Informe Proyecto 2 entrega 7

Juan Luis Solórzano (carnet: 201598) Micaela Yataz (carnet: 18960)

2025-01-20

https://github.com/JusSolo/Mineria_Proyecto2.git

git: https://github.com/JusSolo/Mineria_Proyecto2.git

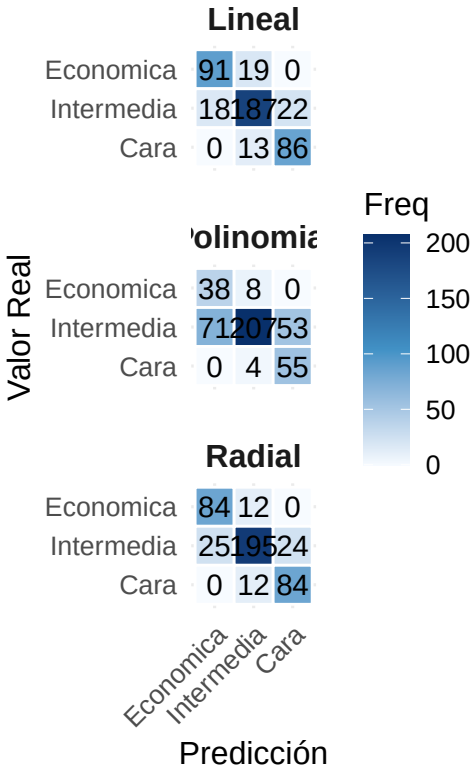
Introducción:

A lo largo del semestre hemos usado diferentes modelos de aprendizaje supervisado tanto en su version de clasificación como de regresión. Para este último informe del proyecto 2 se pretende probar máquina de vectores de soporte con diferentes topológias para clasificar los precios de las casas en categorías y otras redes para predecir el precio de las mismas. Por otro lado se desea comparar todos los modelos anteriores.

Modelo de Clasificación con redes Neuronales

Comparacion de primeros 3 modelos con distintos kernels

Matrices de Confusión por Modelo



Observando las matrices de confución se puede concluir que los 3 modelos se equivocan más clasificando la categoría de las casas intermedias. El modelo radial y lineal parecen ser los mejores, Siendo el radial más equilibrado y el lineal mejor en la clasificacion entre las casas econimicas e intermedias.

Table 1: Comparación de métricas macro-promediadas para los 3 modelos SVM usando datos de prueba

	Accuracy	Kappa	Sensitivity	Precision	F1
Linear	0.835	0.733	0.828	0.840	0.834
Radial	0.833	0.725	0.813	0.850	0.828
Polynomial	0.688	0.442	0.601	0.795	0.634

Table 2: Comparación de métricas macro-promediadas para los 3 modelos SVM usando datos de entrenamiento

	Accuracy	Kappa	Sensitivity	Precision	F1
Linear	0.890	0.822	0.882	0.895	0.888
Radial	0.929	0.885	0.922	0.935	0.928
Polynomial	0.800	0.651	0.734	0.900	0.776

Comparado ambas tablas el único modelo que parece sobre ajustado es el polynomial, pues es que tiene mayores diferencias en las métricas de desempeño con los datos de prueba y entrenamiento.

Modelo ajustado

Entre los 3 modelos anteriores se decidió ajustar el modelo radial, pues es más flexible que el lineal y fue un poco pero que el lineal. Posiblemente al ajustarlo su desempeño mejore.

```
## Mejores hiperparámetros:
```

```
##    sigma   C
## 21 0.001 100
```

Table 3: Métricas macro-promediadas del modelo SVM Radial tuneado (entrenamiento vs prueba)

	Accuracy	Kappa	Sensitivity	Precision	F1
Tuned Radial (Train)	0.902	0.842	0.892	0.911	0.901
Tuned Radial (Test)	0.823	0.713	0.813	0.831	0.821

