IMMUNE

TECHNOLOGY INSTITUTE

Máster Online Data Science 2022-23
Capstone - Grupo 3
Predict H1N1 and Seasonal Flu Vaccines

Predict H1N1 and Seasonal Flu Vaccines

FICHA TÉCNICA

Título: Predicción de haber recibido la vacuna de la influenza H1N1 y Gripe Estacional

Descripción: Este caso consiste en predecir si las personas se vacunaron contra la gripe H1N1 y la Gripe Estacional usando la información que compartieron sobre sus antecedentes, opiniones y conductas de salud.

Enlace:
https://www.drivendata.org/competitions/66/flu-shot-learning/page/210/

EQUIPO

Gabriela Canales

Adriana Vargas

Diana Aplicano

MENTOR

Javier Castellar

Capstone - Grupo 3
Máster Online Data Science 2022-23

Agenda

- Introducción
- Ficha técnica del dataset
- Procesos de EDA y calidad del dato
- Training target H1N1 / Hyperparameters tuning
- Training target Seasonal / Hyperparameters tuning
- Otros rounds de entrenamiento
- Conclusiones y Trabajo Futuro

Introducción

Predicción de haber recibido la vacuna de la influenza H1N1 y Gripe Estacional.

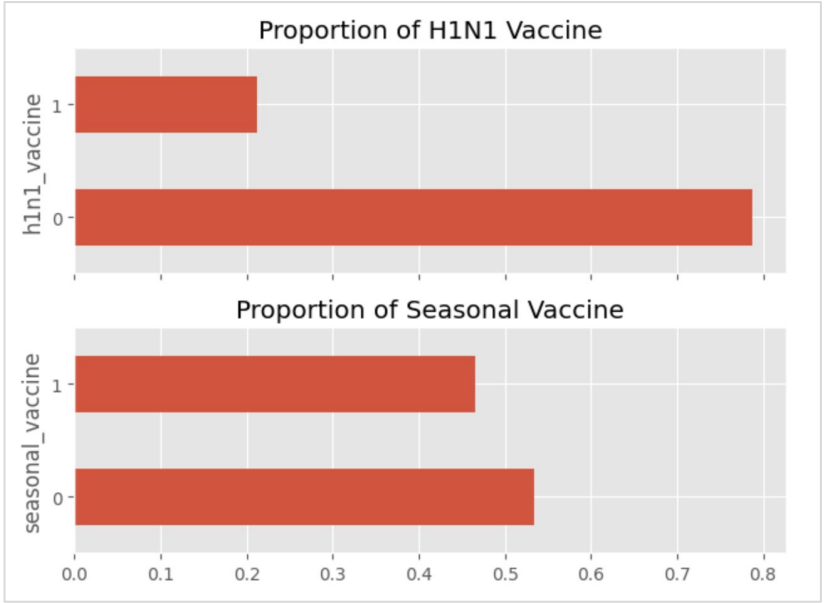
Contexto	Vacunación, como una medida de salud pública clave, usada para combatir enfermedades infecciosas
Campo	Data Science for social good
Tipo de Dataset	Público
Organización	United States National Center for Health Statistics
Tipo de Problema	Multi target classification

FICHA TÉCNICA DEL DATASET

Dataset

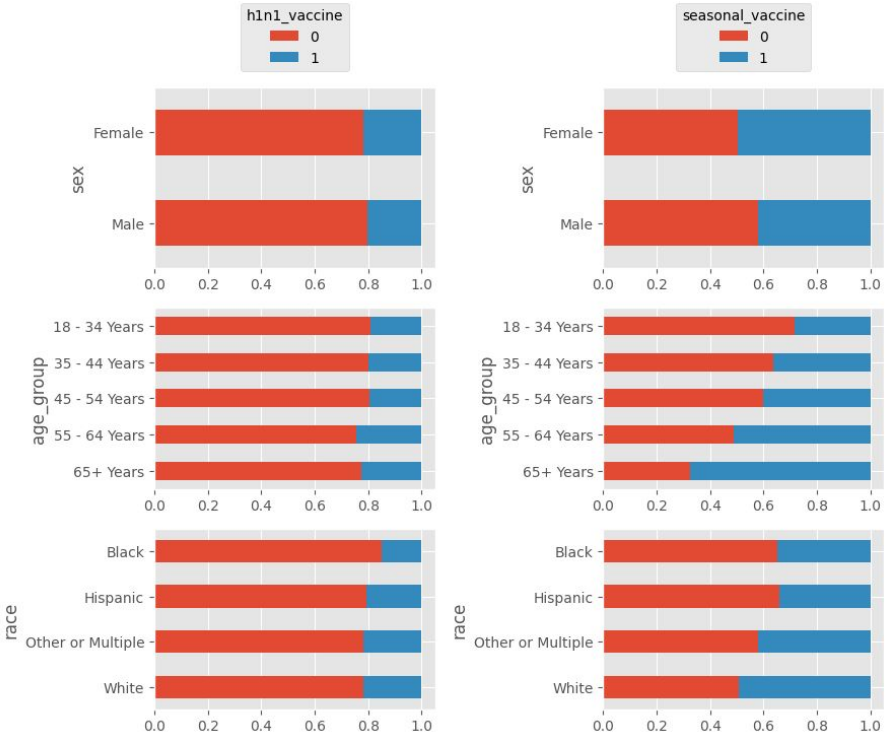
En esta fase analizamos el dataset, la distribución de los datos para las 2 variables target, y su relación con algunas variables independientes.

FICHA TÉCNICA	
Número de registros	26707
Número de variables independientes	36
Variables numéricas	24
Variables categóricas	12
Variables numéricas binarias	16
Variables tipo fecha	0
Variables target	2

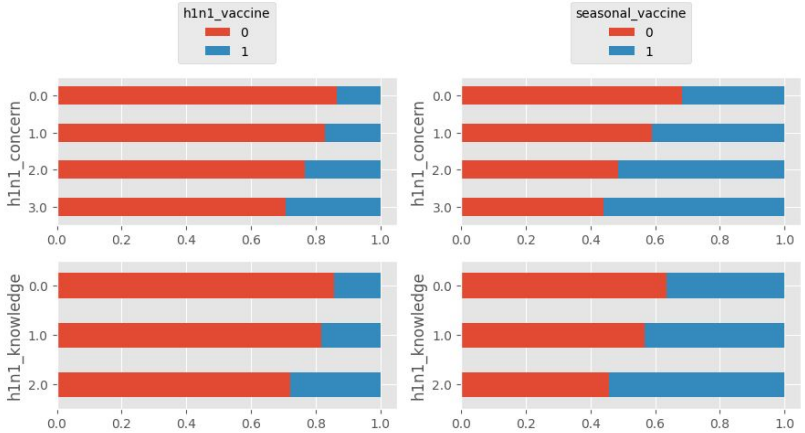


La mitad de la población recibió la vacuna de gripe estacional (clases balanceadas), pero solo un 20% la de H1N1 (clases moderadamente desbalanceadas)

Dataset



Distribución poblacional de quienes recibieron / no recibieron las vacunas.



Nivel de preocupación/conocimiento por cada tipo de vacuna.

Concern:
0 = Not at all concerned; 1 = Not very concerned; 2 = Somewhat concerned; 3 = Very concerned.
Knowledge:
0 = No knowledge; 1 = A little knowledge; 2 = A lot of knowledge.

PROCESOS DE EDA
Y CALIDAD DEL DATO

Calidad del Dato

En la fase de EDA (Exploratory Data Analysis), eliminamos algunas variables irrelevantes, analizamos la correlación entre variables, tratamos los valores nulos, y las variables categóricas fueron convertidas en numéricas.

Variables Categóricas (12)

Antes de EDA

	empty_rows	Factor1	CasesFactor1	Factor2	CasesFactor2	Factor3	CasesFactor3	unique_values
age_group	0	65+ Years	6843	55 - 64 Years	5563	45 - 54 Years	5238	5
education	1407	College Graduate	10097	Some College	7043	12 Years	5797	4
race	0	White	21222	Black	2118	Hispanic	1755	4
sex	0	Female	15858	Male	10849			2
income_poverty	4423	<= $75,000, Above Poverty	12777	> $75,000	6810	Below Poverty	2697	3
marital_status	1408	Married	13555	Not Married	11744			2
rent_or_own	2042	Own	18736	Rent	5929			2
employment_status	1463	Employed	13560	Not in Labor Force	10231	Unemployed	1453	3
hhs_geo_region	0	lzgpxyit	4297	fpwskwrf	3265	qufhixun	3102	10
census_msa	0	MSA, Not Principle City	11645	MSA, Principle City	7864	Non-MSA	7198	3
employment_industry	13330	fcxhlnwr	2468	wxleyezf	1804	ldnlellj	1231	21
employment_occupation	13470	xtkaffoo	1778	mxkfnird	1509	emcorrxb	1270	23

- → *3 variables categóricas binarias*
- → *Variables* `hhs_geo_region`, `employment_industry`, `employment_occupation`, con datos irrelevantes.

Calidad del Dato

Variables Numéricas (24)

	empty_rows	mean	median	min	max	unique_values
respondent_id	0	13353	13353	0	26706	26707
h1n1_concern	92	1,618485816	2	0	3	4
h1n1_knowledge	116	1,262532436	1	0	2	3
behavioral_antiviral_meds	71	0,04884367022	0	0	1	2
behavioral_avoidance	208	0,7256122873	1	0	1	2
behavioral_face_mask	19	0,06898231415	0	0	1	2
behavioral_wash_hands	42	0,8256141009	1	0	1	2
behavioral_large_gatherings	87	0,3586401202	0	0	1	2
behavioral_outside_home	82	0,337314554	0	0	1	2
behavioral_touch_face	128	0,6772640054	1	0	1	2
doctor_recc_h1n1	2160	0,2203120544	0	0	1	2
doctor_recc_seasonal	2160	0,3297347945	0	0	1	2
chronic_med_condition	971	0,283260802	0	0	1	2
child_under_6_months	820	0,08258971685	0	0	1	2
health_worker	804	0,1119175385	0	0	1	2
health_insurance	12274	0,8797200859	1	0	1	2
opinion_h1n1_vacc_effective	391	3,850623195	4	1	5	5
opinion_h1n1_risk	388	2,342566207	2	1	5	5
opinion_h1n1_sick_from_vacc	395	2,357669504	2	1	5	5
opinion_seas_vacc_effective	462	4,025985902	4	1	5	5
opinion_seas_risk	514	2,719161608	2	1	5	5
opinion_seas_sick_from_vacc	537	2,118112342	2	1	5	5
household_adults	249	0,8864993575	1	0	3	4
household_children	249	0,5345831129	0	0	3	4

Antes de EDA

- ➔ *13 variables numéricas binarias*
- ➔ *60762 valores nulos en total*

Calidad del Dato

Variables Numéricas (49)

	empty_rows	mean	median	min	max	unique_values
respondent_id	0	13353	13353	0	26706	26707
h1n1_concern	0	1,619800052	2	0	3	4
h1n1_knowledge	0	1,261392144	1	0	2	3
behavioral_antiviral_meds	0	0,04871382035	0	0	1	2
behavioral_avoidance	0	0,7277492792	1	0	1	2
behavioral_face_mask	0	0,06893323848	0	0	1	2
behavioral_wash_hands	0	0,8258883439	1	0	1	2
behavioral_large_gatherings	0	0,3574718239	0	0	1	2
behavioral_outside_home	0	0,3362788782	0	0	1	2
behavioral_touch_face	0	0,6788107987	1	0	1	2
doctor_recc_h1n1	0	0,2024937282	0	0	1	2
doctor_recc_seasonal	0	0,3030666117	0	0	1	2
chronic_med_condition	0	0,2729621448	0	0	1	2
child_under_6_months	0	0,08005391845	0	0	1	2
health_worker	0	0,1085483207	0	0	1	2
health_insurance	0	0,4754184296	0	0	1	2
opinion_h1n1_vacc_effective	0	3,852810125	4	1	5	5
opinion_h1n1_risk	0	2,337589396	2	1	5	5
opinion_h1n1_sick_from_vacc	0	2,352379526	2	1	5	5
opinion_seas_vacc_effective	0	4,025536376	4	1	5	5
opinion_seas_risk	0	2,705320702	2	1	5	5
opinion_seas_sick_from_vacc	0	2,095630359	2	1	5	5
household_adults	0	0,8875575692	1	0	3	4
household_children	0	0,5295989815	0	0	3	4

Después de EDA

- Número final de variables: 49.
- Se eliminaron las columnas `hhs_geo_region`, `employment_industry`, `employment_occupation`, por no tener datos relevantes.
- Se eliminó la variable `income_poverty` que presenta el mayor número de valores nulos, y baja correlación con las variables target.

