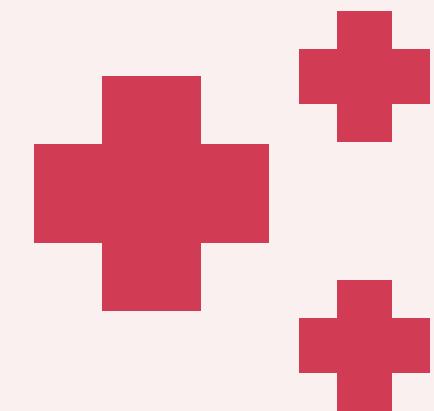


# Modelos predictivos para la identificación de ataques cardíacos

Lander Combarro Exposito

20 noviembre 2024

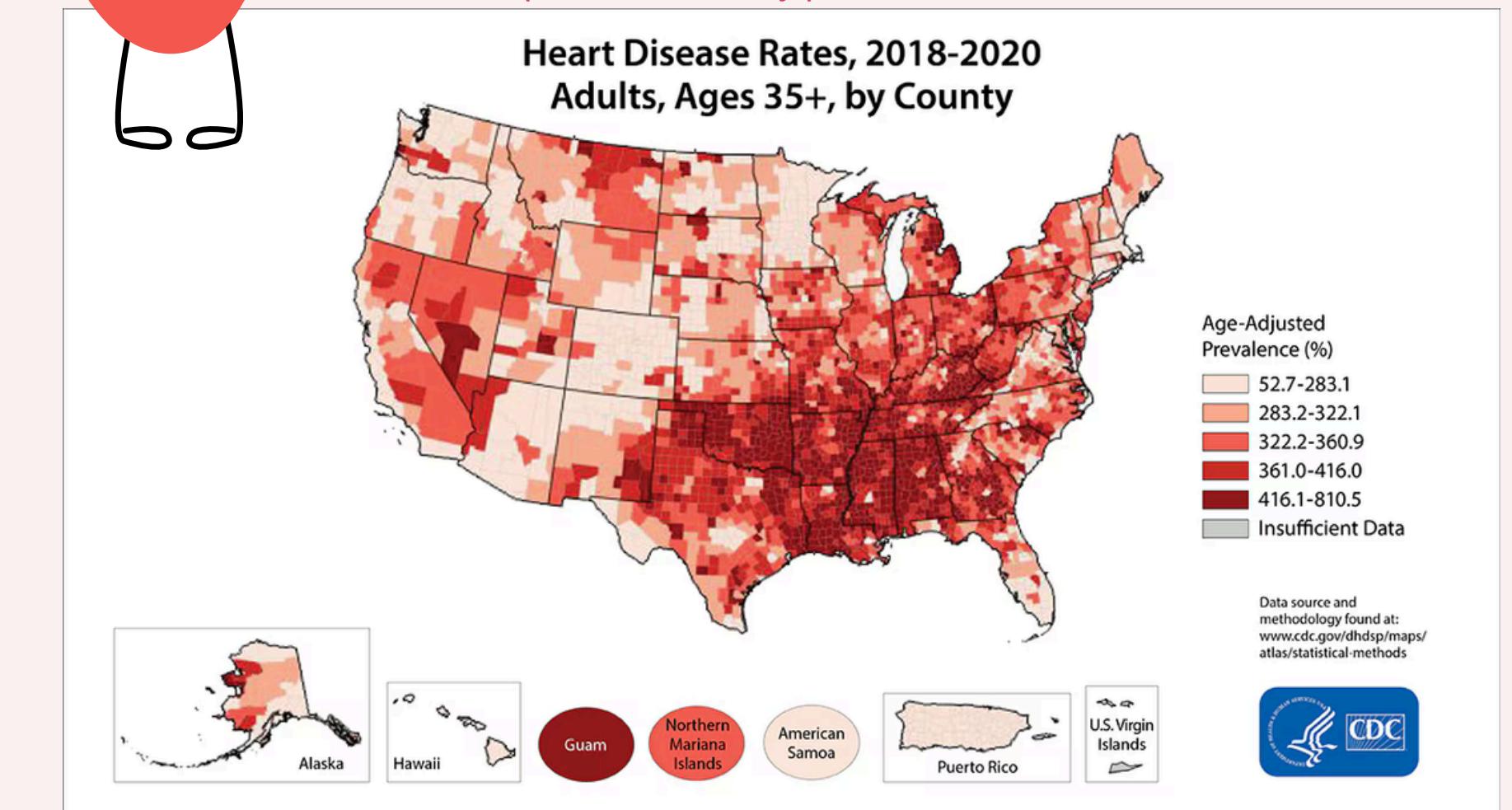


# Enfermedades cardíacas en EEUU

- Principal causa de muerte.
- Afecta al 47 % de la población.
- Una muerte cada 33 segundos por problemas cardiovasculares.
- Suponen más de 252.000 millones de dolares al año.