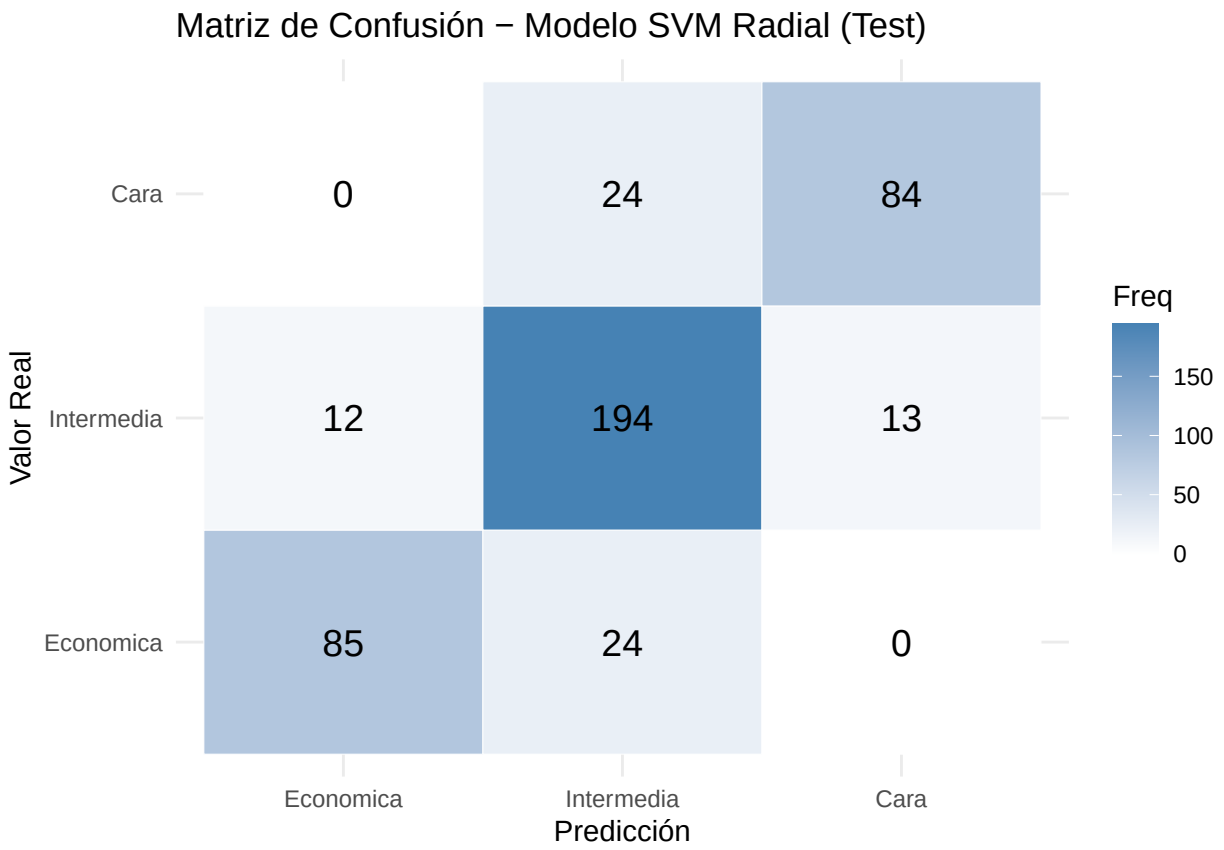
El modelo tuneado tiene claramente sobre ajuste y es peor, por lo que vamos a volver a ujustarlo pero separando los datos de entrenamiento en 2 entrenamiento y validación para evitar sobre ajuste.

```
## Mejores hiperparámetros:
```

```
##   sigma   C
## 3 0.001 100
```

Table 4: Métricas del modelo SVM Radial tuneado (con train)