IMMUNE CODING INSTITUTE

Calidad del Dato

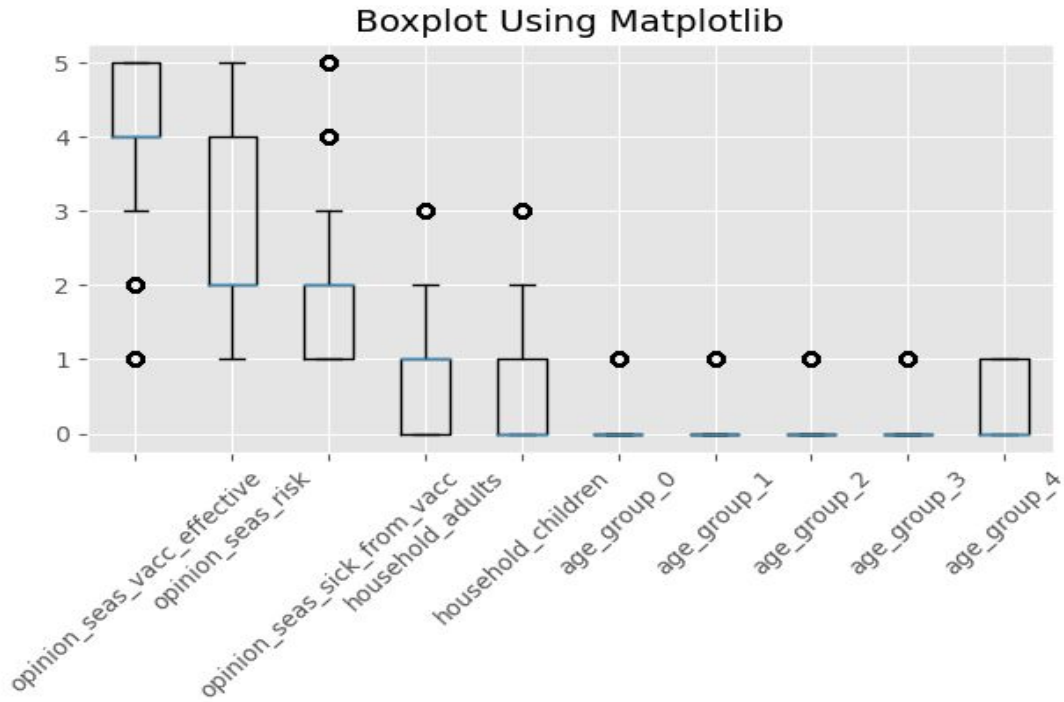
Variables Numéricas (49)

	empty_rows	mean	median	min	max	unique_values
age_group_0	0	0,1952671584	0	0	1	2
age_group_1	0	0,1440820759	0	0	1	2
age_group_2	0	0,1961283559	0	0	1	2
age_group_3	0	0,2082974501	0	0	1	2
age_group_4	0	0,2562249597	0	0	1	2
education_0	0	0,217059198	0	0	1	2
education_1	0	0,088478676	0	0	1	2
education_2	0	0,4307484929	0	0	1	2
education_3	0	0,2637136331	0	0	1	2
race_0	0	0,07930505111	0	0	1	2
race_1	0	0,06571310892	0	0	1	2
race_2	0	0,06035870745	0	0	1	2
race_3	0	0,7946231325	1	0	1	2
sex_0	0	0,5937769124	1	0	1	2
sex_1	0	0,4062230876	0	0	1	2
marital_status_0	0	0,560265099	1	0	1	2
marital_status_1	0	0,439734901	0	0	1	2
rent_or_own_0	0	0,7779982776	1	0	1	2
rent_or_own_1	0	0,2220017224	0	0	1	2
employment_status_0	0	0,5625117011	1	0	1	2
employment_status_1	0	0,3830830868	0	0	1	2
employment_status_2	0	0,05440521212	0	0	1	2
census_msa_0	0	0,4360280076	0	0	1	2
census_msa_1	0	0,2944546374	0	0	1	2
census_msa_2	0	0,269517355	0	0	1	2

Después de EDA

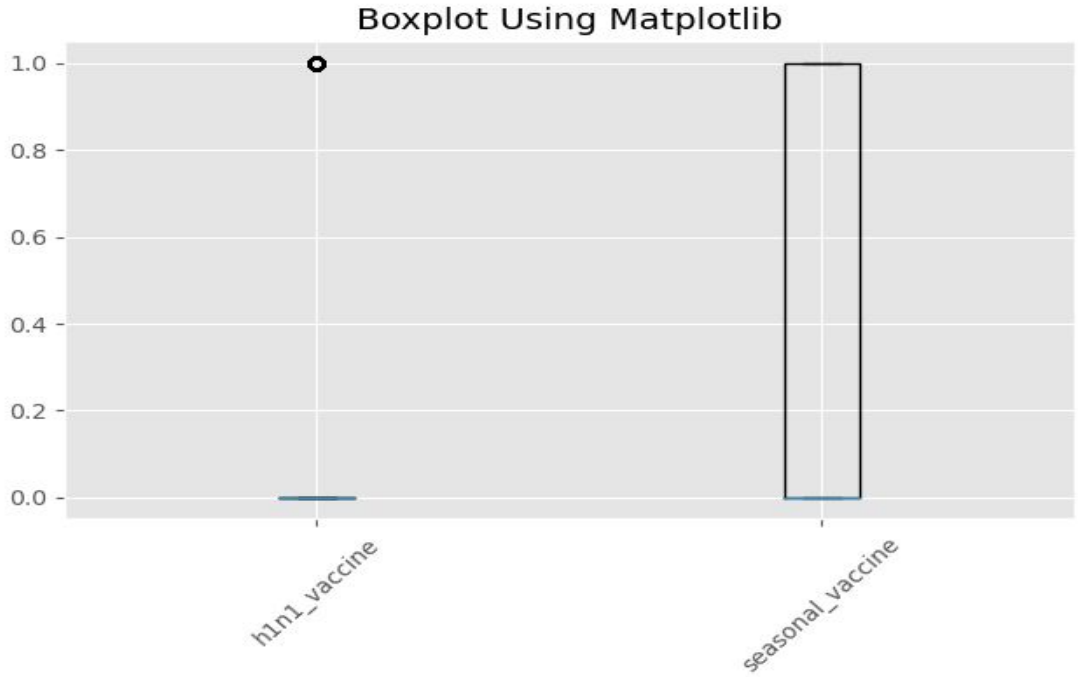
- Se usó Label Encoder y One-Hot Encoder para convertir las variables categóricas en numéricas, después de llenar los valores nulos con la moda (`age_group`, `education`, `race`, `sex`, `marital_status`, `rent_or_own`, `employment_status`, `census_msa`).
- Se poblaron los nulos de las variables `behavioral_outside_home` y `behavioral_large_gatherings` entre ellas, dado que tienen una correlación de +0.59
- Se poblaron los nulos de las variables `doctor_recc_h1n1` y `doctor_recc_seasonal` entre ellas, dado que tienen una correlación de +0.67

Identificación de outliers - Algunas Features



Realizamos un análisis de algunas de las variables del dataset para identificar la cantidad de outliers presentes.

Identificación de outliers - Variables Target



h1n1_Vaccine = **5674 outliers**

Seasonal_Vaccine = **0 outliers**

Training target H1N1
Hyperparameters tuning

Training H1N1

Para la predicción de la variable target H1N1, realizamos la ejecución de Hyperparameters Tuning con GridSearchCV y training de los algoritmos, en los siguientes rounds:

FICHA TÉCNICA	
Hyperparameters Tuning	GridSearchCV
Scoring	roc_auc
Algoritmos evaluados	1. Logistic Regression 2. Random Forest 3. SVM (Support Vector Machines) 4. Naive Bayes 5. Decision Tree 6. K-Neighbors

ROUNDS			
	SPLIT	**TOP 3**	**HIGHEST SCORE**
1	80%-20%	1. SVM (Support Vector Machines) 2. Random Forest 3. Logistic Regression	SVM balanced **AUC = 0.862754**
2	70%-30%	1. SVM (Support Vector Machines) 2. Random Forest 3. Logistic Regression	SVM balanced AUC = 0.858395

***** SVM balanced = Ejecución con `class_weight='balanced'`.

IMMUNE CODING INSTITUTE

Training H1N1 - Round 1

Ejecución de Hyperparameters Tuning con GridSearchCV y training, round 1 con split 80%-20%.
Con y sin el parámetro `class_weight='balanced'`.