CDC: Centro para el control y prevención de enfermedades de los EEUU



# Objetivos



Análisis exploratorio de datos médicos y demográficos



Diseño de modelos de aprendizaje supervisado

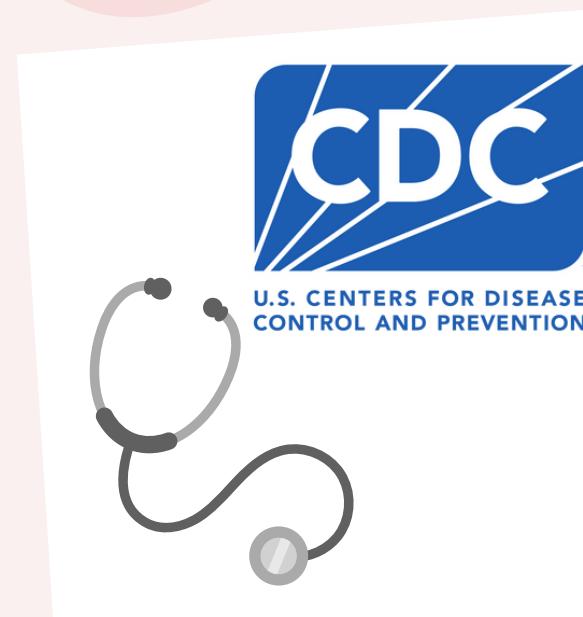


Predicción de probabilidad de ataque cardíaco



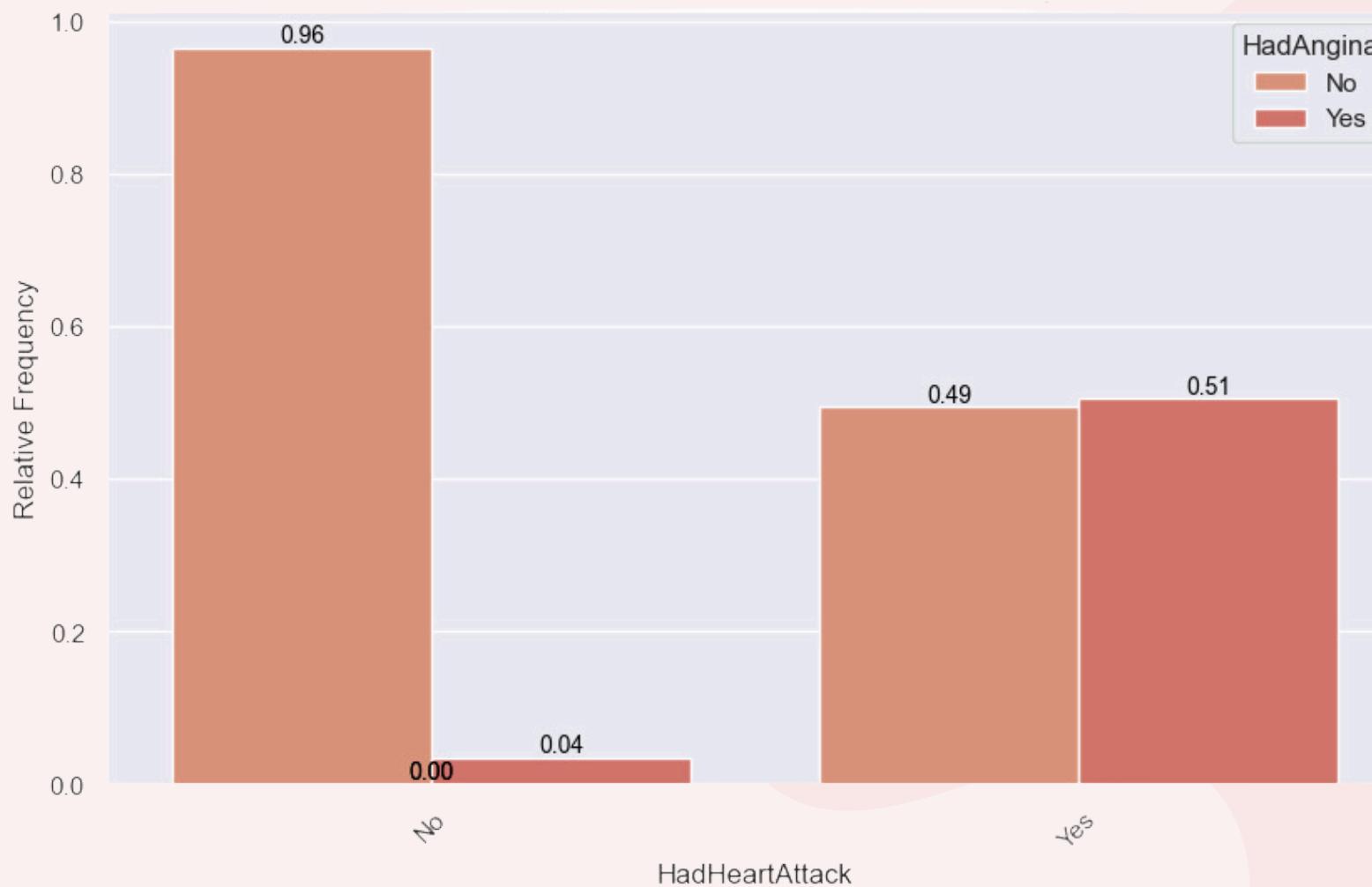
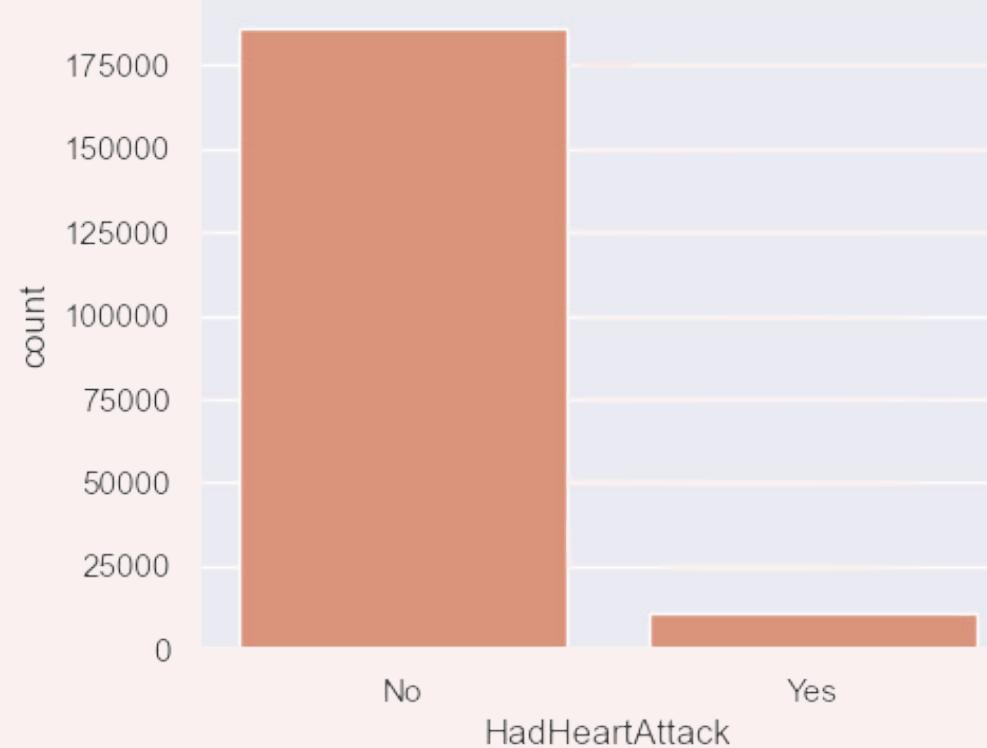
Evaluación de impacto de resultados