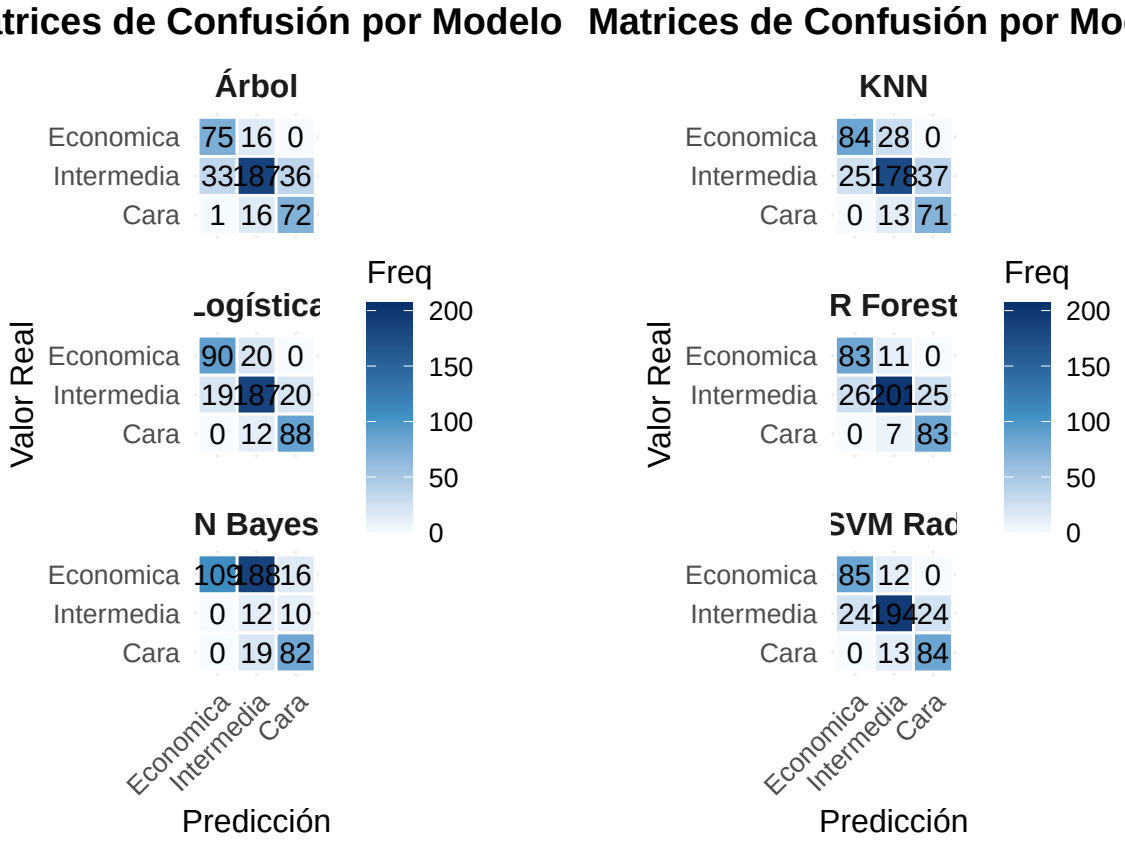
	Accuracy	Kappa	Sensitivity	Precision	F1
Validación (subtrain)	0.896	0.830	0.883	0.907	0.894
Prueba (test final)	0.833	0.726	0.814	0.848	0.829



Viendo las métricas y las matrices de confusión y las métricas el modelo tuneado radial parece ser un poco mejor que el lineal, pero por muy poco.

Comparacion de todos los modelos de clasificacion usados a lo largo del proyecto

Matrices de confución



Observado las matrices de confucion se puede notar que naive bayes el el único modelo que no clasifica para nada bien. Los demás parecen mas o menos igual de buenos clasificando, observar las matrices de confucion es un poco engorroso.