FICHA TÉCNICA	
Hyperparameters Tuning	GridSearchCV
Scoring	roc_auc
Algoritmos evaluados	1. Logistic Regression 2. Random Forest 3. SVM (Support Vector Machines) 4. Naive Bayes 5. Decision Tree 6. K-Neighbors
Split training and evaluation set	80%-20%
Top 3	1. SVM (Support Vector Machines) 2. Random Forest 3. Logistic Regression

	estimator	target	params	score
multi_class	lr	h1n1	multinomial	0.852310
penalty	lr	h1n1	l1	0.852310
solver	lr	h1n1	saga	0.852310
C	lr	h1n1	0.5	0.852310
n_estimators	rf	h1n1	500	0.850748
criterion	rf	h1n1	entropy	0.850748
max_features	rf	h1n1	auto	0.850748
max_depth	rf	h1n1	8	0.850748
C	svm	h1n1	1	0.850106
gamma	svm	h1n1	0.01	0.850106
kernel	svm	h1n1	rbf	0.850106
var_smoothing	naive	h1n1	0.02848	0.808018
max_features	dt	h1n1	auto	0.803148
max_depth	dt	h1n1	7	0.803148
criterion	dt	h1n1	entropy	0.803148
ccp_alpha	dt	h1n1	0.001	0.803148
n_neighbors	kn	h1n1	30	0.801494

AUC = 0.858524
AUC = 0.858176

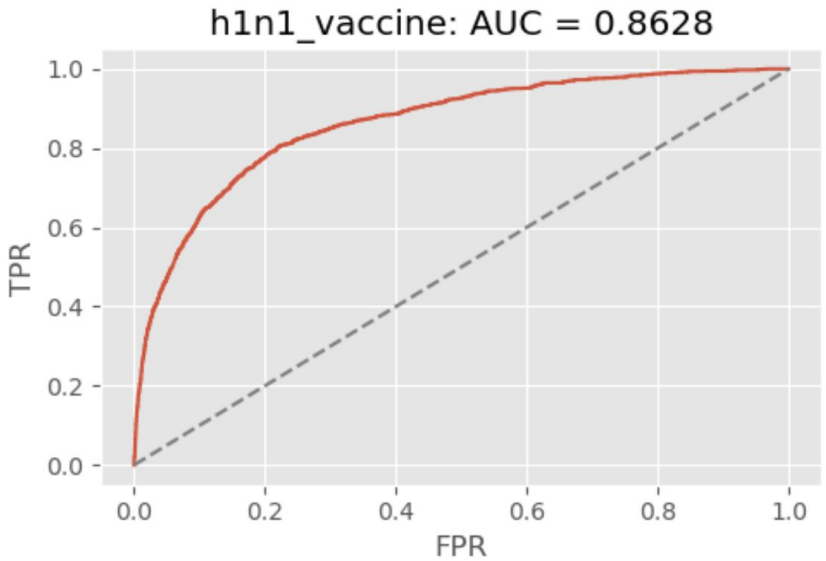
AUC = 0.857541
AUC = 0.858906

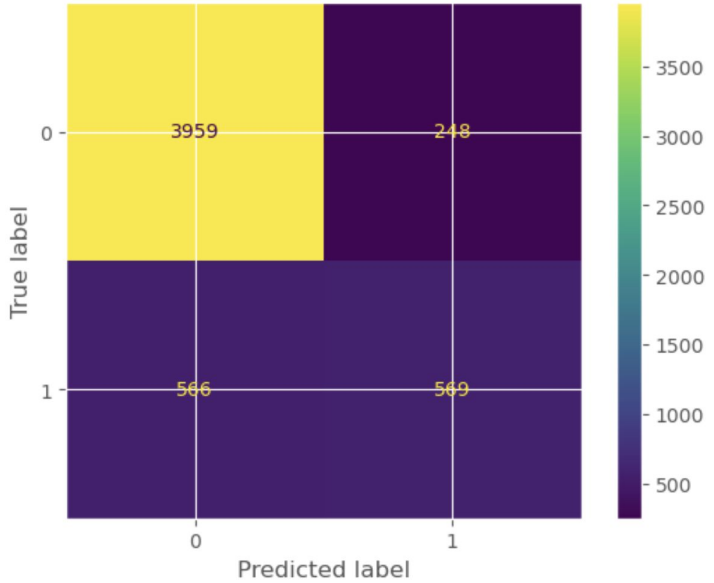
AUC = 0.847230
AUC = 0.862754

*El score más alto es **0.862754** con SVM balanced.*

Training H1N1 - Round 1

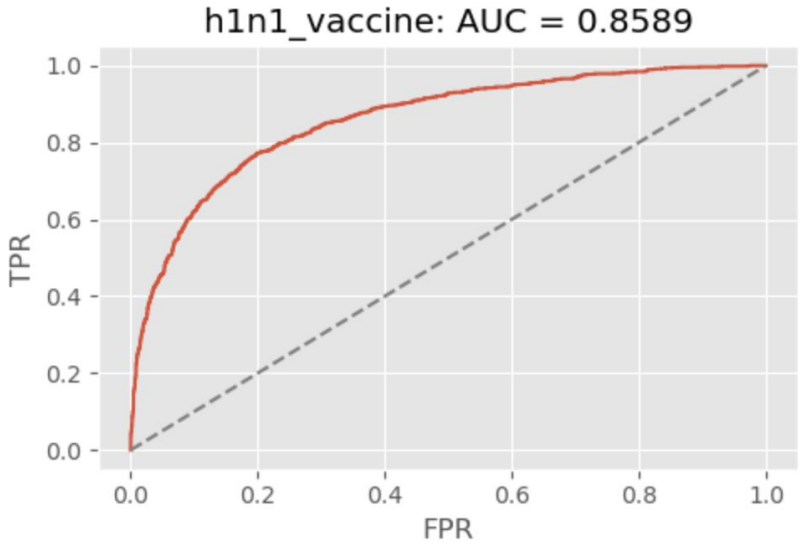
best estimator	C	gamma	kernel	class_weight	auc
SVM	1	0.01	rbf	balanced	**0.862754**

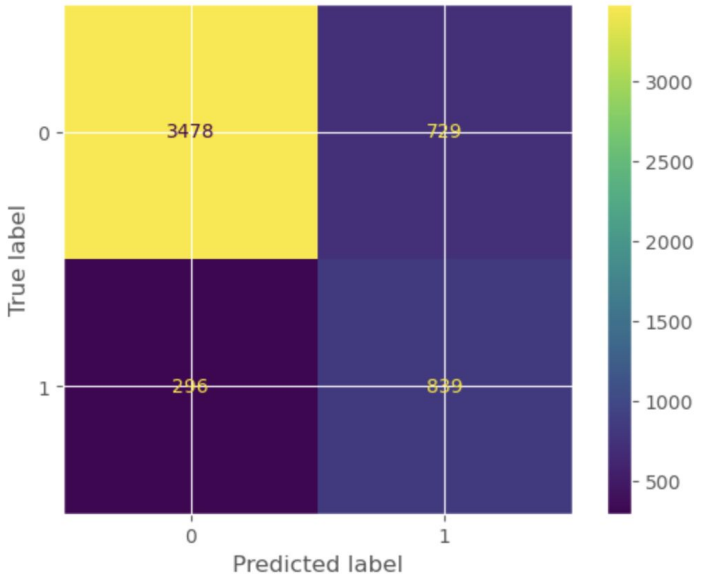




Training H1N1 - Round 1

estimator	criterion	max_depth	max_features	n_estimators	class_weight	auc
Random Forest	entropy	8	auto	500	balanced	0.858906

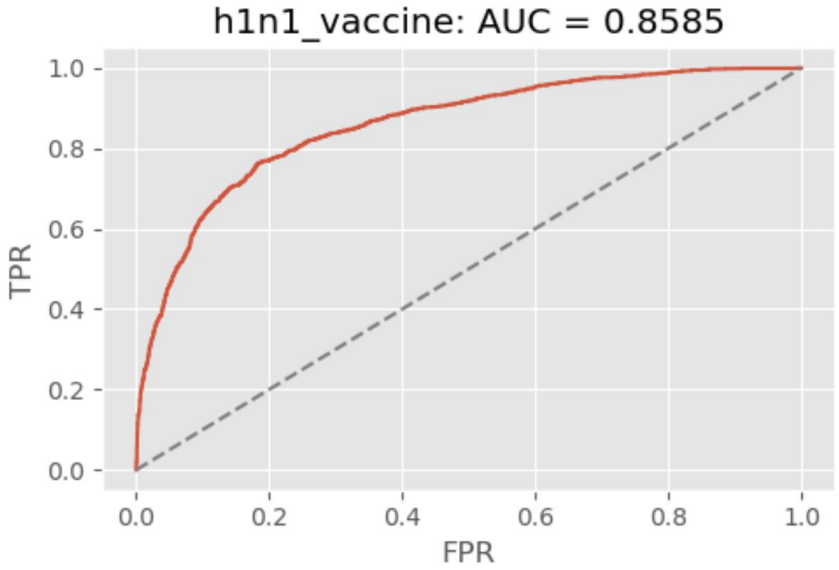


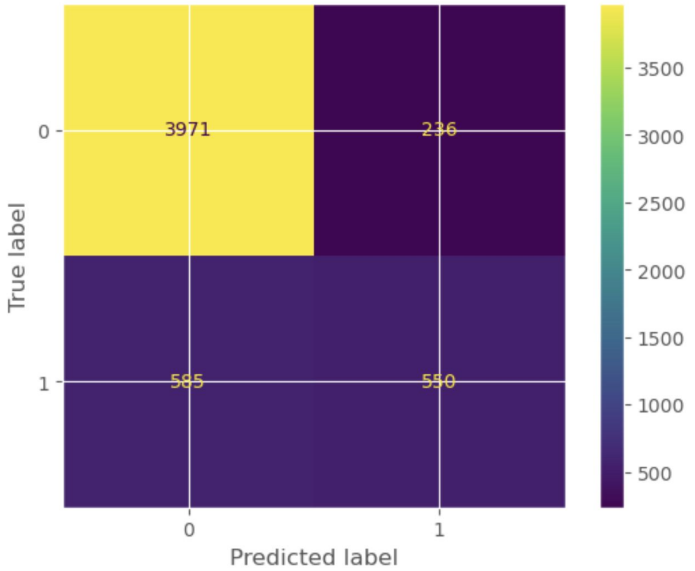


IMMUNE CODING INSTITUTE

Training H1N1 - Round 1

estimator	C	solver	multi_class	penalty	class_weight	auc
Logistic Regression	0.5	saga	multinomial	l1	None	0.858524





IMMUNE CODING INSTITUTE

Training H1N1 - Round 2

Ejecución de Hyperparameters Tuning con GridSearchCV y training, round 1 con split 70%-30%.
Con y sin el parámetro `class_weight='balanced'`.

