Universidad del Valle de Guatemala
Minería de Datos
Sección 10

Hoja de Trabajo 6

Brian Carrillo - 21108
Diego Alonzo - 20172
Carlos López - 21666

Guatemala, 11 de abril del 2024

1. **Cree una variable dicotómica por cada una de las categorías de la variable respuesta categórica que creó en hojas anteriores. Debería tener 3 variables dicotómicas (valores 0 y 1) una que diga si la vivienda es cara o no, media o no, económica o no.**

	PoolArea	MiscVal	MoSold	YrSold	Cara	Economica	Intermedia
1	0	0	2	2008	1	0	0
2	0	0	5	2007	0	0	1
3	0	0	9	2008	1	0	0
4	0	0	2	2006	0	0	1
5	0	0	12	2008	1	0	0
6	0	700	10	2009	0	0	1

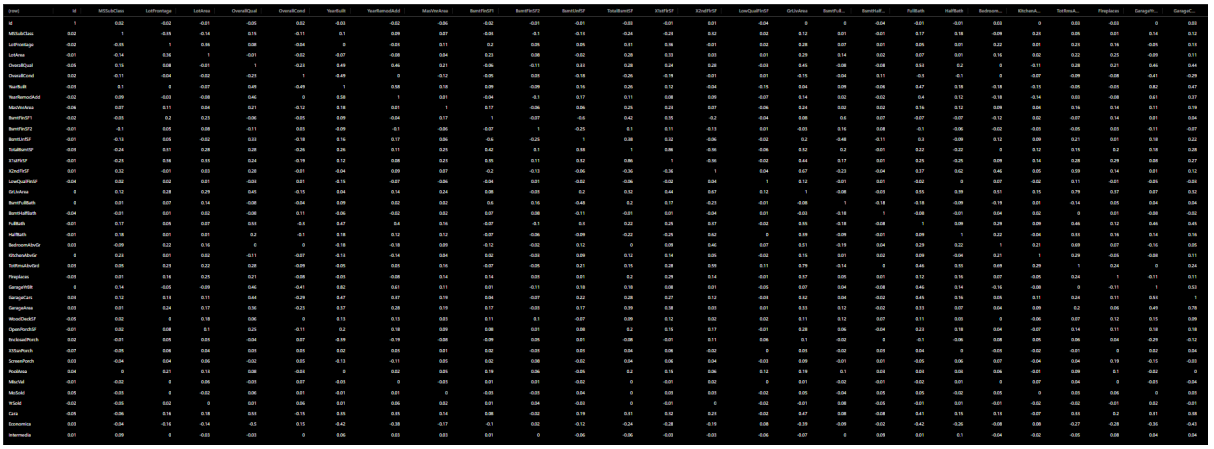
De manera que como se observa se generaron 3 columnas, una para la cara, otra para la económica y otra para la intermedia e indica a cuál pertenece cada una.

2. **Use los mismos conjuntos de entrenamiento y prueba que utilizó en las hojas anteriores.** Se utilizaron los mismos conjuntos de hojas anteriores como se observa en la generación de las variables dicotómicas del inciso anterior.

3. **Elabore un modelo de regresión logística para conocer si una vivienda es cara o no, utilizando el conjunto de entrenamiento y explique los resultados a los que llega. El experimento debe ser reproducible por lo que debe fijar que los conjuntos de entrenamiento y prueba sean los mismos siempre que se ejecute el código. Use la validación cruzada.**
4. **Analice el modelo. Determine si hay multicolinealidad en las variables, y cuáles son las que aportan al modelo, por su valor de significación. Haga un análisis de correlación de las variables del modelo y especifique si el modelo se adapta bien a los datos.**

Coefficients:

	Estimate	Std. Error	z value	Pr(>|z|)	
(Intercept)	-6.537e+01	1.733e+02	-0.377	0.706094	
MSSubClass	-1.175e-02	4.061e-03	-2.894	0.003806	**
LotFrontage	-7.548e-03	6.149e-03	-1.227	0.219654	
LotArea	3.645e-05	2.091e-05	1.743	0.081267	.
OverallQual	1.219e+00	1.740e-01	7.006	2.45e-12	***
OverallCond	3.822e-01	1.685e-01	2.268	0.023319	*
YearBuilt	4.890e-02	1.425e-02	3.431	0.000602	***
YearRemodAdd	3.840e-02	1.210e-02	3.173	0.001510	**
MasVnrArea	-2.884e-04	7.897e-04	-0.365	0.714988	
BsmtFinSF1	1.659e-03	8.546e-04	1.942	0.052180	.
BsmtFinSF2	2.009e-03	1.054e-03	1.906	0.056640	.
BsmtUnfSF	1.341e-03	7.987e-04	1.679	0.093155	.
X1stFlrSF	3.427e-03	9.257e-04	3.702	0.000214	***
X2ndFlrSF	4.472e-03	7.745e-04	5.774	7.73e-09	***
LowQualFinSF	3.412e-03	4.070e-03	0.838	0.401846	
BsmtFullBath	9.123e-01	3.424e-01	2.664	0.007722	**
BsmtHalfBath	-3.856e-01	5.883e-01	-0.655	0.512214	
FullBath	-2.184e-01	4.083e-01	-0.535	0.592717	
HalfBath	-4.438e-01	3.827e-01	-1.160	0.246230	
BedroomAbvGr	7.966e-02	2.544e-01	0.313	0.754206	
KitchenAbvGr	-1.588e+00	1.113e+00	-1.427	0.153607	
TotRmsAbvGrd	3.565e-02	1.690e-01	0.211	0.832899	
Fireplaces	2.979e-01	2.191e-01	1.359	0.174018	
GarageYrBlt	-3.304e-02	1.434e-02	-2.304	0.021246	*
GarageCars	7.554e-01	4.870e-01	1.551	0.120912	
GarageArea	1.987e-03	1.406e-03	1.413	0.157662	
WoodDeckSF	6.602e-05	9.793e-04	0.067	0.946249	
OpenPorchSF	9.622e-04	1.935e-03	0.497	0.618969	
EnclosedPorch	4.417e-03	2.326e-03	1.899	0.057609	.
X3SsnPorch	9.731e-04	2.849e-03	0.342	0.732633	
ScreenPorch	2.179e-03	2.028e-03	1.074	0.282655	
PoolArea	-1.456e-02	4.399e-03	-3.310	0.000934	***
MiscVal	-2.139e-04	3.892e-04	-0.550	0.582567	
MoSold	-9.996e-04	4.257e-02	-0.023	0.981264	
YrSold	-3.073e-02	8.624e-02	-0.356	0.721586	



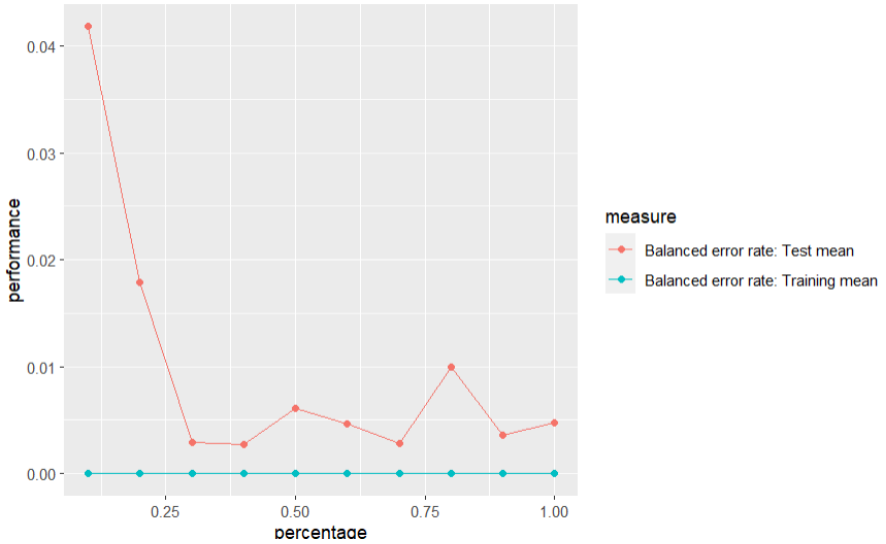
Al obtener la matriz de correlación se observó que existe correlación positiva entre las variables YearBuilt y GarageYrBlt, BsmtFinSF1 y X1stFlrSF, GrLivArea y TotRmsAbvGrd. Así mismo, se observa que las variables que realmente aportan al modelo son MSSubClass, OverallQual, OverallCond, YearBuilt, YearRemodAdd, X1stFlrSF, X2ndFlrSF, BsmtFullBath, GarageYrBlt, PoolArea. El AIC es de 471 por lo que no mejora significativamente, así mismo el BIC es de 624.

5. **Utilice el modelo con el conjunto de prueba y determine la eficiencia del algoritmo para clasificar.**

```
Confusion Matrix and Statistics

          Reference
Prediction   0   1
         0 232  27
         1   7  57

               Accuracy : 0.8947
                 95% CI : (0.856, 0.926)
    No Information Rate : 0.7399
    P-Value [Acc > NIR] : 3.498e-12

                  Kappa : 0.7036

 Mcnemar's Test P-Value : 0.00112

            Sensitivity : 0.9707
            Specificity : 0.6786
         Pos Pred Value : 0.8958
         Neg Pred Value : 0.8906
             Prevalence : 0.7399
         Detection Rate : 0.7183
   Detection Prevalence : 0.8019
      Balanced Accuracy : 0.8246

       'Positive' Class : 0
```

Utilizando la matriz de confusión se observan los siguientes resultados se obtuvieron 232 verdaderos positivos, es decir casas que realmente no eran caras, lo que equivale a un 97.07% de efectividad en la detección de no caras, y 57 verdaderos negativos lo que es 67.86% de casas que realmente son caras.