# Adquisición de datos



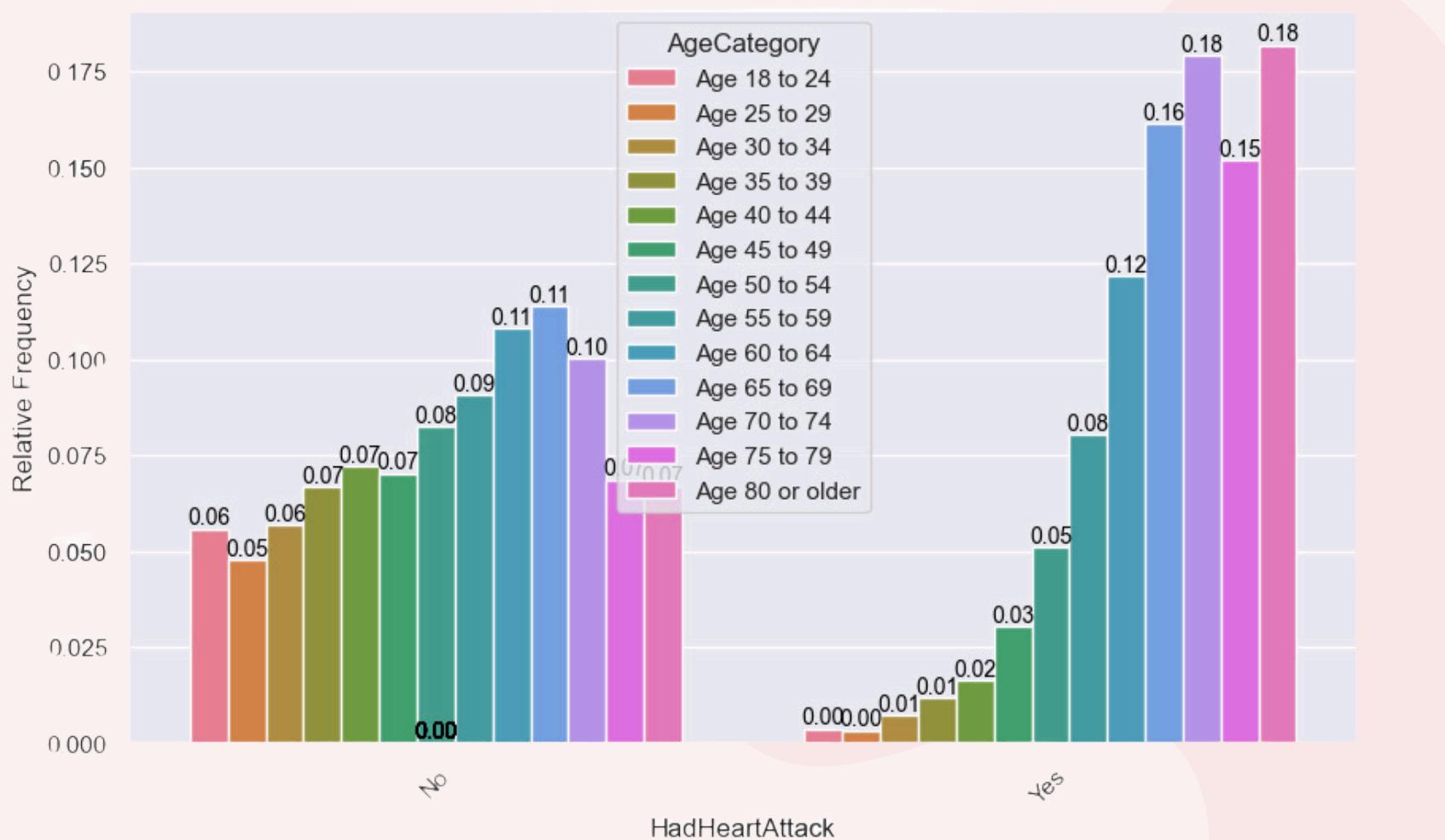
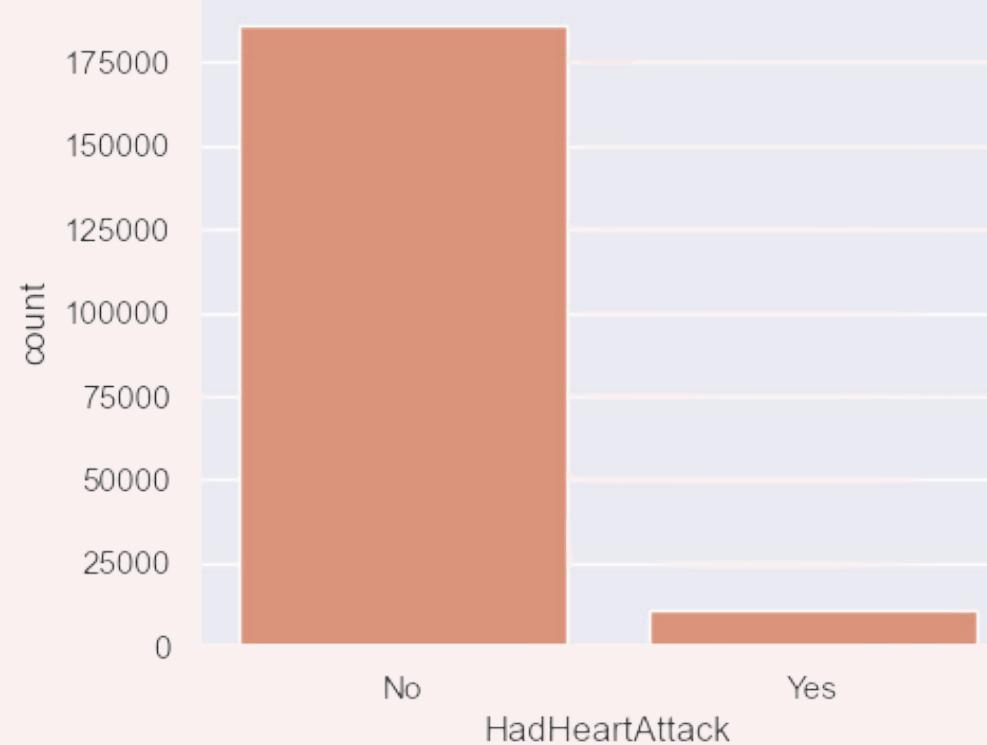
- *BRFSS* : Sistema de vigilancia de factores de riesgo conductuales.
- Iniciativa telefónica establecida en 1984.
- Recopila información de la salud de los 50 estados
  