Tablas comparando Espacio en memoría usada en ejecucion y tiempo de ejecucion

```
##
## Attaching package: 'pryr'

## The following objects are masked from 'package:purrr':
##
##     compose, partial

## The following object is masked from 'package:dplyr':
##
##     where

## Warning in microbenchmark(predict(modelo, newdata = datos), times = 10, : less
## accurate nanosecond times to avoid potential integer overflows
```

Table 5: Comparación de complejidad espacial y temporal de modelos ordenados por tiempo de ejecucion

Modelo	Tiempo (ms)	Memoria (MB)	Precisión
Árbol de Decisión	0.638	0.08	0.766
Regresión Logística	2.158	1.65	0.837
SVM Radial	10.154	0.48	0.833
KNN	16.017	0.78	0.768
Random Forest	16.213	6.58	0.842
Naive Bayes	56.801	0.02	0.466

Table 6: Comparación de complejidad espacial y temporal de modelos ordenados por espacio en memoria

Modelo	Tiempo (ms)	Memoria (MB)	Precisión
Naive Bayes	56.801	0.02	0.466
Árbol de Decisión	0.638	0.08	0.766
SVM Radial	10.154	0.48	0.833
KNN	16.017	0.78	0.768
Regresión Logística	2.158	1.65	0.837
Random Forest	16.213	6.58	0.842

Table 7: Comparación de modelos ordenados por precisión

Modelo	Tiempo (ms)	Memoria (MB)	Precisión
Random Forest	16.213	6.58	0.842
Regresión Logística	2.158	1.65	0.837
SVM Radial	10.154	0.48	0.833
KNN	16.017	0.78	0.768
Árbol de Decisión	0.638	0.08	0.766
Naive Bayes	56.801	0.02	0.466

Arriba se puede observar la misma tabla pero ordenada según cada una de sus columnas. Como el SVN es el único modelo que aparece en el top 3 en las 3 tablas podemos considerar que en términos generales es el mejor modelo. Si se toma en cuenta el tiempo de ejecución y la presicion el mejor modelo es la regresión logística, pero es el segundo que ocupa más memoria. Para estos datos de la manera que fueron tratados el peor modelo es Naive Bayes, esto puede deberse a que algunas de las variables no son independientes (ir a ver nalisis exploratorio informe 1). En conclucion exceptuando Naive Bayes todos los modelos funcionan. Si se necesitara un modelo muy rapido y que no ocupe mucha memorio (en la ejecucion) el algoritmo a elegir sería El Árbo de desicion. Si se buscara la presicion a toda costa seria el Random Forest el elegido. Pero si se desea un eqilibrio entre los 3 el modelo a elegir sería le regresión logística (si no importa mucho el espacio en memoría) y un SVM radial (si no importa que el tiempo de ejecucion sea mas lento). En conclusión no hay un modelo que sea el mejor en todo. El que se elija usar debería ser el que se pueda ejecutar en el tiempo y espeacio que se requiere con la presicion más alta. Para tomar esa desicion podria ser util usar alguna funcion de costo que pondere las metricas y elegir el modelo que lo minimice y cumpla con las restricciones.

Modelo de Regresion con redes Neuronales

Para los modelos de regresión, se seleccionó la variable (SalePrice) como variable respuesta, ya que es el que representa el precio de venta de las casas y es el objetivo principal.

Se entrenan dos modelos de redes neuronales con distintas topoloias y parámetros, entrenados con validacion cruzada.

