Análisis Predictivo

Final

Juan Ignacio Scorza

Profesores:

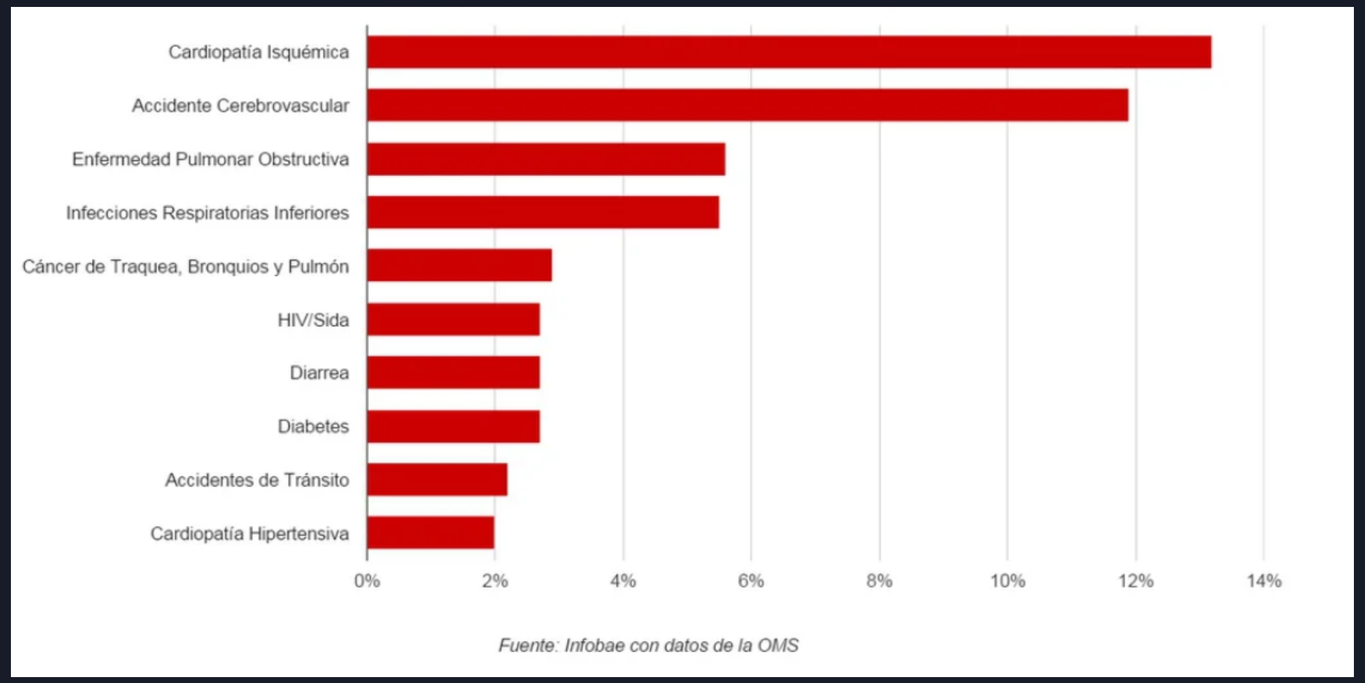
Ezequiel Martín Eliano Sombory

Leonardo Andrés Caravaggio

Francisco Valentini

2023 1Q

PRESENTACIÓN CASO DE NEGOCIO



Presión Arterial alta

EDA

1. Observaciones generales
2. Nulls/Missings
3. Target y Correlaciones
4. Creacion de Variables/ Encoding

EDA

1. Observaciones generales
   - Creación del dataset ⟶ 

   - Alcance y limitaciones
   - Dimensiones: (4983, 27)
   - Procesamiento
     - Selección
     - Rename
     - Replace