- Conjunto de datos “reducido” del año 2022.
- Más de 240.000 instancias y 40 parámetros.
- Características demográficas e información médica: salud general, actividad física, edad, enfermedades, dificultad para caminar, IMC, edad, etc.

# Análisis exploratorio



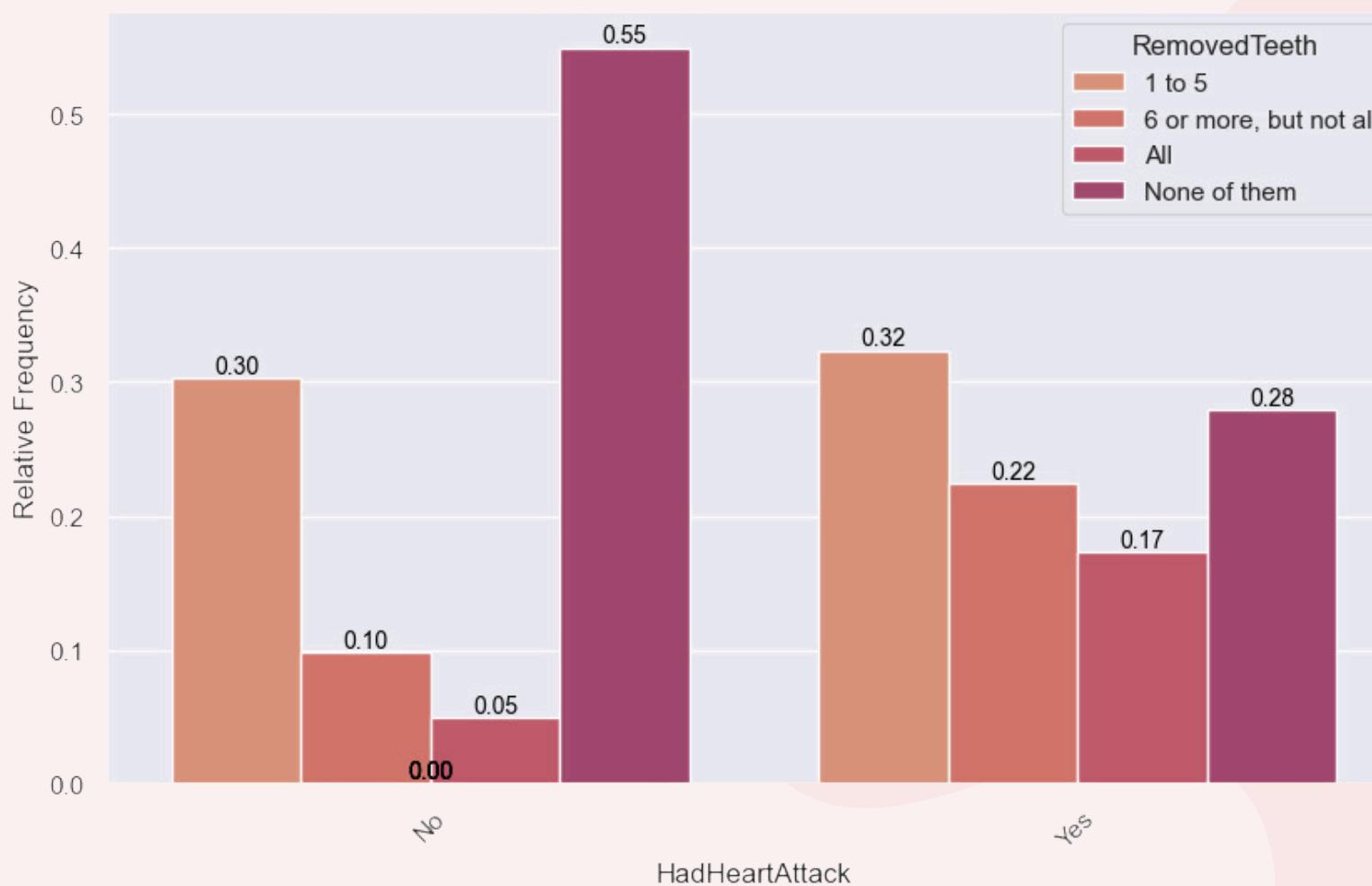
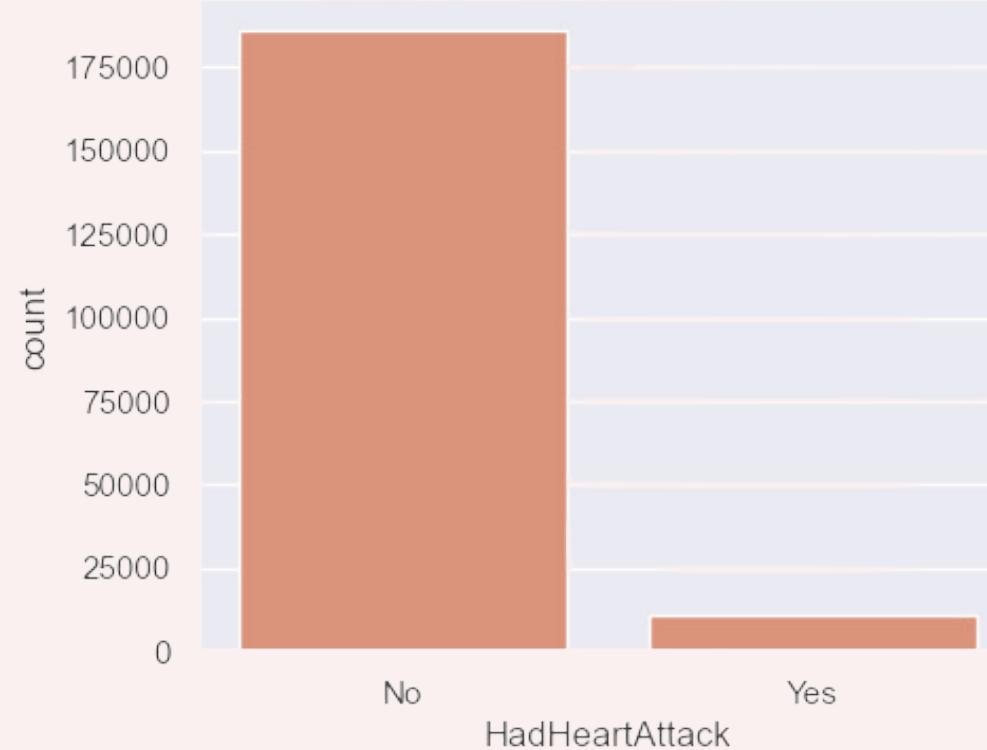
- Clases desbalanceadas.
- Otras enfermedades (anginas, derrames, cánceres, etc).

