniendo el error puede considerarse como ganancia en sí misma .Debido a las características del PCA en el que las variables deben ser lo menos correlacionadas posibles para evitar la redundancia de información; se utilizó en los data frames de las dos bases de datos de órbitas e impactos el comando para examinar sus matrices de correlación . Después de una revisión exhaustiva se observó que en las tablas se podría determinar una correlación significativa cuando esta fuera mayor igual a una magnitud de 0.1 (teniendo en cuenta que pueden ser correlaciones negativas también). En la base de datos de impactos se observó cada variable y se determinó en cada una lo anterior ;contando con cuantas otras se cumplia esta condicion excluyendo la variable en tratamiento en ese momento .Los resultados son : 3,6,2,5,4,6,6,6 .Se determinó mediante este análisis que si habian 4 variables de 8 en total que estaban muy correlacionadas con otras 6 ,1 variable correlacionada con otras 5 y una variable correlacionada con otras 4 ; existe información redundante en por lo menos 3 variables (como medida conservadora) por lo que indica que se podrían descartar potencialmente para la extracción de características.Así mismo tenemos en órbitas 3,6,5,3,0,0,0,5,6,6,8,4,8 .Tenemos 3 variables que no están muy correlacionadas con ninguna por lo que son esenciales ;sin embargo , tenemos 2 muy correlacionadas con otras 8 y 3 muy correlacionadas con otras 6 .De un total de 13 variables se estima de nuevo, siendo conservadores ,que al menos hay 2-3 variables que brindan información redundante y que pueden ser descartadas.

Finalmente observando las tablas en impacts se pudo corroborar lo anterior con variables como **maximun palermo scale** y **cumulative palermo scale** las cuales son medidas de riesgo relacionadas que pueden explicar lo anterior .Así mismo la **asteroid magnitud** y **asteroid diameter** son medidas de lo mismo pero medidas de maneras diferentes ,la primera como una medida astronómica dada a una unidad astronómica de distancia (AU) observada y la segunda puede ser tomada reflejando ondas al objeto o derivandose de la primera .En orbits lo anterior se puede evidenciar en el hecho de que son descripciones de la órbita misma ; por ejemplo el afelio y perihelio tienen alta correlación y estas mismas estan correlacionadas con el periodo orbital y la excentricidad de la orbita dando cabida a la redundancia de información anteriormente expuesta.

Debido a todo esto se determino que de 21 características se descartaran 4 para tomar en total 17 componentes .Las 8 características que no se tuvieron en cuenta son simplemente la variable de clasificación del objeto que se volvió en un one hot encoder con 8 valores posibles. Como consecuencia el numero total de componentes a tomar son 17 + 8 = 25 componentes.

10.1. Regresion polinomial multiple

Para la extracción de características no se pudo computar más del grado 1 del polinomio debido a que daba valores muy grandes para la máquina ,los mejores resultados fueron:

eta	vs.	ECM	Grado=1
0.0001	0.001	0.01	0.1
7096+/-15442	5971+/-13179	5558+/-10847	5382+/-10220

Comparandolo con los valores originales de caracteristicas d=29 a c=25 componentes tenemos:

Grado=1

eta	ECM d=29	ECM c=25	mejora %
0.0001	6893 +/- 14743	7096 +/- 15442	-2.95
0.001	6763 +/- 14152	5971 +/- 13179	13.3
0.01	6209 +/- 13092	5558 +/- 10847	11.71
0.1	5557 +/- 11237	5382 +/- 10220	3.25

Obteniendo con el menor error:

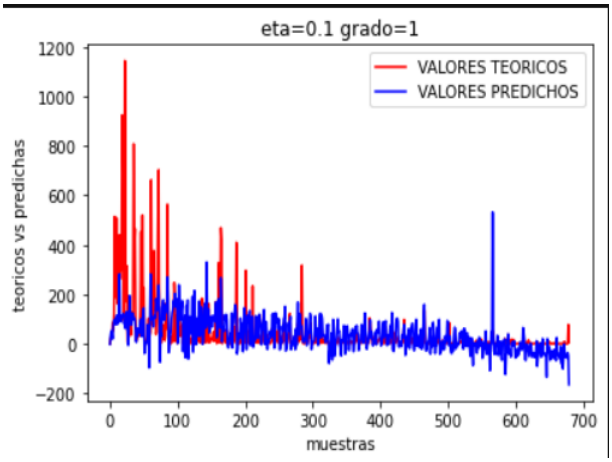

Fig. 12. Relacion entre el error y la tasa de aprendizaje

10.2. Red neuronal Feed forward

Debido a que volvimos a extraer varios componentes ,25 en total , se cambió la topología de la red de nuevo igual a la inicial; teniendo 25 entradas , una capa de entrada de 20 neuronas , una capa oculta de 10 y la de salida de una neurona .Lo mínimo alcanzado fue para 0.1 un error de validación: 1143089 +/- 1846803 ,además lo minimo dentro de este valor fue:

	eta=0.1
ECM minimo	6330

Dando una mejoría con respecto a el error mínimo,respecto a las características originales d=29 con respecto a c=25 componentes y una eta en este caso de 0.1 en comparación a la original de 0.001, igual a:

	d=29	c=25	mejora %
ECM minimo	6843	6330	8.1

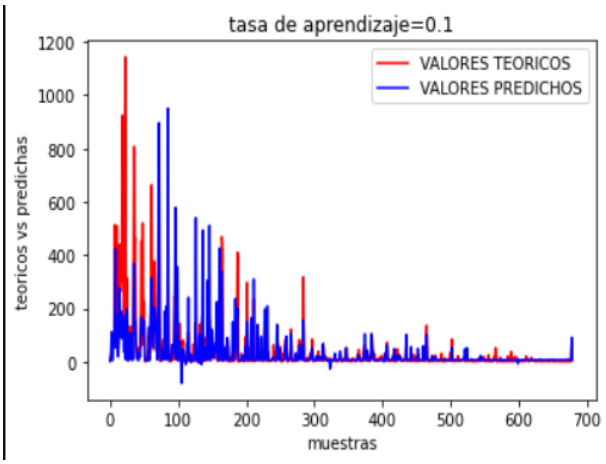

Fig. 13. Relacion entre el error y la tasa de aprendizaje

10.3. Árbol de decisión regresión

Para árboles se hizo unas iteraciones para encontrar la mejor combinación de parámetros para determinar la impureza ganada al partir un nodo , la profundidad y su podado así como el parámetro de selección de combinaciones de características que recibe un número entero .Todo esto se hizo después de haber seleccionado los 25 componentes ,los mejores resultados fueron:

ccp	random state	impureza	ECM
0.2	15	1.0	15996 +/- 23072
1.6	15	1.0	15988 +/- 23080
0.1	11	1.2	15971 +/- 22191
0.2	11	1.2	15952 +/- 22174

De d=29 caracteristicas a c=25 componentes extraídos tenemos :

ECM d=29	ECM c=25	mejora %
10881 +/- 18271	15996 +/- 23072	-47
10407 +/- 18780	15988 +/- 23080	-53.6
10393 +/- 18731	15971 +/- 22191	-53.6
10389 +/- 18784	15952 +/- 22174	-53.54

