Informe Proyecto 2 entrega 3

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

2025-01-20

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

1. Elabore un modelo de regresión usando bayes ingenuo (naive bayes), el conjunto de entrenamiento y la variable respuesta SalesPrice. Prediga con el modelo y explique los resultados a los que llega. Asegúrese que los conjuntos de entrenamiento y prueba sean los mismos de las hojas anteriores para que los modelos sean comparables.

En esta ocasión como naive Bayes así lo permite se tomaran todas las variables incluso las culitativas.

```
y<- datos$SalePrice
set.seed(123)
trainI<- createDataPartition(y, p=0.7, list=FALSE)

train<-datos[trainI, ]
test<-datos[-trainI, ]

trainC<-datosC[trainI, ]
testC<-datosC[-trainI, ]
```

```
modelo <- naiveBayes(SalePrice ~ ., data = train)
```

2. Analice los resultados del modelo de regresión usando bayes ingenuo. ¿Qué tan bien le fue prediciendo? Utilice las métricas correctas.

```
## RMSE test: 44719.19
## MAE test: 30229.49
## RMSLE test: 0.2477994
## R^2 test: 0.6699454
## RMSE train: 43160.73
## MAE train: 19555.73
## RMSLE train: 0.1966725
## R^2 train: 0.7094866
```

El R^2 es de 0.6699 no esta muy cercano a 1, pero tampoco es terrible. El RMSLE es de 0.25 lo que nos indica que el modelo tiene margen de mejora.

3. Compare los resultados con el modelo de regresión lineal y el árbol de regresión que hizo en las entregas pasadas. ¿Cuál funcionó mejor?

```
##                Modelo     RMSE      MAE        R2
## 1         Naïve Bayes 44719.19 30229.49 0.6699454
## 2  Regresión Lineal  37983.42 22724.28 0.7618854
## 3 Árbol de Regresión 46340.50 30161.08 0.6455791
```

En la tabla comparativa con cualquiera de las tres métricas se puede concluir que el mejor modelo es el lineal, el segundo mejor es el Naive Bayes y el peor es el árbol de regresión.

4. Haga un modelo de clasificación, use la variable categórica que hizo con el precio de las casas (barata, media y cara) como variable respuesta.

```
train$precio_categoria<-cut(train$SalePrice,
                            breaks = c(0, 129975, 214000, 400000, Inf),
                            labels = c("Economica", "Intermedia", "Cara", "Lujo"),
                            include.lowest = TRUE)

test$precio_categoria<-cut(test$SalePrice,
                            breaks = c(0, 129975, 214000, 400000, Inf),
                            labels = c("Economica", "Intermedia", "Cara", "Lujo"),
                            include.lowest = TRUE)


#modelo NB para clasificacion

modeloClas<-naiveBayes(precio_categoria~.- SalePrice, data=train)
```

5. Utilice los modelos con el conjunto de prueba y determine la eficiencia del algoritmo para predecir y clasificar.

```
#prediccion del modelo
pred_nb_clas<-predict(modeloClas, newdata=test[, !names(test) %in% c("salePrice", "PriceCateoria") ])
```

6. Haga un análisis de la eficiencia del modelo de clasificación 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.

```r
#evvaluar modelo
confusionMatrix(pred_nb_clas, test$precio_categoria)
```

```
## Confusion Matrix and Statistics
##
##             Reference
## Prediction   Economica Intermedia Cara Lujo
##   Economica        109        115    3    0
##   Intermedia         0         18    5    0
##   Cara               0         85   72    0
##   Lujo               0          1   19    9
##
## Overall Statistics
##
##                Accuracy : 0.4771
##                  95% CI : (0.4293, 0.5251)
##     No Information Rate : 0.5023
##     P-Value [Acc > NIR] : 0.8647
##
##                   Kappa : 0.3121
##
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: Economica Class: Intermedia Class: Cara Class: Lujo
## Sensitivity                    1.0000           0.08219      0.7273     1.00000
## Specificity                    0.6391           0.97696      0.7478     0.95316
## Pos Pred Value                 0.4802           0.78261      0.4586     0.31034
## Neg Pred Value                 1.0000           0.51332      0.9032     1.00000
## Prevalence                     0.2500           0.50229      0.2271     0.02064
## Detection Rate                 0.2500           0.04128      0.1651     0.02064
## Detection Prevalence           0.5206           0.05275      0.3601     0.06651
## Balanced Accuracy              0.8196           0.52958      0.7375     0.97658
```

Según la matriz de confusion, el moodelo mosto una efectividad en las categorias económicas y lujo con 109 y 9 predicciones correctas, respectivamente. Con efectviad moderada es el modelo es la categoria cara con 84 predicicones correctas. La categoria con menor efectividad es la de intermedio.