FICHA TÉCNICA	
Tool	GridSearchCV
Scoring	roc_auc
Algoritmos evaluados	1. Logistic Regression 2. Random Forest 3. SVM (Support Vector Machines) 4. Naive Bayes 5. Decision Tree 6. K-Neighbors
Split training and evaluation set	70%-30%
Top 3	1. SVM (Support Vector Machines) 2. Random Forest 3. Logistic Regression

	estimator	target	params	score
multi_class	lr	h1n1	multinomial	0.854133
penalty	lr	h1n1	l2	0.854133
solver	lr	h1n1	lbfgs	0.854133
C	lr	h1n1	0.5	0.854133
n_estimators	rf	h1n1	500	0.852615
criterion	rf	h1n1	entropy	0.852615
max_features	rf	h1n1	auto	0.852615
max_depth	rf	h1n1	8	0.852615
C	svm	h1n1	1	0.851652
gamma	svm	h1n1	0.01	0.851652
kernel	svm	h1n1	rbf	0.851652
var_smoothing	naive	h1n1	0.02848	0.809980
max_features	dt	h1n1	auto	0.807349
max_depth	dt	h1n1	9	0.807349
criterion	dt	h1n1	entropy	0.807349
ccp_alpha	dt	h1n1	0.001	0.807349
n_neighbors	kn	h1n1	30	0.803284

LOGISTIC REGRESSION
AUC = 0.853926
AUC = 0.854038

RANDOM FOREST
AUC = 0.856697
AUC = 0.856729

SVM (SUPPORT VECTOR MACHINES)
AUC = 0.846443
AUC = 0.858395

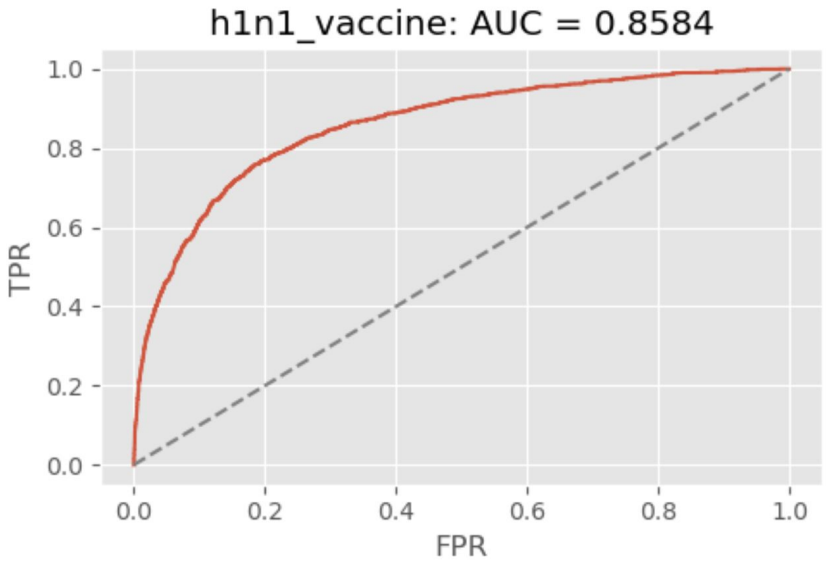
AUC no balanced

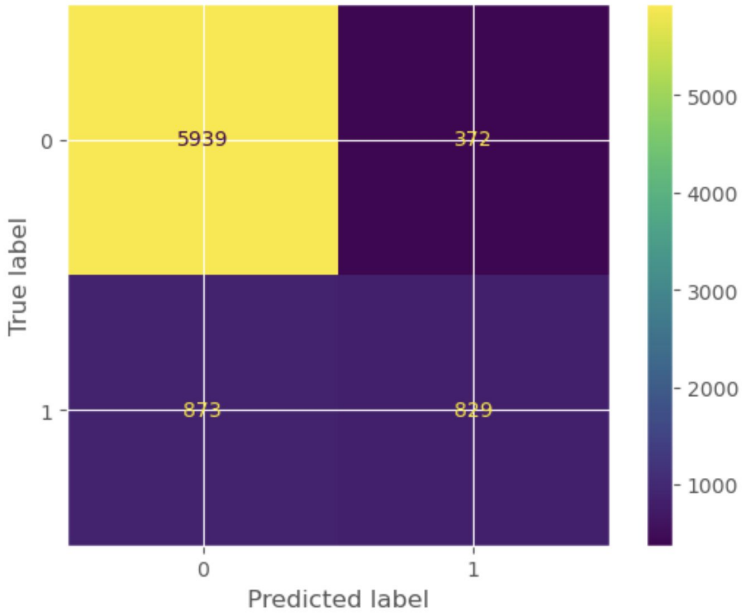
AUC balanced

*El score más alto es **0.858395** con SVM.*

Training H1N1 - Round 2

best estimator	C	gamma	kernel	class_weight	auc
SVM	1	0.01	rbf	balanced	**0.858395**

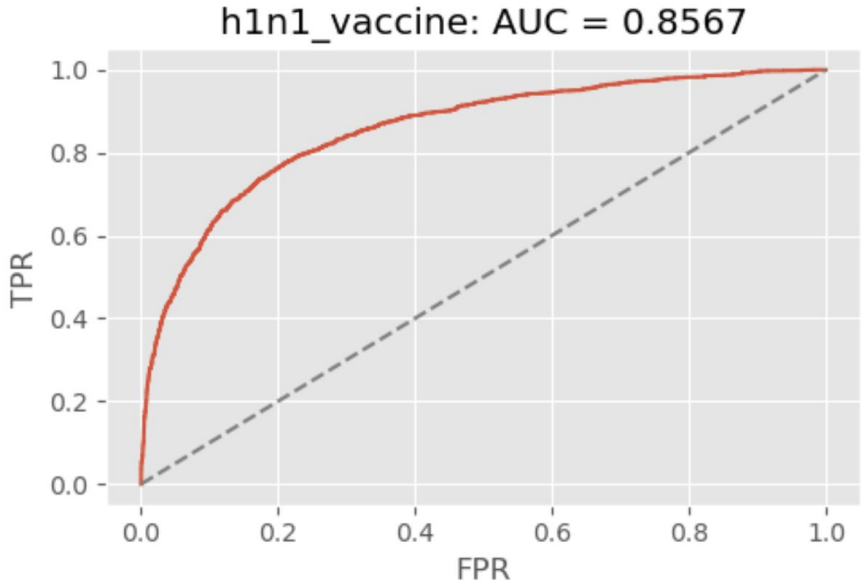


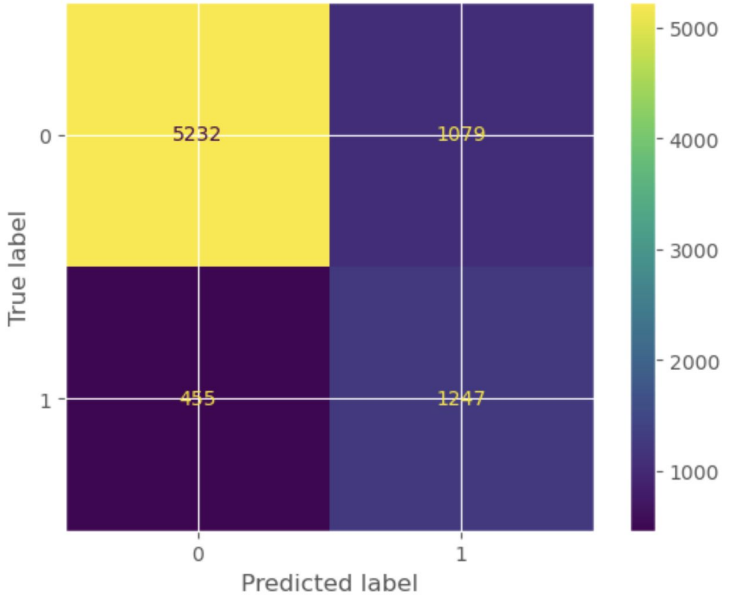


IMMUNE CODING INSTITUTE

Training H1N1 - Round 1

estimator	criterion	max_depth	max_features	n_estimators	class_weight	auc
Random Forest	entropy	8	auto	500	balanced	**0.856729**

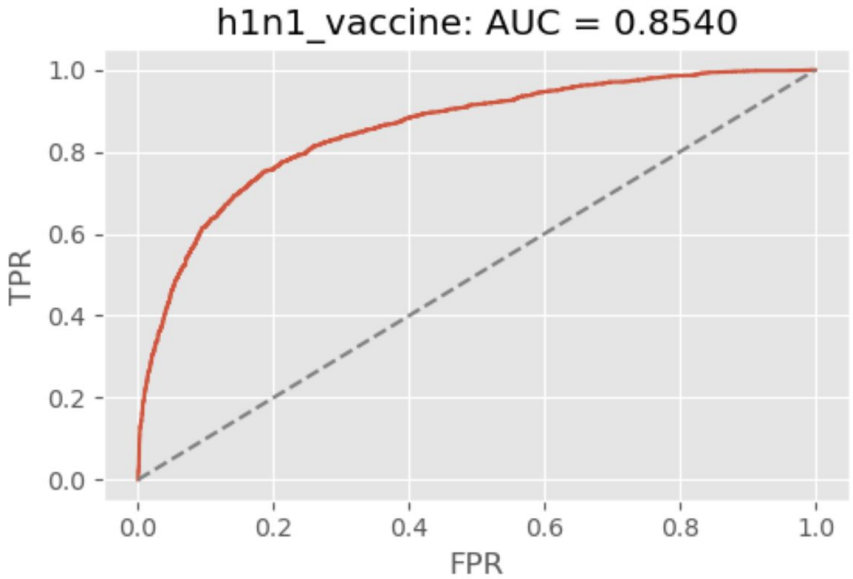


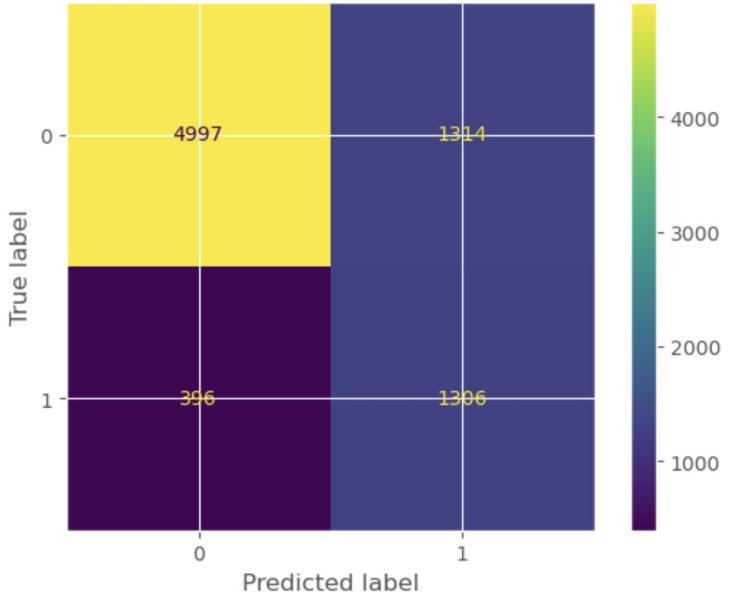


IMMUNE CODING INSTITUTE

Training H1N1 - Round 1

estimator	C	solver	multi_class	penalty	class_weight	auc
Logistic Regression	0.5	saga	multinomial	l1	balanced	0.854038





Training target SEASONAL
Hyperparameters tuning

Training Seasonal

Para la predicción de la variable target Seasonal, realizamos la ejecución de Hyperparameters Tuning con GridSearchCV y training de los algoritmos, en los siguientes rounds:

FICHA TÉCNICA	
Hyperparameters Tuning	GridSearchCV
Scoring	roc_auc
Algoritmos evaluados	1. Logistic Regression 2. Random Forest 3. SVM (Support Vector Machines) 4. Naive Bayes 5. Decision Tree 6. K-Neighbors

ROUNDS			
	SPLIT	**TOP 3**	**HIGHEST SCORE**
1	80%-20%	1. SVM (Support Vector Machines) 2. Random Forest 3. K-Neighbors	SVM balanced AUC = 0.855847
2	70%-30%	1. SVM (Support Vector Machines) 2. Random Forest 3. K-Neighbors	SVM balanced **AUC = 0.858624**

***** SVM balanced = Ejecución con `class_weight='balanced'`.

IMMUNE CODING INSTITUTE

Training Seasonal - Round 1

Ejecución de Hyperparameters Tuning con GridSearchCV y training, round 1 con split 80%-20%.
Con y sin el parámetro `class_weight='balanced'`.