EDA

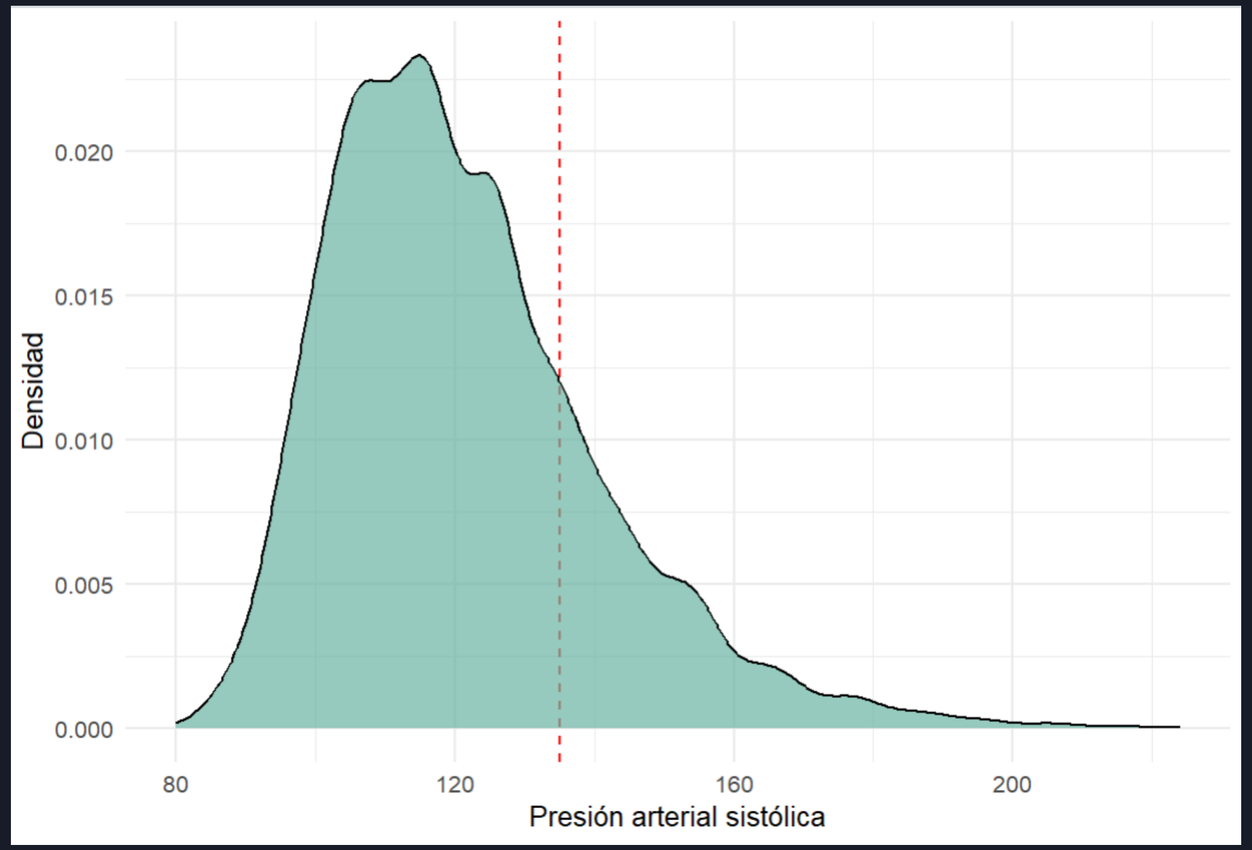
2 Nulls / Missings

seqn	0
age	0
race_ethnicity	0
gender	0
education_child	3735
education_adult	1249
household_income	179
country_birth	0
energy_intake	0
DR1DRSTZ	0
sodium_prep	0
sodium_table	0
water_intake	0
alcohol_intake	0
energy_intake_dr2	0
sugar_intake	0
sodium_table_dr2	0
alcohol_intake_dr2	0
water_intake_dr2	0
weight	33
systolic_bp	0
diastolic_bp	0
total_percent_fat	2277
health_insurance	0
monthly_income	244
alcohol_ever	1155
alcohol_past_12mo	1551

- Education_gral
- household_income ---> media
- monthly_income ---> media
- weight --> media por edad
- total_percent_fat --> regresion x var relacionadas
- alcohol --> ceros.