Con lo que el modelo en general cuenta con precision del 59.86%,valor de Accuracy, por lo que se puede conciderar un rendimiento moderado. Con Kapa de 0.4545 que es la concordancia entre el valor real y el valor de prediccion.

Por estadistica por clase el modelo identifica correctamente todas las casas económicas, asi como las casas de lujo, segida de la categoria Cara que identifico en parte mayoría las casas caras y por último la categoria intermedia, que fue la que tuvo menos aciertos en clasificar las casas intermedias

#7. Analice el modelo. ¿Cree que pueda estar sobreajustado?

```r
#evvaluar modelo
pred_nb_train<-predict(modeloClas, newdata=train[, !names(train) %in% c("salePrice", "PriceCateoria") ])

confusionMatrix(pred_nb_train, train$precio_categoria)
```

```
## Confusion Matrix and Statistics
##
##             Reference
```

```
## Prediction   Economica Intermedia Cara Lujo
##   Economica        251        256    9    0
##   Intermedia         3         42    1    0
##   Cara               2        215  176    0
##   Lujo               0          1   49   19
##
## Overall Statistics
##
##                Accuracy : 0.4766
##                  95% CI : (0.4456, 0.5077)
##     No Information Rate : 0.502
##     P-Value [Acc > NIR] : 0.9512
##
##                   Kappa : 0.3132
##
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: Economica Class: Intermedia Class: Cara Class: Lujo
## Sensitivity                    0.9805           0.08171      0.7489     1.00000
## Specificity                    0.6549           0.99216      0.7250     0.95025
## Pos Pred Value                 0.4864           0.91304      0.4478     0.27536
## Neg Pred Value                 0.9902           0.51738      0.9065     1.00000
## Prevalence                     0.2500           0.50195      0.2295     0.01855
## Detection Rate                 0.2451           0.04102      0.1719     0.01855
## Detection Prevalence           0.5039           0.04492      0.3838     0.06738
## Balanced Accuracy              0.8177           0.53693      0.7370     0.97512
```

```
confusionMatrix(pred_nb_clas, test$precio_categoria)
```

```
## Confusion Matrix and Statistics
##
##             Reference
## Prediction   Economica Intermedia Cara Lujo
##   Economica        109        115    3    0
##   Intermedia         0         18    5    0
##   Cara               0         85   72    0
##   Lujo               0          1   19    9
##
## Overall Statistics
##
##                Accuracy : 0.4771
##                  95% CI : (0.4293, 0.5251)
##     No Information Rate : 0.5023
##     P-Value [Acc > NIR] : 0.8647
##
##                   Kappa : 0.3121
##
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: Economica Class: Intermedia Class: Cara Class: Lujo
## Sensitivity                    1.0000           0.08219      0.7273     1.00000
```

```
## Specificity                 0.6391          0.97696      0.7478      0.95316
## Pos Pred Value              0.4802          0.78261      0.4586      0.31034
## Neg Pred Value              1.0000          0.51332      0.9032      1.00000
## Prevalence                  0.2500          0.50229      0.2271      0.02064
## Detection Rate              0.2500          0.04128      0.1651      0.02064
## Detection Prevalence        0.5206          0.05275      0.3601      0.06651
## Balanced Accuracy           0.8196          0.52958      0.7375      0.97658
```

El modelo de clasificacion Naive Bayes no se muestra signos significativos de sobreajuste. La presicion y el valor de Kappa son similares en los conjuntos de test y prueba, Ademas de que muestra dificultades similares al clasificar las categorias cara e intermedias, por lo que el modelo tiene rendimiento moderado y sin sobreajuste

#8. Haga un modelo usando validación cruzada, compare los resultados de este con los del modelo anterior. ¿Cuál funcionó mejor?

```
## NAs después de imputación: 0
```

Ambos modelos tienen rendimiento menor que el 30% por lo que no necesariamente Naive Bayes el mas adecuado para este problema para la clasificacion. La validacion cruzada no mejoró el rendimiento.

#9. Tanto para los modelos de regresión como de clasificación, pruebe con varios valores de los hiperparámetros, use el mejor modelo del tuneo, ¿Mejoraron los modelos? Explique

#10. Compare la eficiencia del algoritmo con el resultado obtenido con el árbol de decisión (el de clasificación) y el modelo de random forest que hizo en la hoja pasada. ¿Cuál es mejor para predecir? ¿Cuál se demoró más en procesar?