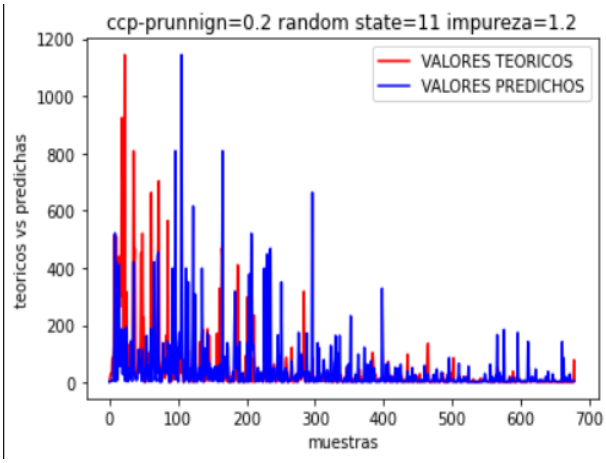

Fig. 14. Relacion entre el error y la tasa de aprendizaje

11. Discusión

1. Para comenzar cabe aclarar que es difícil encontrar bases de datos que expliquen el mismo problema a tratar en este artículo .Sin embargo se pueden encontrar muchos artículos relacionados que pretenden descubrir más información acerca de las diferentes propiedades de los asteroides como su órbita por ejemplo ;esto puede ser de gran ayuda para completar las bases de datos tratadas en en este trabajo para asi quizas llegar a mejores resultados .
2. Puede que la falta de datos sea un factor determinante en el aprendizaje ; una base de datos de decenas de miles podria ayudar mucho más ,especialmente a las redes neuronales artificiales que simplemente 680 muestras.Los modelos usados en los artículos investigados son bastante avanzados en comparación a los tratados aquí .Esto es particularmente cierto en las redes neuronales convolucionales las cuales tienen una gran ventaja sobre las máquinas de vectores de soporte de hasta 20% ;ya que en estos trabajos fueron tratadas además para alimentar otras redes neuronales convolucionales .Las redes neuronales profundas utilizadas en aquellos trabajos se aprovecharon de bases de datos generados artificialmente que ,en la validación con datos reales, lograron eficiencias del 98% en adelante ,debido a que también se lograron crear 40000 muestras.Finalmente la técnica de Extreme machine learning sobre las feed forward neural networks demostró sobrepasar todos los modelos de este trabajo .En este trabajo las redes neuronales solo fueron feedforward sin tratamiento previo mas que una escalada de los valores ,entre los valores máximos y mínimos .Los resultados aquí fueron los mejores con la topología más compleja de dos capas ocultas y hubo una mejora progresiva en la selección y extracción de características en sus valores mínimos ,tomando 10 características y 25 componentes respectivamente ,mejorando 2.66% y 8.1% con respecto al error mínimo original de la red .En los trabajos leídos no hubo un problema de regresión como tal para una probabilidad de impactos , o si lo hubo fue más calculando una distancia,tiempo o una aceleración orbital .El error de la regresión fue prometedor brindando el más bajo de todos los que se vieron ;que solo empeoro un 2% seleccionando 10 características pero perdiendo 19 ,lo cual es aún una ganancia en la reducción de dimensionalidad .Para la extraccion de caracteristicas mostro mejora del 3.25% respecto a la original lo cual es prometedor ,con 25 componentes y habiendo reducido también el grado del polinomio de 6 a 1 que garantiza una reducción en la complejidad significativa . Aunque en los artículos leídos nunca se habló de árboles de decisión ,en este trabajo el que se trató dio un error medianamente bueno aunque fue el tercero mejor de los modelos originales.Seleccionando 10 características el modelo mejoró significativamente en 11.25% aunque con la extracción de componentes a 25 empeoró en 53.24% lo cual no se esperaba por el análisis de correlación entre variables y teniendo en cuenta que siempre se buscó exhaustivamente la mejor combinación de parámetros de pureza podado y agrupación de características en el modelo ,este fue la desmejora más grande que hubo en todo el trabajo.El método de parzen resultó ser el peor modelo dando resultados de hasta millones;inclusive intentando con la búsqueda de mejor dimensión de la ventana; prediciendo valores disjuntos a los teóricos lo cual pudo evidenciar que tal vez sería mejor usar ventana de parzen para un problema de clasificación.El modelo de vector de soporte fue el escogido para la función wrapper ; como modelo en sí en los trabajos leídos, se pudo superar

su rendimiento, como ya se dijo, con redes neuronales convolucionales.En este trabajo en particular fue el cuarto mejor modelo adelante solo de ventana de parzen por lo que no se procedió a mejorarlo con extracción o selección de características.

3. En cuanto a la metodología de validación,en los trabajos a veces escogen bootstrapping como metodología ,especialmente para las redes neuronales que necesitan muchos datos de una sola vez para generalizar bien ; y se demostraba que así era suficiente porque los mismos datos habían sido generados previamente de una manera ya deseada en donde el sobreajuste no sería un problema;en esos casos dividían la base de datos en relaciones 70/30-80/20 lo que demostró ser suficiente con al menos un entrenamiento en batch debido al número mismo de muestras. En otros trabajos se optó por una validación cruzada de 5 por ejemplo ; sin embargo en este trabajo como tal se optó por siempre hacer una validación cruzada de 10 ya que se concluyó que sería suficiente para evitar el sobreajuste desde la raíz ,por así decirlo , debido a que la base de datos a trabajar fue relativamente pequeña con apenas 680 muestras y esto sería suficiente para cada modelo.

Acknowledgements. Parte de este trabajo fue apoyado por el profesor *Antonio Jesus TamayoHerrera -Ph. D. (c) en Ciencias de la Computación* del curso Simulacion de sistemas

(1) (2) (3)

References

[1] J. Allworth, L. Windrim, J. Bennett, and M. Bryson, "A transfer learning approach to space debris classification using observational light curve data," *Acta Astronautica*, vol. 181, pp. 301–315, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0094576521000588

[2] J. C. Smith, R. L. Morris, C. Rumpf, R. Longenbaugh, N. McCurdy, C. Henze, and J. Dotson, "An automated bolide detection pipeline for goes glm," *Icarus*, vol. 368, p. 114576, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0019103521002451

[3] Y. Song and S. Gong, "Solar-sail trajectory design for multiple near-earth asteroid exploration based on deep neural networks," *Aerospace Science and Technology*, vol. 91, pp. 28–40, 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1270963819301476

[4] R. Furfaro, R. Barocco, R. Linares, F. Topputo, V. Reddy, J. Simo, and L. Le Corre, "Modeling irregular small bodies gravity field via extreme learning machines and bayesian optimization," *Advances in Space Research*, vol. 67, no. 1, pp. 617–638, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0273117720304415