EDA

3 Target y correlaciones

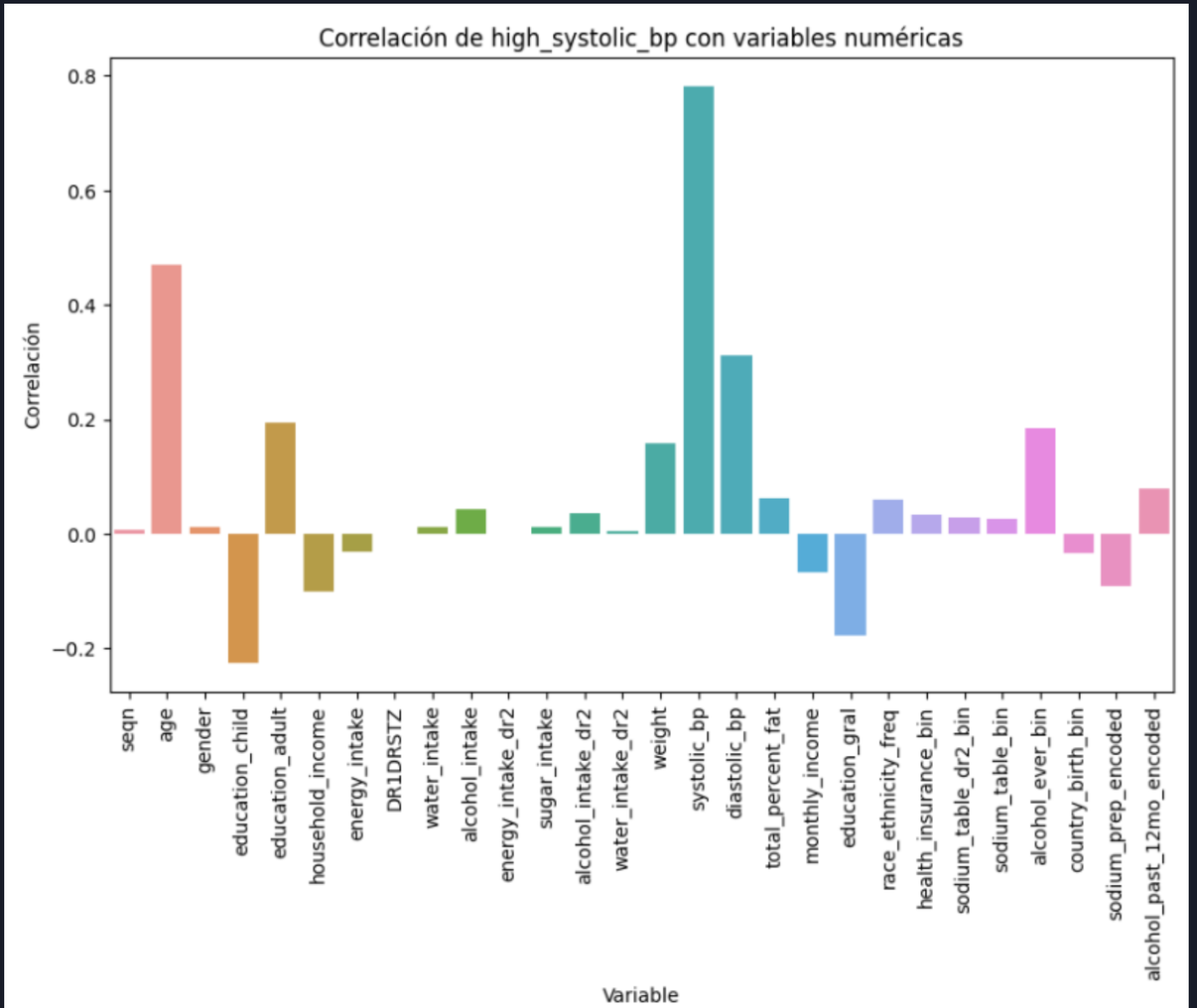



```
Número de filas con systolic_bp > 135: 1045
Número de filas con systolic_bp < 135: 3938
```