FICHA TÉCNICA	
Hyperparameters Tuning	GridSearchCV
Scoring	roc_auc
Algoritmos evaluados	1. Logistic Regression 2. Random Forest 3. SVM (Support Vector Machines) 4. Naive Bayes 5. Decision Tree 6. K-Neighbors
Split training and evaluation set	80%-20%
Top 3	1. SVM (Support Vector Machines) 2. Random Forest 3. K-Neighbors

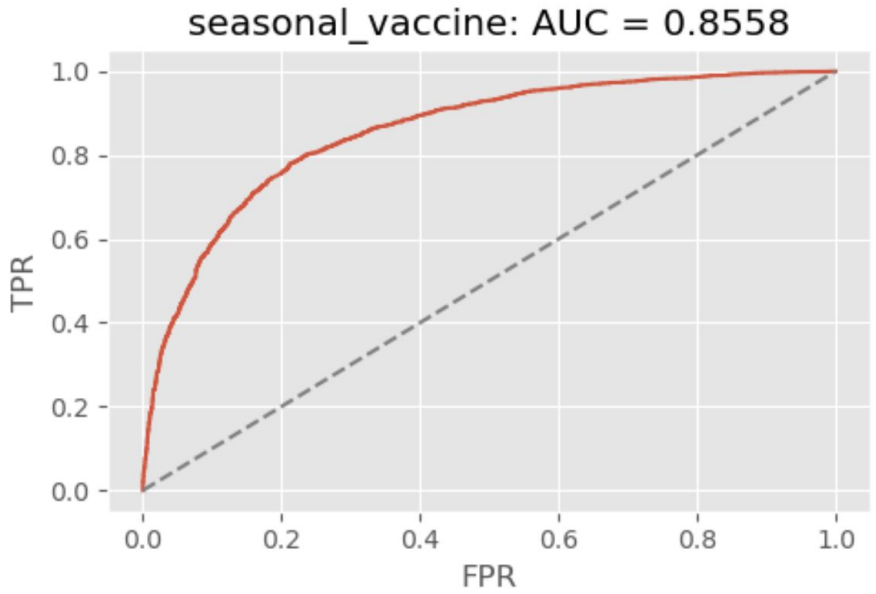
	estimator	target	params	score
C	svm	seasonal	1	0.853812
gamma	svm	seasonal	0.01	0.853812
kernel	svm	seasonal	rbf	0.853812
criterion	rf	seasonal	entropy	0.848905
n_estimators	rf	seasonal	500	0.848905
max_features	rf	seasonal	auto	0.848905
max_depth	rf	seasonal	8	0.848905
n_neighbors	kn	seasonal	30	0.823578
var_smoothing	naive	seasonal	0.053367	0.816904
criterion	dt	seasonal	entropy	0.788398
max_depth	dt	seasonal	9	0.788398
max_features	dt	seasonal	auto	0.788398
ccp_alpha	dt	seasonal	0.001	0.788398
C	lr	seasonal	0.5	0.774605
solver	lr	seasonal	saga	0.774605
penalty	lr	seasonal	l1	0.774605
multi_class	lr	seasonal	auto	0.774605

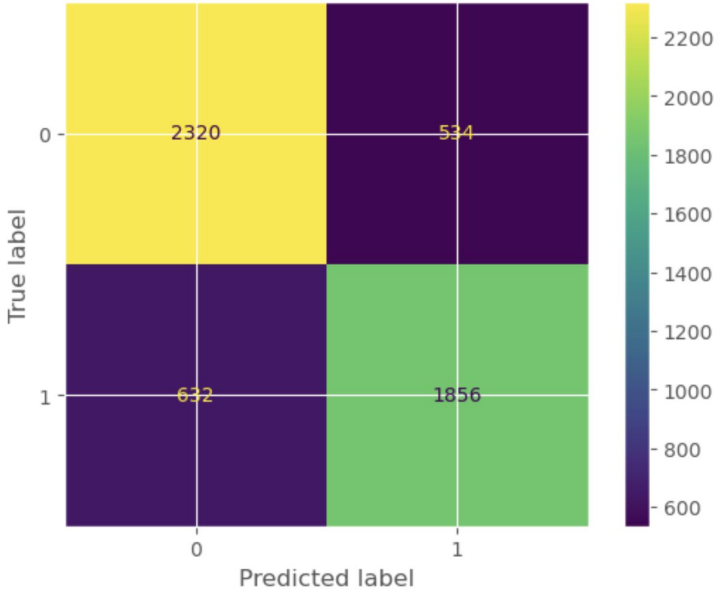
*El score más alto es **0.855847** con SVM balanced.*

IMMUNE CODING INSTITUTE

Training Seasonal - Round 1

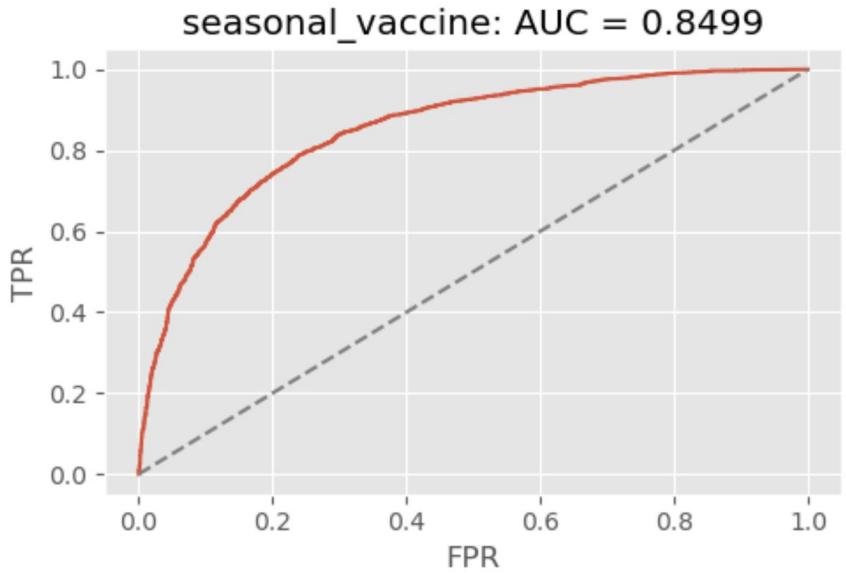
best estimator	C	gamma	kernel	class_weight	score
SVM	1	0.01	rbf	balanced	**0.855847**

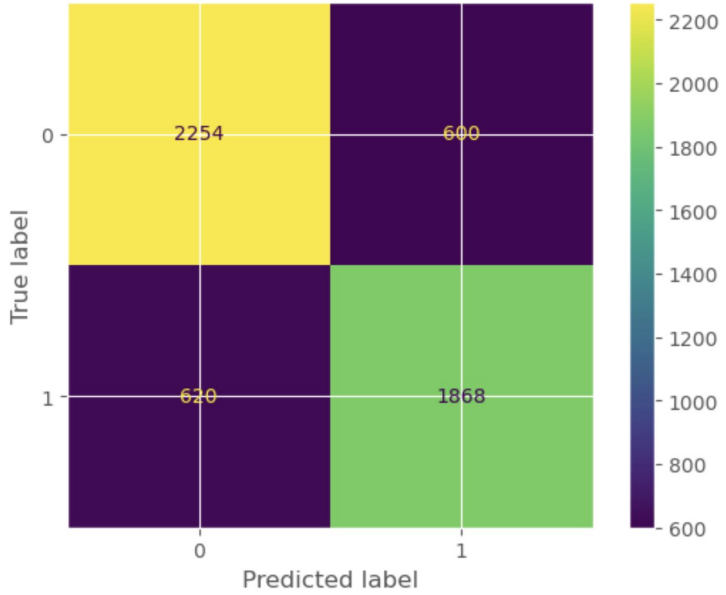




Training Seasonal - Round 1

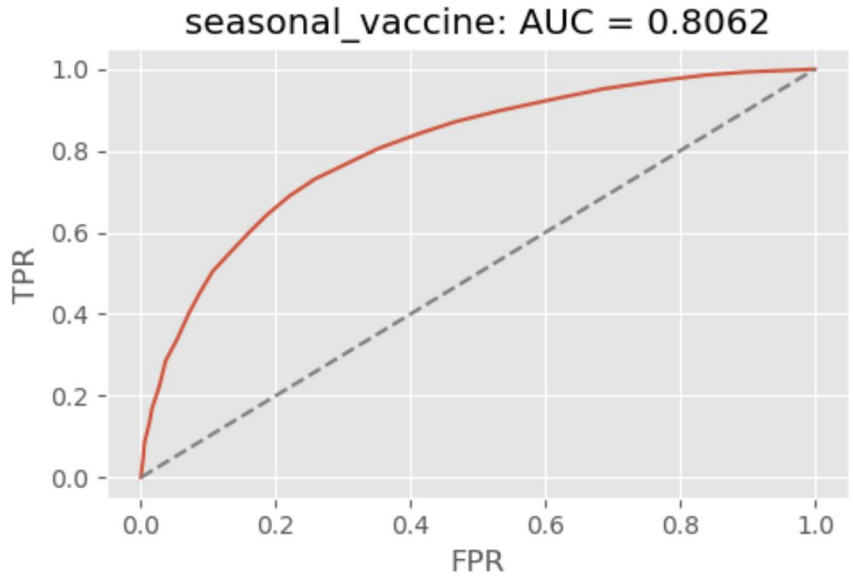
estimator	criterion	max_depth	max_features	n_estimators	class_weight	auc
Random Forest	entropy	8	auto	500	balanced	0.849939

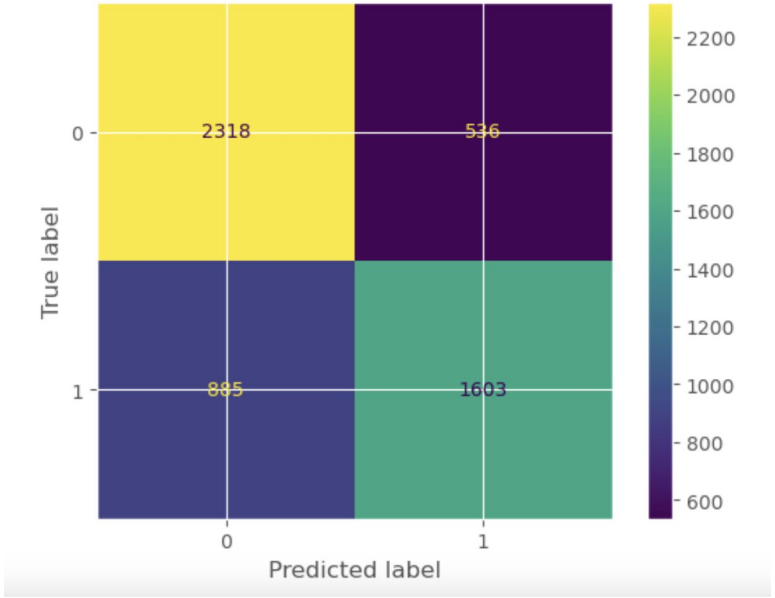




Training Seasonal - Round 1

best estimator	n_neighbors	score
K-Neighbors	30	0.806208





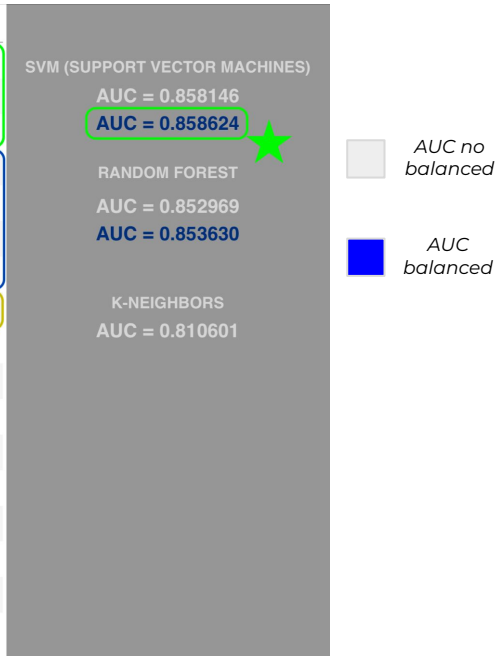
IMMUNE CODING INSTITUTE

Training Seasonal - Round 2

Ejecución de Hyperparameters Tuning con GridSearchCV y training, round 1 con split 70%-30%.
Con y sin el parámetro `class_weight='balanced'`.