# Análisis exploratorio



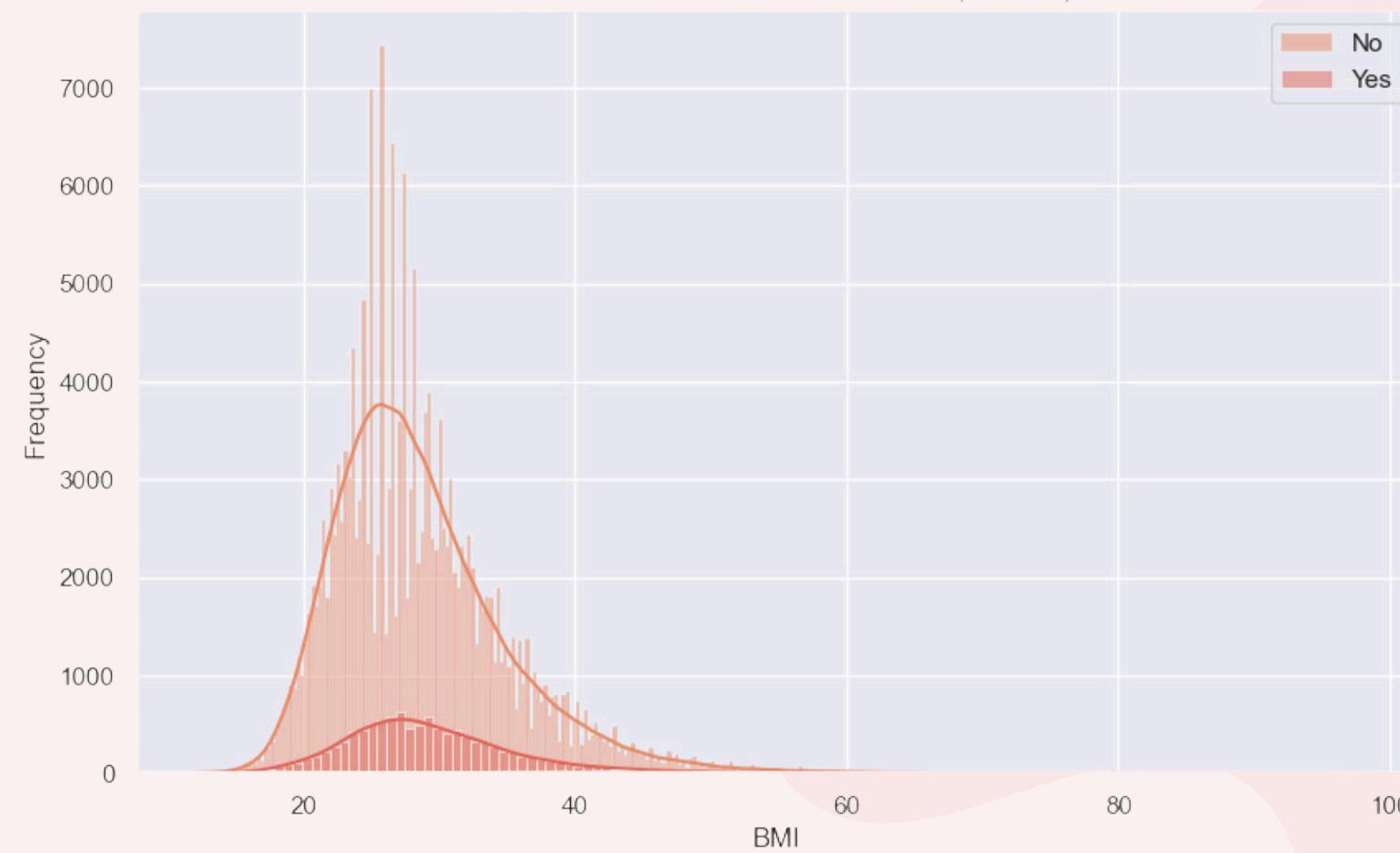
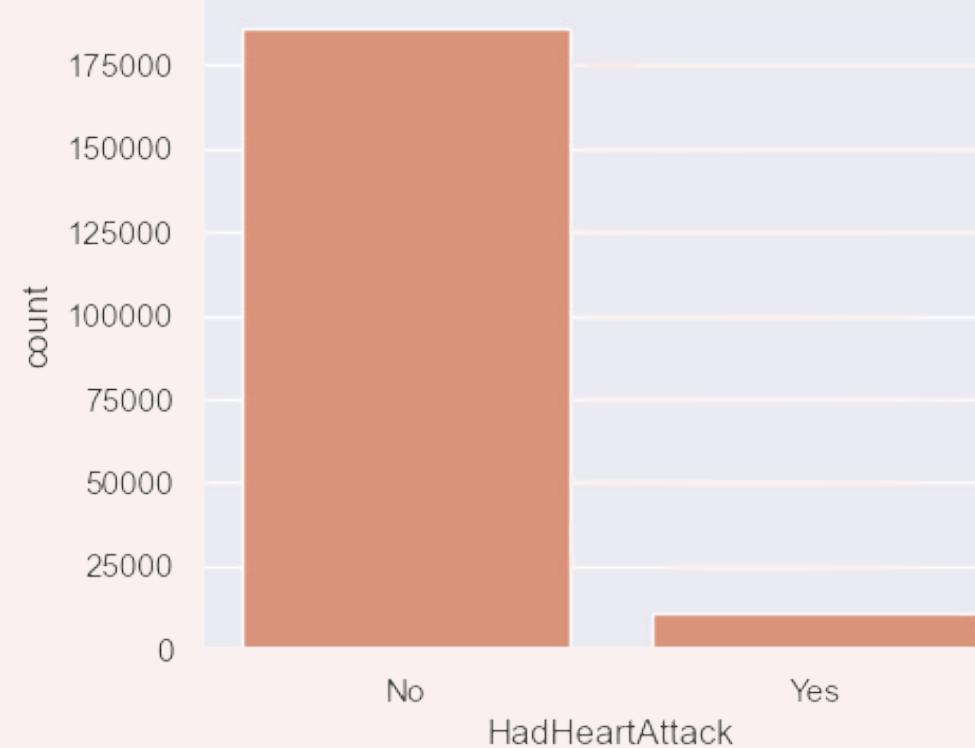
- Clases desbalanceadas.
- Otras enfermedades (anginas, derrames, cánceres, etc).
- Relaciones lógicas o esperadas.

# Análisis exploratorio



- Clases desbalanceadas.
- Otras enfermedades (anginas, derrames, cánceres, etc).
- Relaciones lógicas o esperadas.
- Relaciones no tan lógicas.