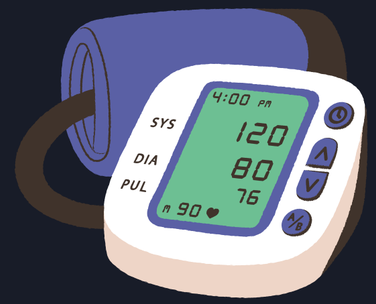
Se crea high_systolic_bp binaria

EDA

3 Target y Correlaciones



(correlación de la variable target con el resto de variables)



EDA

4 Creación de Variables/Encoding

Variables Categóricas

- race_ethnicity
- country_birth
- sodium_prep
- sodium_table
- sodium_table_dr2
- health_insurance
- alcohol_ever
- alcohol_past_12mo

Categórica  Numérica

EDA

4 Creación de Variables/Encoding

Frequency Encoding race_ethnicity

Variables binarias

- country_birth → US = 1 Other= 0
- sodium_table, sodium_table_dr2, health_insurance, alcohol_ever → Yes = 1 No = 0

Ordinal Encoding sodium_prep y alcohol_past_12mo

Modelos Predictivos

1. Particiones Utilizadas.
2. Métodos de Ajuste.
3. Modelos Utilizados.
4. Mejores Modelos.

Modelos Predictivos

1 Particiones Utilizadas.

```python
from sklearn.model_selection import train_test_split

model_num_columns = df.select_dtypes(include=['float64', 'int64']).columns.to_list()
model_num_columns.remove('diastolic_bp')
model_num_columns.remove('systolic_bp')
model_num_columns.remove('high_systolic_bp')

X = df[model_num_columns]
Y = df['high_systolic_bp']
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size = 0.20, random_state = 1704)
```

Test Size Utilizados

- 0.10
- 0.15
- 0.20
- 0.25
- 0.30

Modelos Predictivos

2 Métodos de Ajuste

Blackbox optimization

- from ***sklearn.model_selection*** import **GridSearchCV**
- from **skopt** import **BayesSearchCV**

Score

- Accuracy
- Recall

Para calcular la exhaustividad (recall) usaremos la siguiente fórmula:

$$recall = \frac{TP}{TP + FN}$$

		predicción	
		0	1
realidad	0	TN	FP
	1	FN	TP

Exhaustividad (recall)