FICHA TÉCNICA	
Tool	GridSearchCV
Scoring	roc_auc
Algoritmos evaluados	1. Logistic Regression 2. Random Forest 3. SVM (Support Vector Machines) 4. Naive Bayes 5. Decision Tree 6. K-Neighbors
Split training and evaluation set	70%-30%
Top 3	1. SVM (Support Vector Machines) 2. Random Forest 3. K-Neighbors

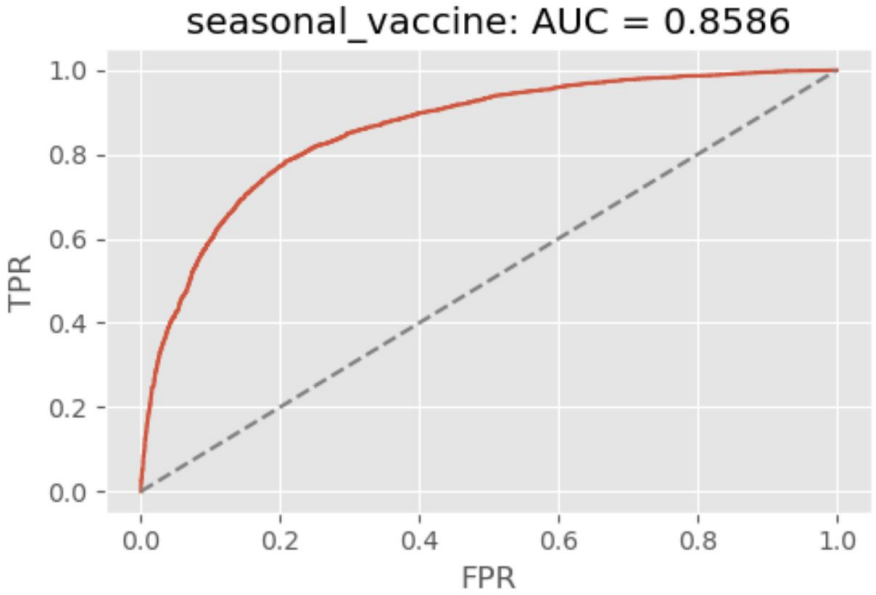
	estimator	target	params	score
C	svm	seasonal	1	0.854065
gamma	svm	seasonal	0.01	0.854065
kernel	svm	seasonal	rbf	0.854065
n_estimators	rf	seasonal	500	0.849205
max_features	rf	seasonal	auto	0.849205
max_depth	rf	seasonal	8	0.849205
criterion	rf	seasonal	gini	0.849205
n_neighbors	kn	seasonal	30	0.821667
var_smoothing	naive	seasonal	0.053367	0.816363
criterion	dt	seasonal	entropy	0.792845
max_depth	dt	seasonal	8	0.792845
max_features	dt	seasonal	auto	0.792845
ccp_alpha	dt	seasonal	0.001	0.792845
C	lr	seasonal	0.5	0.775169
solver	lr	seasonal	saga	0.775169
penalty	lr	seasonal	l1	0.775169
multi_class	lr	seasonal	auto	0.775169

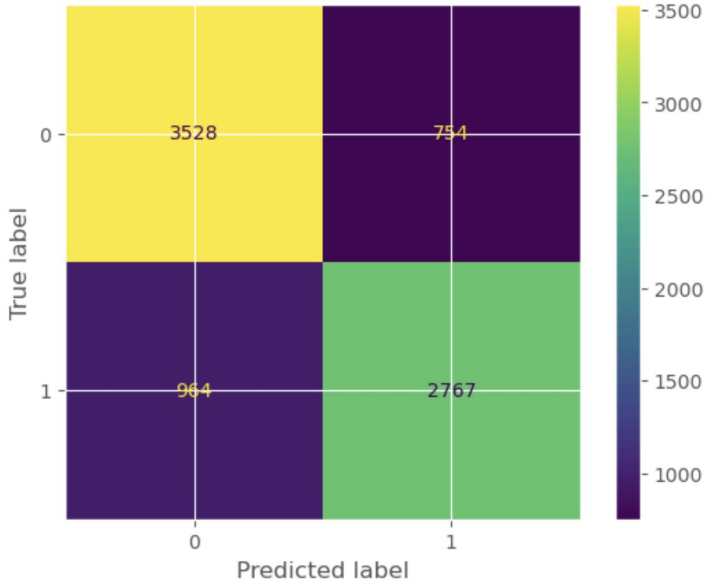


*El score más alto es **0.858624** con SVM.*

Training Seasonal - Round 2

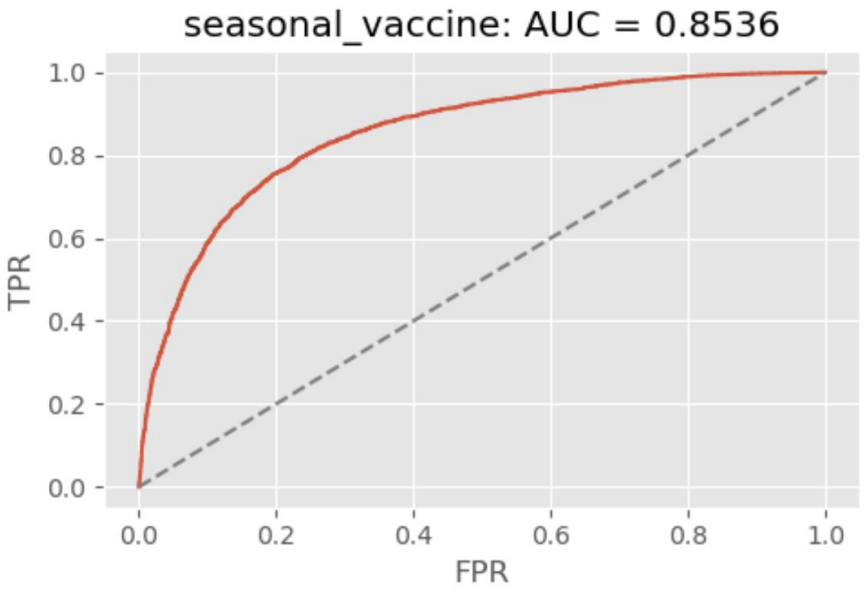
best estimator	C	gamma	kernel	class_weight	score
SVM	1	0.01	rbf	balanced	**0.858624**

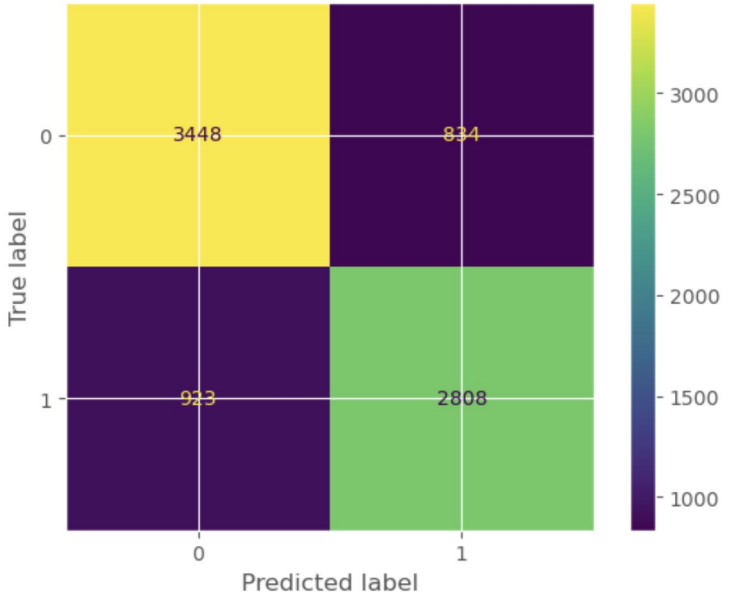




Training Seasonal - Round 2

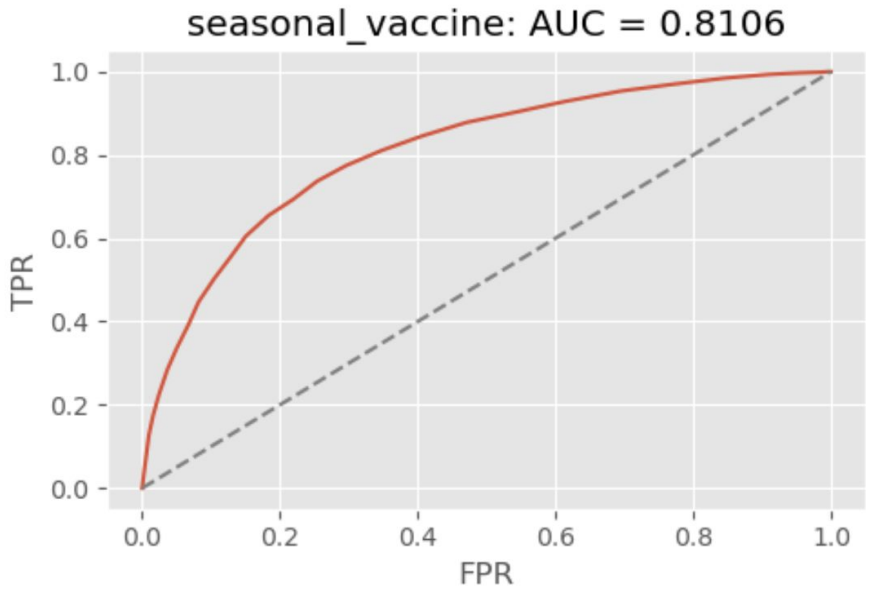
estimator	criterion	max_depth	max_features	n_estimators	class_weight	auc
Random Forest	entropy	8	auto	500	balanced	0.853630