6. **Explique si hay sobreajuste (overfitting) o no (recuerde usar para esto los errores del conjunto de prueba y de entrenamiento). Muestre las curvas de aprendizaje usando los errores de los conjuntos de entrenamiento y prueba.**



El modelo tiene un alto error al principio cuando tiene pocos datos para aprender, lo cual es esperado. A medida que se añaden más datos de entrenamiento, la tasa de error en los datos de prueba disminuye, lo que indica una mejor generalización. Hacia el final de la gráfica, donde se usa el 100% de los datos de entrenamiento, las líneas de entrenamiento y prueba parecen converger. Esto es una buena señal de que no hay un sobreajuste significativo.
El rendimiento en los datos de entrenamiento es bastante bueno y estable. El modelo se beneficia de tener más datos de entrenamiento para mejorar la precisión en los datos de prueba.

7. **Haga un tuneo del modelo para determinar los mejores parámetros, recuerde que los modelos de regresión logística se pueden regularizar como los de regresión lineal.**

```
Coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept)  -1.479e+02  2.387e+01  -6.193 5.89e-10 ***
MSSubClass   -1.419e-02  4.124e-03  -3.440 0.000581 ***
OverallQual   1.408e+00  2.133e-01   6.601 4.09e-11 ***
OverallCond   5.458e-01  2.093e-01   2.608 0.009104 **
YearBuilt     3.718e-02  1.396e-02   2.664 0.007732 **
YearRemodAdd  3.839e-02  1.456e-02   2.638 0.008348 **
X1stFlrSF     7.237e-03  7.994e-04   9.054  < 2e-16 ***
X2ndFlrSF     5.649e-03  5.868e-04   9.627  < 2e-16 ***
BsmtFullBath  1.508e+00  2.887e-01   5.226 1.74e-07 ***
GarageYrBlt  -1.323e-02  1.479e-02  -0.894 0.371095
PoolArea     -2.710e-01  6.772e+04   0.000 0.999997
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 834.03  on 750  degrees of freedom
Residual deviance: 342.63  on 740  degrees of freedom
AIC: 364.63

Number of Fisher Scoring iterations: 7

[1] 415.4632
```

El AIC es de 365 por lo que no mejora significativamente, así mismo el BIC es de 415. Hubo mejora en ambos criterios respecto al análisis anterior.

8. **Haga un análisis de la eficiencia del algoritmo usando una matriz de confusión. Tenga en cuenta la efectividad, donde el algoritmo se equivocó más, donde se equivocó menos y la**

importancia que tienen los errores, el tiempo y la memoria consumida. Para esto último puede usar "profvis" si trabaja con R y "cProfile" en Python.

```
Confusion Matrix and Statistics

          Reference
Prediction   0   1
         0 216  28
         1  14  65

               Accuracy : 0.87
                 95% CI : (0.8283, 0.9047)
    No Information Rate : 0.7121
    P-Value [Acc > NIR] : 1.184e-11

                  Kappa : 0.668

 Mcnemar's Test P-Value : 0.04486

            Sensitivity : 0.9391
            Specificity : 0.6989
         Pos Pred Value : 0.8852
         Neg Pred Value : 0.8228
             Prevalence : 0.7121
         Detection Rate : 0.6687
   Detection Prevalence : 0.7554
      Balanced Accuracy : 0.8190

       'Positive' Class : 0
```

Podemos observar que en cuanto a sensibilidad y especificidad, no existe un cambio significativo respecto a la matriz de confusión obtenida anteriormente.

```
> pm1$sampling.time
[1] 0.04
> pm2$sampling.time
[1] 2.14
> pm3$sampling.time
[1] 1.22
```

```
> sum(pm1$by.total$mem.total)
[1] 6.3
> sum(pm2$by.total$mem.total)
[1] 23153.7
> sum(pm3$by.total$mem.total)
[1] 12642.7
```

9. **Determine cual de todos los modelos es mejor, puede usar AIC y BIC para esto, además de los parámetros de la matriz de confusión y los del profiler.**

Los valores de AIC y BIC para el modelo simple y el modelo obtenido tras la validación cruzada, son similares. Por otra parte, dichos valores mejoran significativamente (especialmente para BIC) con el modelo cv tuneado (coeficientes significativos). Los valores de los modelos cv poseen mejores valores de sensibilidad y especificidad. Respecto al tiempo de ejecución en ms, el tiempo más bajo los posee el modelo simple, seguido del modelo cv tuneado. Finalmente, el uso de memoria en MB es más bajo para el primer modelos, seguido del tercer modelo nuevamente. Es posible concluir que el mejor modelo es el modelo cv con los coeficientes significativos.

10. **Haga un modelo de árbol de decisión, uno de Random Forest y uno de Naive Bayes usando la misma variable respuesta y los mismos predictores que el mejor de los modelos de Regresión Logística.**
11. **Compare la eficiencia de los 3 modelos que creó en el punto anterior y el mejor de los de regresión logística ¿Cuál se demoró más en procesar?¿Cuál se equivocó más?¿Cuál se equivocó menos?¿por qué?**

    Comparando los modelos realizados, el mejor modelo es el realizado con regresión logística, puesto que posee un valor más alto de precisión en cuanto a las predicciones realizadas. Computacionalmente, el modelo más complejo es Random Forest. El modelo que se equivocó más fue Naive Bayes, ya que tiende a ser muy sensible a variables con multicolinealidad.