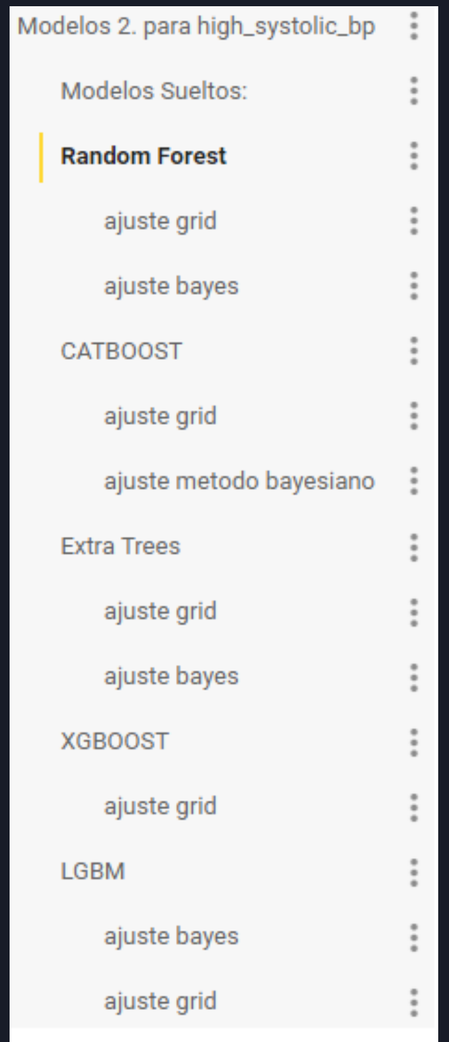
Modelos Predictivos

2 Métodos de Ajuste

```python
from skopt import BayesSearchCV
from catboost import CatBoostClassifier
from sklearn.metrics import confusion_matrix, recall_score
from sklearn.model_selection import train_test_split

# Definir los hiperparámetros a ajustar
param_grid = {
    'iterations': (100, 1000),
    'depth': (3, 10),
    'learning_rate': (0.01, 0.1, 'log-uniform')
}

# Crear el clasificador CatBoost
catboost = CatBoostClassifier(loss_function='Logloss', eval_metric='Accuracy', random_seed=1704)

# Crear el objeto BayesSearchCV
bayes_search = BayesSearchCV(catboost, param_grid, scoring='recall', cv=5, n_jobs=-1, refit=True)

# Ajustar el modelo con los datos de entrenamiento
bayes_search.fit(X_train, Y_train)

# Obtener las predicciones en el conjunto de prueba
y_pred = bayes_search.predict(X_test)

# Calcular la matriz de confusión
confusion = confusion_matrix(Y_test, y_pred)

# Calcular el recall del modelo
recall = recall_score(Y_test, y_pred)

# Calcular la accuracy del modelo
accuracy = accuracy_score(Y_test, y_pred)

# Imprimir la matriz de confusión y el recall
print("Matriz de Confusión:")
print(confusion)
print("Recall del modelo:", recall)
print("Accuracy del modelo:", accuracy)
print("best_params: ",bayes_search.best_params_)
```

Modelos Predictivos

3 Modelos Utilizados



Top 3 modelos

1. LGBM // bayes recall
2. RandomForest // bayes recall
3. Extra Trees // bayes recall

Modelos Predictivos

3 Modelos Utilizados

3. Extra Trees Regressor

from **skopt** import **BayesSearchCV**

Recall: 37,5%

Accuracy 82,6%

Best Params:

max_depth: 10

max_features: 1.0

min_samples_leaf: 1

min_samples_split: 10

n_estimators: 100

```
Matriz de Confusión:
[[749  48]
 [125  75]]
Accuracy del modelo: 0.8264794383149449
Recall del modelo: 0.375
```

Modelos Predictivos

3 Modelos Utilizados

2. RandomForest Regressor

from **skopt** import **BayesSearchCV**

Recall: 38,5%

Accuracy 82%

Best Params:

max_depth: 19

min_samples_leaf: 1

min_samples_split: 10

n_estimators: 10

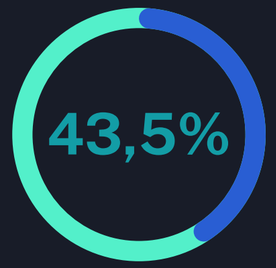
```
Matriz de confusión:
[[740  57]
 [123  77]]
Precisión: 0.819458375125376
```

Modelos Predictivos

3 Modelos Utilizados

1. LGBM

from **skopt** import **BayesSearchCV**

Recall: 

Accuracy 

Best Params:

colsample_bytree: 0.926055022505385

learning_rate: 0.29999999999999993

max_depth: 7

min_child_samples: 21

min_child_weight: 9

n_estimators: 179

num_leaves: 25

reg_alpha: 0

reg_lambda: 0

subsample: 0.8844228649546503

```
Matriz de Confusión:
[[706  91]
 [113  87]]
Accuracy del modelo: 0.7953861584754263
Recall del modelo: 0.435
```

Muchas Gracias