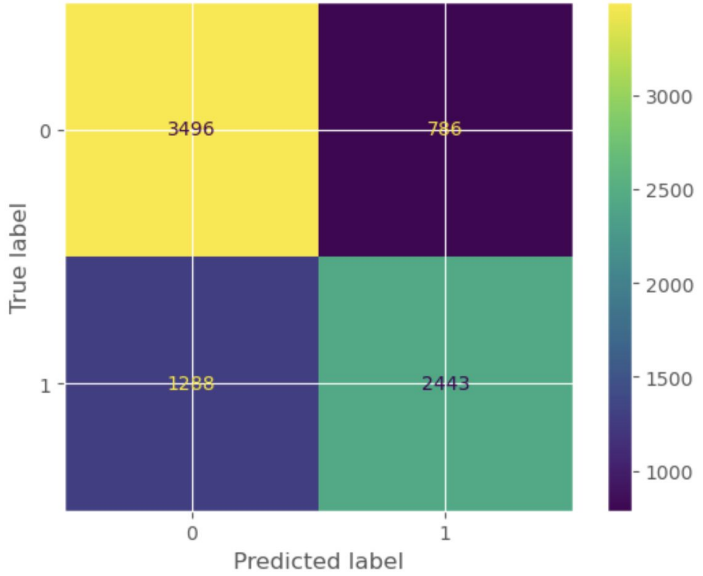




Training Seasonal - Round 2

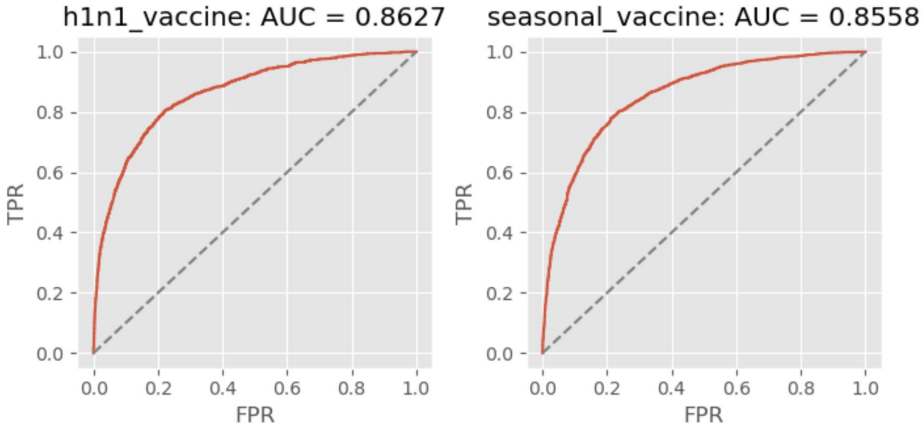
best estimator	n_neighbors	score
K-Neighbors	30	0.810601





OTROS ROUNDS DE ENTRENAMIENTO

Training MultiOutputClassifier - SVM



Ejecución de un round usando multi target classification con `MultiOutputClassifier`.

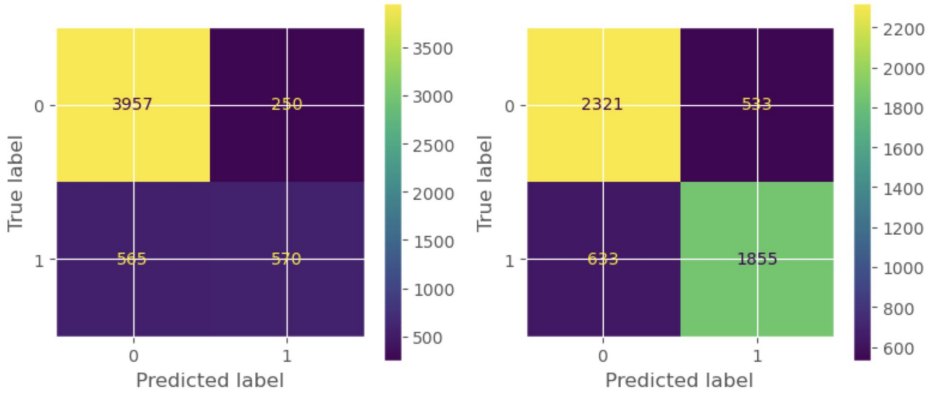
Con el mejor algoritmo para las 2 variables target H1N1 y seasonal, con split 80%-20%:

- `SMV: C=1, gamma=0.01, kernel='rbf'`

Aplicando el parámetro:
`class_weight='balanced'`

Obtenemos los 2 mejores valores de AUC:

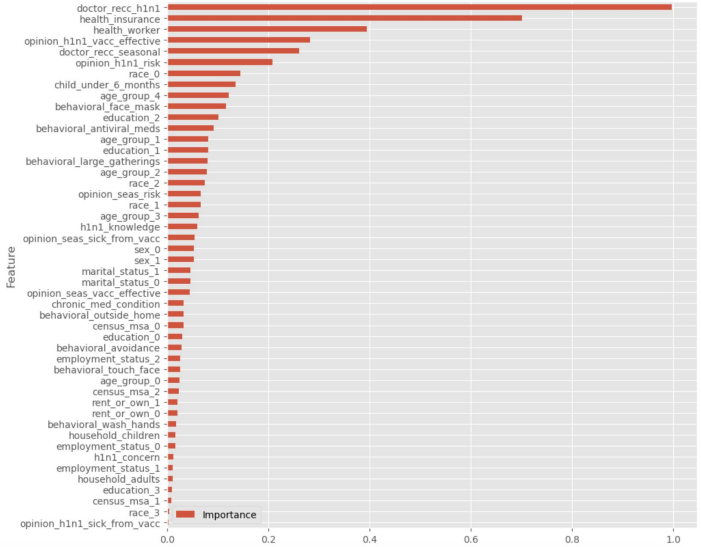
- H1N1 AUC = **0.862736**
- Seasonal AUC = **0.855842**



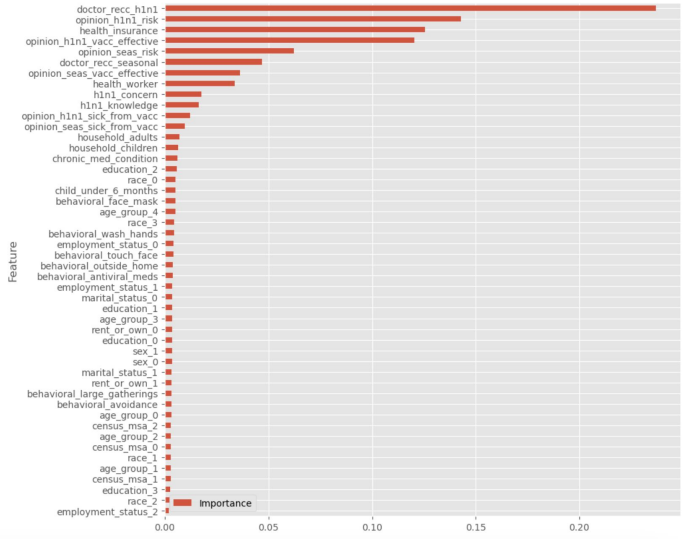
IMMUNE CODING INSTITUTE

Training H1N1 / Seasonal - Features Importances

En otro round de ejecución, evaluamos `feature_importances_` para varios de los algoritmos, extrayendo un subset con las variables más importantes, pero sin lograr valores más altos a los ya obtenidos.



Logistic Regression: Subset de 24/49 variables con un % de importancia sobre el 5%, no obtuvimos valores superiores.

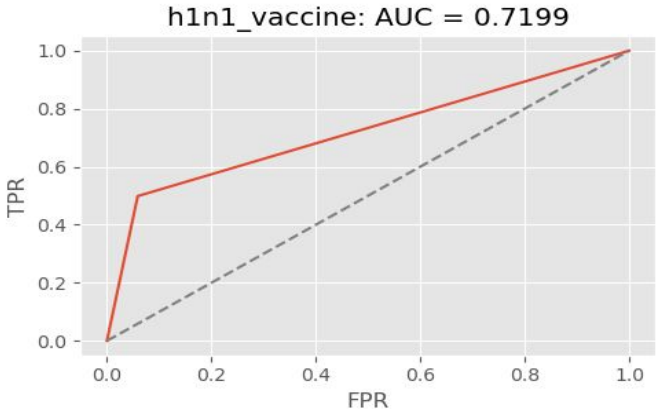


Random Forest: Subset de 12 variables con un % de importancia considerable, no obtuvimos valores superiores.

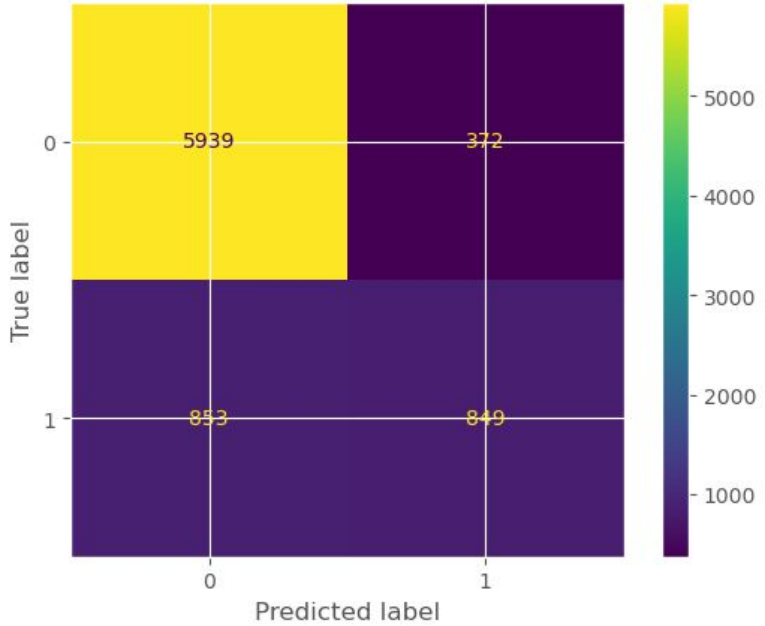
Training H1N1 / Red Neuronal Dense - Round 1

Hyperparameters Tuning (GridSearchCV)			
Activation	**Hidden_Units**	**Optimizer**	**Score**
relu	16	adam	0.7198
relu	64	adam	0.7195
tanh	16	adam	0.72
tanh	32	sgd	0.7157
relu	32	adam	0.7239

En otro round de ejecución entrenamos una red neuronal dense para la variable H1N1, con split 70%-30%.