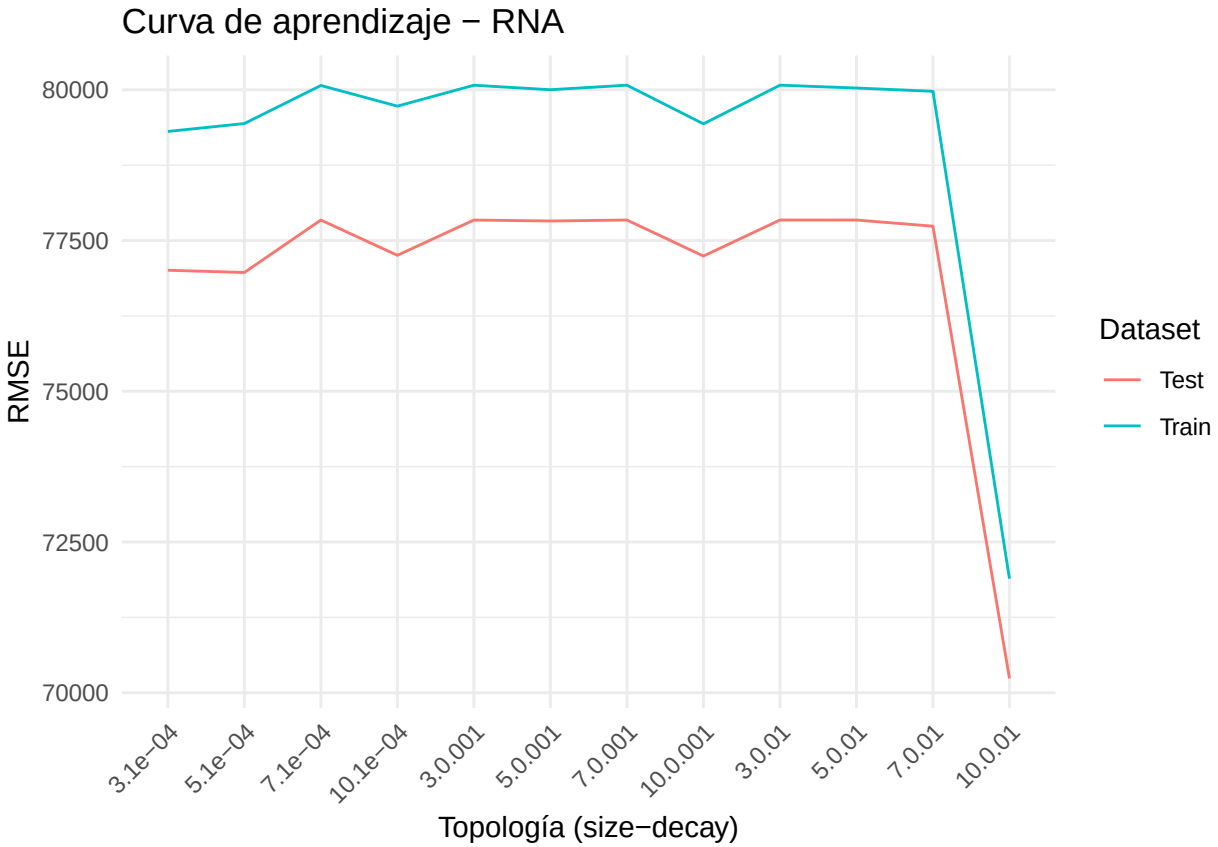
```
##   Modelo     RMSE
## 1   RNA1 63260.48
## 2   RNA2 76804.70
```

Al comparar los dos modelos de redes neuronales, se observa que el modelo RNA1 tiene un RMSE de 63260.48 a comparación con RNA2 que tiene un RMSE de 76804.70, indicando que RNA1 es el que mejor rendimiento tiene de estos dos modelos.

La verificacion de sobreajusto se realizó una gráfica de curva de aprendizaje. En el que no hay sobreajuste.

```
##   size decay
## 6    5  0.01
```

```
##    size decay RMSE_Train RMSE_Test
## 1     3 1e-04   79309.22  77007.26
## 2     5 1e-04   79440.27  76970.26
## 3     7 1e-04   80071.36  77840.05
## 4    10 1e-04   79729.37  77255.11
## 5     3 1e-03   80075.37  77840.08
## 6     5 1e-03   80001.31  77823.90
## 7     7 1e-03   80076.61  77840.15
## 8    10 1e-03   79434.99  77242.87
## 9     3 1e-02   80076.61  77840.16
## 10    5 1e-02   80029.71  77839.79
## 11    7 1e-02   79974.60  77738.19
## 12   10 1e-02   71891.52  70238.63
```



Se realizó sobreajuste al modelo RNA utilizando la validacion cruzada, mostrando que la mejor combinacion fue:

el de tamaño 5 y decay 0.1

```
## Warning in nominalTrainWorkflow(x = x, y = y, wts = weights, info = trainInfo,
## : There were missing values in resampled performance measures.
```

```
##   size decay
## 3    5   0.1
```

Este modelo final mejoró levemente el RMSE respecto a los modelos iniciales, sin sobreajuste, ya que los errores se mantuvieron cercanos, tanto para entrenamiento como validación.