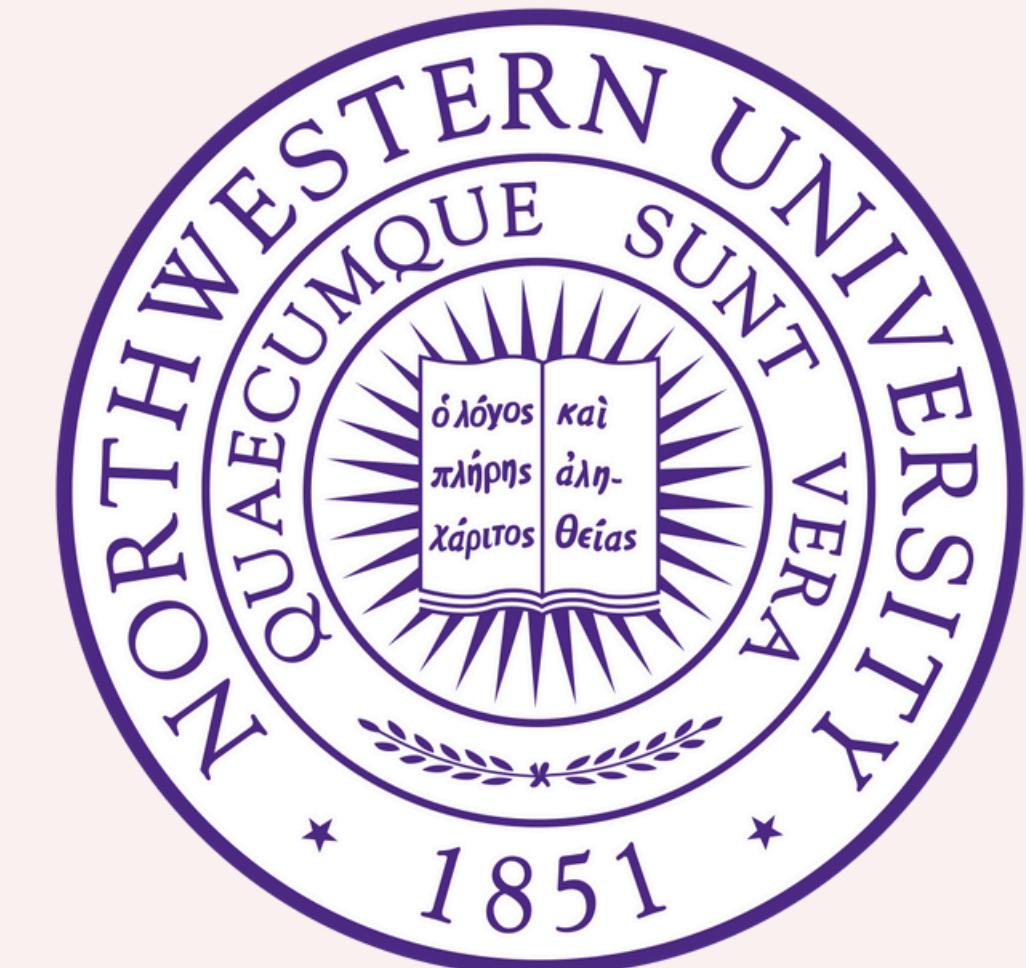
# Análisis exploratorio



- Clases desbalanceadas.
- Otras enfermedades (anginas, derrames, cánceres, etc).
- Relaciones lógicas o esperadas.
- Relaciones no tan lógicas.
- Distribuciones no-Gaussianas.

# Selección de características

- Características **categóricas**: codificación (ordinal y “One-Hot”).
- Características **numéricas**: transformaciones de potencia.
- Selección de características:
  - ~~Análisis visual, colinealidad, información mutua, tests de hipótesis.~~
  - **RFEcv**: Eliminación recursiva de características con validación cruzada.
  - Colaboración con **experta cardióloga** de la Universidad Northwestern (Chicago, IL).



# Selección de modelos

- Optimización: Sensibilidad (**Recall**)
- Minimizar los Falsos Negativos (FN)
- Validación cruzada
- Ponderizar métricas, coste computacional y complejidad