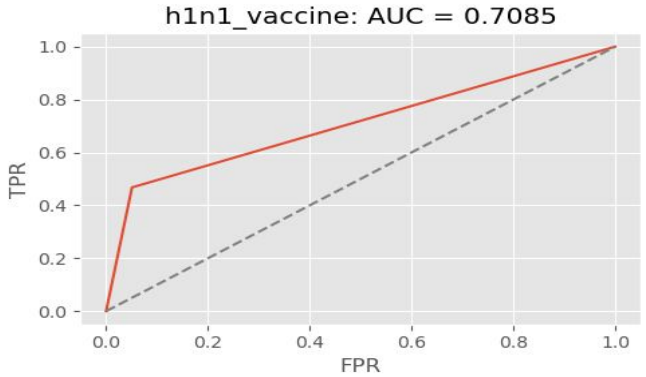
AUC en training real



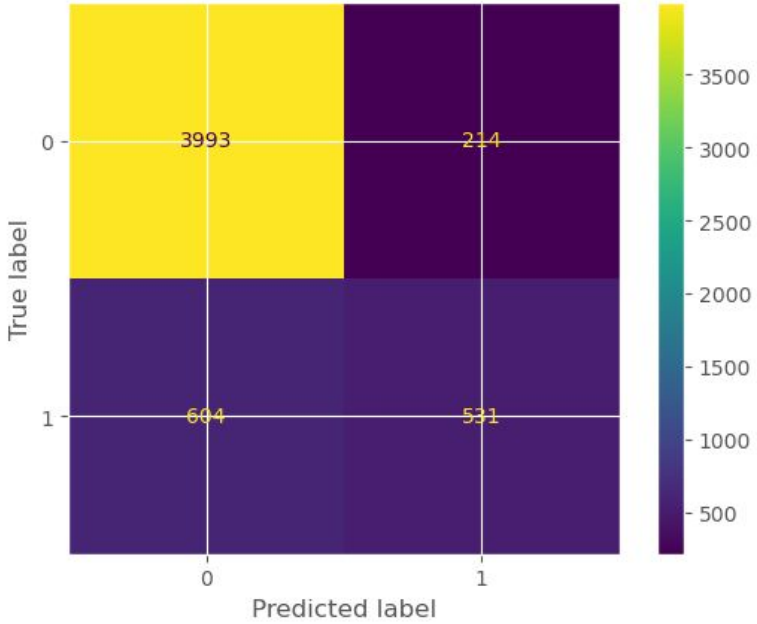
Training H1N1 / Red Neuronal Dense - Round 2

Hyperparameters Tuning (GridSearchCV)			
Activation	**Hidden_Units**	**Optimizer**	**Score**
relu	16	sgd	0.7162
relu	32	adam	0.7196
tanh	16	adam	0.7148
tanh	32	sgd	0.7161
relu	16	adam	0.7202

En otro round de ejecución entrenamos una red neuronal dense para la variable target H1N1, con split 80%-20%.



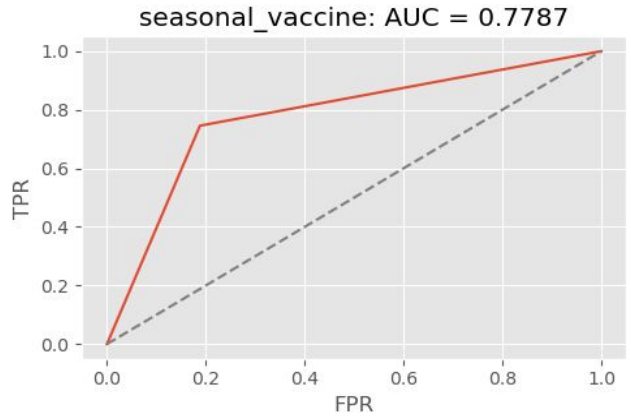
AUC en training real



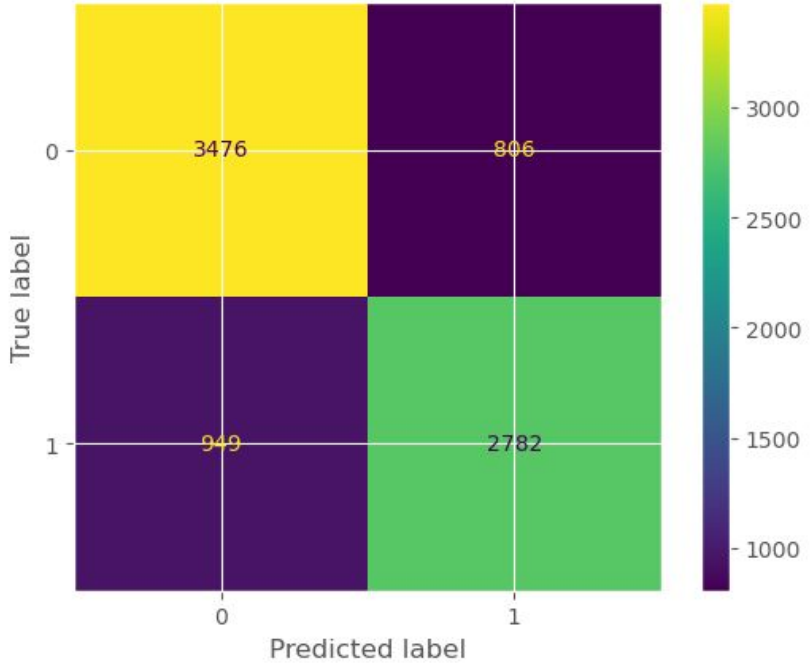
Training Seasonal / Red Neuronal Dense - Round 1

Hyperparameters Tuning (GridSearchCV)			
Activation	**Hidden_Units**	**Optimizer**	**Score**
relu	16	adam	0.7738
relu	32	adam	0.7707
tanh	16	adam	0.7701
tanh	64	sgd	0.7738
tanh	64	sgd	0.7711

En otro round de ejecución entrenamos una red neuronal dense para la variable target Seasonal, con split 70%-30%.



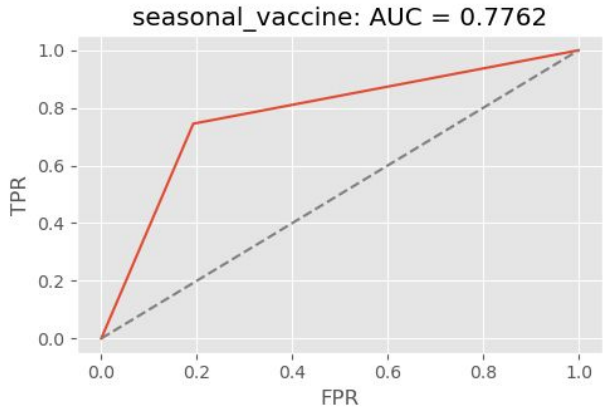
AUC en training real



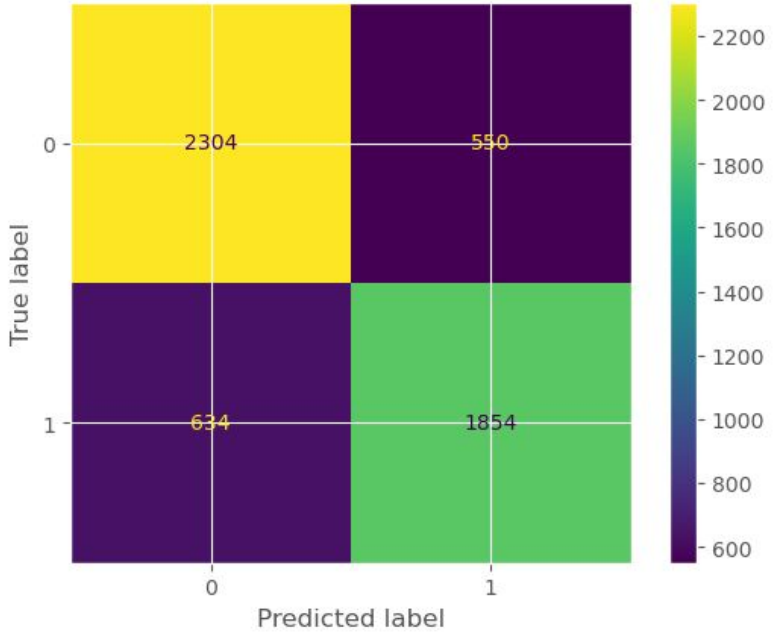
Training Seasonal / Red Neuronal Dense - Round 2

Hyperparameters Tuning (GridSearchCV)			
Activation	**Hidden_Units**	**Optimizer**	**Score**
relu	16	adam	0.7738
relu	32	adam	0.7733
tanh	16	adam	0.7690
tanh	64	sgd	0.7722
relu	16	adam	0.7738

En otro round de ejecución entrenamos una red neuronal dense para la variable target Seasonal, con split 80%-20%.



AUC en training real



Conclusiones

- La fase de EDA garantiza que el dataset se encuentre en condiciones óptimas para el proceso de hyperparameters tuning y training posterior, por lo que en esta fase, a través de un análisis de correlación, eliminamos algunas variables no relevantes, convertimos las variables categóricas en numéricas, y en general eliminamos los valores nulos del dataset.

- El proceso de hyperparameters tuning es un proceso exhaustivo que requiere rounds de pruebas de ejecución donde se puede incluir una configuración amplia de parámetros por algoritmo, así como configurar el tipo de métrica a generar, y técnicas de cross-validation para dividir el set de training en sets más pequeños durante el entrenamiento y así evitar el overfitting durante el proceso de tuning.

- **SVM(C=1, gamma=0.01, kernel=RBF)** resultó ser el algoritmo que produjo los mayores auc_scores (H1N1=0.862754 - split 80/20, seasonal=0.858624 - split 70/30), posiblemente debido a que una de sus ventajas es el ser robusto ante la presencia de outliers, y en este caso se detectaron grandes cantidades de estos valores atípicos. Así mismo, al hacer uso del `class_weight=balanced`, se mejoró el score, lo anterior como una forma de disminuir el desbalance identificado en la variable target “h1n1_vaccine”.

- Diversas configuraciones de split para el set de training versus el de evaluación, generaron diferentes resultados. Un split 80%-20% generó un mejor score para el caso de la variable target H1N1, y para el caso de la variable target Seasonal, el mejor score se logró con un split 70%-30%.

Conclusiones

- K-Neighbors fue un clasificador en el top 3 de entrenamiento de la variable target Seasonal, que no presenta outliers, pero no en la variable H1N1. K-Neighbors se comporta mejor ante la ausencia de ellos, dado que es un algoritmo basado en distancias.
- Para varios de los algoritmos identificamos los porcentajes de importancia de las variables usadas por el algoritmo al momento de training, y seleccionamos un subset de variables del dataset para nuevos rounds de entrenamiento, que si bien dieron buenos resultados, fueron inferiores a los highest scores ya logrados con SVM.
- Aplicado a un caso de clasificación, realizamos varios rounds de ejecución, con split 80%-20% y 70%-30%, con redes neuronales dense con varios parámetros, con buenos resultados, pero inferiores a los highest scores ya logrados con SVM.

Trabajo Futuro

- Explorar otras técnicas para el manejo de desbalance en la variable target H1N1 como Oversampling (SMOTE), Undersampling, o enfoques de Ensemble (Bagging ó Boosting para mejorar la predicción de la clase minoritaria).
- Explorar técnicas adicionales para el manejo de outliers, como Winsorizing, que reemplaza los valores extremos con valores cercanos a los límites de un rango específico.
- Aplicar técnicas de clusterización para un análisis más profundo de la distribución de los datos.
- Explorar otros clasificadores como XGBoost ó StackingClassifier.

¡ Muchas gracias !