Comparacion de todos los modelos de regresion usados a lo largo del proyecto.

Se comparó el rendimiento del mejor modelo de RNA con modelos realizados anteriormente, como regresión lineal, KNN y arbol, Naive Bayes. Se muestrar a continuacion los resultados.

```
##             Modelo     RMSE
## 1 Regresión Lineal 36745.35
## 2            Árbol 46340.50
## 3              KNN 39930.13
## 4      Naive Bayes       NA
## 5             RNA1 63260.48
## 6             RNA2 76804.70
## 7        RNA Tuned 60096.00
```

Entre todos los modelos de regresion entrenados, el mejor modelo fue de Regresion lineal ya que tiene menos error con un RMSE de 36745.35 secundado por KNN DE 39930.13. Notese que ninguno de los modelos de RNA supero a la regresion lineal

```
## Confusion Matrix and Statistics
##
##             Reference
## Prediction   Economica Intermedia Cara
##   Economica          0          0    0
##   Intermedia       108        189   23
##   Cara               1         30   85
##
## Overall Statistics
##
##                Accuracy : 0.6284
##                  95% CI : (0.5812, 0.6739)
##     No Information Rate : 0.5023
##     P-Value [Acc > NIR] : 7.646e-08
##
##                   Kappa : 0.3429
##
##  Mcnemar's Test P-Value : < 2.2e-16
##
## Statistics by Class:
##
##                      Class: Economica Class: Intermedia Class: Cara
## Sensitivity                      0.00            0.8630      0.7870
## Specificity                      1.00            0.3963      0.9055
## Pos Pred Value                    NaN            0.5906      0.7328
## Neg Pred Value                  0.75             0.7414      0.9281
```

```
## Prevalence                       0.25            0.5023      0.2477
## Detection Rate                   0.00            0.4335      0.1950
## Detection Prevalence             0.00            0.7339      0.2661
## Balanced Accuracy                0.50            0.6297      0.8463
```

```
## Confusion Matrix and Statistics
##
##             Reference
## Prediction   Economica Intermedia Cara
##   Economica         84         11    0
##   Intermedia        25        194   20
##   Cara               0         14   88
##
## Overall Statistics
##
##                Accuracy : 0.8394
##                  95% CI : (0.8016, 0.8727)
##     No Information Rate : 0.5023
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.7378
##
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: Economica Class: Intermedia Class: Cara
## Sensitivity                    0.7706            0.8858      0.8148
## Specificity                    0.9664            0.7926      0.9573
## Pos Pred Value                 0.8842            0.8117      0.8627
## Neg Pred Value                 0.9267            0.8731      0.9401
## Prevalence                     0.2500            0.5023      0.2477
## Detection Rate                 0.1927            0.4450      0.2018
## Detection Prevalence           0.2179            0.5482      0.2339
## Balanced Accuracy              0.8685            0.8392      0.8861
```

```
##          Modelo  Accuracy
## 1           RNA 0.6284404
## 2 Random Forest 0.8394495
## 3           KNN 0.7683486
```

En clasificación, el modelo Random Forest obtuvo mejor precisión, por el Acuaracy (0.8394495) segudio por KNN, el modelo RNA logro un buen desempeño, aunque no el mejor,

```
##    user  system elapsed
##   0.002   0.000   0.002
```

```
##    user  system elapsed
##   0.002   0.000   0.002
```

Dada que las redes neuronales no fueron muy precisas en regresión, el modelo tuneado obtuvo mejor significativa respecto a RNA iniciales. Pero la regresión lineal sigue siendo el modelo mas preciso.

Conclusiones

El mejor modelo de regresion fue la regesion lineal, superando al modelo RNA tuneado. En clasificación, random Forest fue el mas preciso. Las redes neuronales fueron útiles, pero no superaron a los medelos mas simples Los modelos de regresión lineal y Random Forest son los recomendados para este conjunto de datos.