CatBoost  
LightGBM

```
base_models = {  
    'LogisticRegression': LogisticRegression(random_state=random_state),  
    'RandomForestClassifier': RandomForestClassifier(random_state=random_state),  
    'XGBClassifier': XGBClassifier(random_state=random_state),  
    'SVC': SVC(random_state=random_state),  
    'KNeighborsClassifier': KNeighborsClassifier(n_neighbors=5),  
    'GradientBoostingClassifier': GradientBoostingClassifier(random_state=random_state),  
    'AdaBoostClassifier': AdaBoostClassifier(random_state=random_state),  
    'CatBoostClassifier': CatBoostClassifier(random_state=random_state),  
    'LGBMClassifier': LGBMClassifier(random_state=random_state),  
    'Perceptron': Perceptron(random_state=random_state),  
    'SGDClassifier': SGDClassifier(random_state=random_state),  
}
```

```

# pipeline: balance classes, features preprocessing and CatBoost classification model
pipeline = Pipeline([
    ('undersampling', RandomUnderSampler(random_state=random_state)),
    ('preprocessor', ColumnTransformer(
        transformers=[
            ('cat', SimpleImputer(strategy='most_frequent'), CAT_FEATURES),
            ('num', SimpleImputer(strategy='median'), NUM_FEATURES)
        ],
        remainder='drop'
    )),
    ('model', CatBoostClassifier(cat_features=cat_features_index,
                                task_type='GPU',
                                devices='0',
                                random_seed=random_state))
])

# grid parameters
param_grid = {
    'model__iterations': [100, 200, 500],
    'model__learning_rate': [0.01, 0.05, 0.1],
    'model__depth': [5, 7, 10],
    'model__l2_leaf_reg': [1, 5, 7],
    'model__bagging_temperature': [0.0, 1.0, 3.0]
}

# set up GridSearchCV kwargs, using StratifiedKFold
kwargs = {
    'cv': StratifiedKFold(n_splits=5, shuffle=True, random_state=random_state),
    'scoring': 'recall',
    'n_jobs': -5,
    'verbose': 1
}

# hyperparameters tuning
grid_search = GridSearchCV(
    estimator=pipeline,
    param_grid=param_grid,
    **kwargs
)

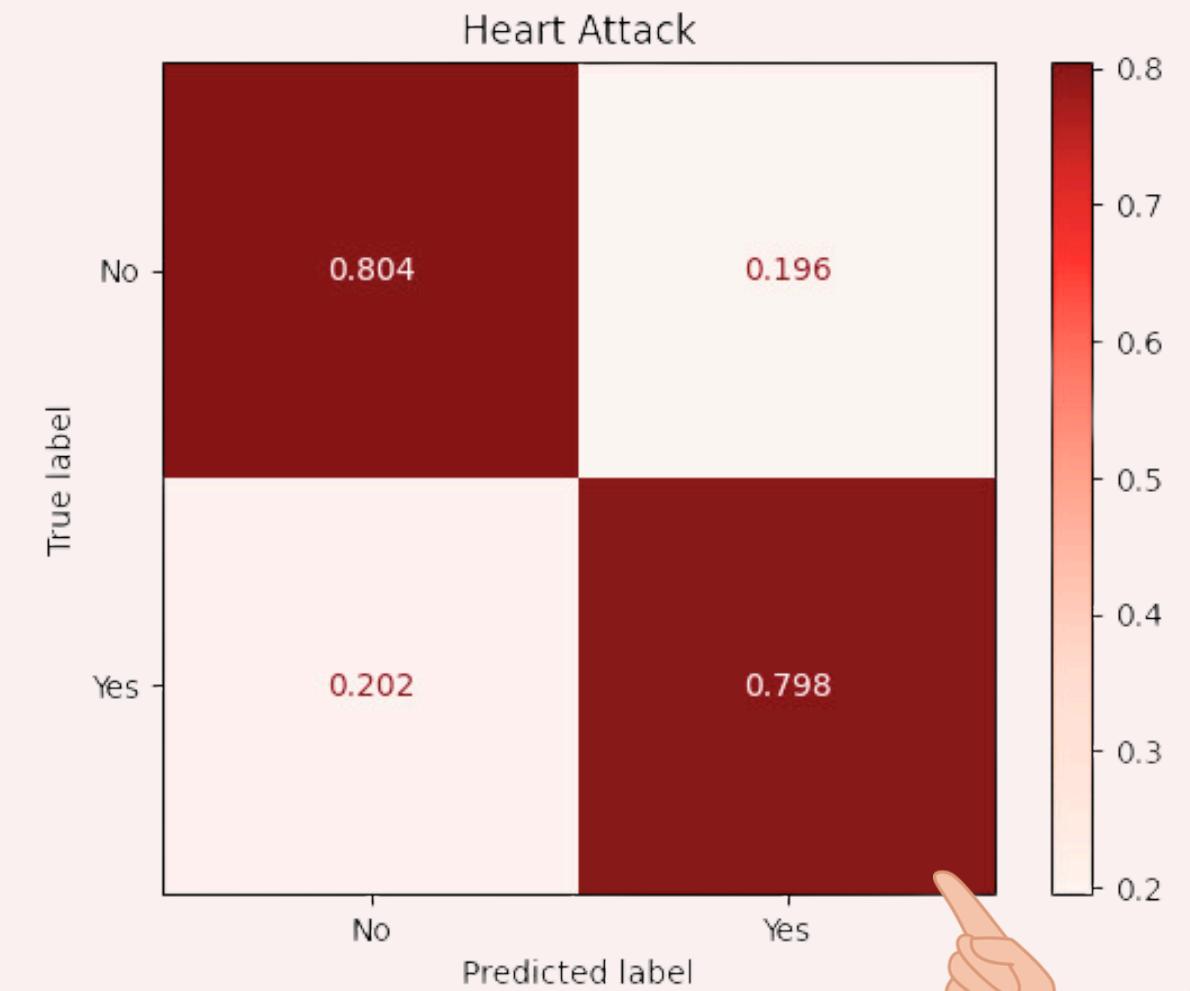
```

# Optimización de hiperparámetros

- *imbalanced-learn Pipelines*
- *GridSearchCV*
- *KFolds*
- Guardar modelo

# Métricas y conclusiones

LightGBM	Precision	Recall	F1-Score	Support
No	0.99	0.80	0.89	46518
Yes	0.19	0.80	0.31	2687
Accuracy	-	-	0.80	49205
Macro AVG	0.59	0.80	0.60	49205
Weighted AVG	0.94	0.80	0.85	49205



- Detectar verdaderos positivos es importante para poder realizar intervenciones a tiempo.
- Detectar verdaderos negativos es igualmente importante, porque falsos positivos pueden generar estrés emocional, ansiedad o técnicas invasivas innecesarias.



# Futuro e implementación

- El proceso de entrenamiento se condensa en un *script* : “*train.py*” (y varios módulos).
- Se guarda el modelo en formato .joblib.
- Objetivo: implementación en un entorno de producción

```
python train.py --model <model_name> --tune --output <output_filename>
```



# Gracias

[lander.combarro@gmail.com](mailto:lander.combarro@gmail.com)

<https://github.com/elecomexp/>