Clasificación de cantos de anuros utilizando técnicas de machine learning. Modelos de simulación y sistemas II.

Juan David Cano Gómez, Julián Andrés Castaño Jiménez

Resumen – En este trabajo se presenta la utilización del conocimiento adquirido en este curso enfocado en la clasificación de una base de datos que contiene valores numéricos que están relacionados con los cantos de anuros. Los datos de cantos de anuros son extraídos de una base de datos de coeficientes denominados MFCC, se entrena los datos y se ejecutan las pruebas para realizar una asignación adecuada de un determinado canto a su anuro correspondiente. Durante el transcurso de la ejecución del script que realiza la clasificación se revisa constantemente las métricas de error y se procura analizar lo sucedido con los modelos.

Índice de Términos – MFCC, machine learning, classification algorithms, feature extraction.

Las palabras clave fueron consultadas en el "thesaurus" de términos de IEEE, MFCC refiere a Mel Frequency Cepstral Coefficients. Si se nos permitiera poner dos palabras claves adicionales después de machine learning, classification algorithms y feature extraction serían: **Anuran Calls**.

Estos descriptores "Anuran Calls" no se encuentran en el "thesaurus" de IEEE, son relativamente nuevos, pero en los buscadores usados y las páginas científicas se dirige de manera directa a los problemas de clasificación de cantos de anuros, que es de lo que se habla en este trabajo.

Un anuro puede ser tanto una rana o un sapo.

I. Comprensión de los cantos de anuros abordado por medio de machine learning

En el ámbito biológico el estudio de algunas especies es algo que se procura hacer con el cuidado de no intervenir con actividades antropogénicas para mantener el equilibrio de la vida. Los investigadores que se han dedicado al estudio de los anuros han encontrado en las señales bio acústicas una alternativa que mantiene los factores de estudio en sus condiciones naturales, con ello se podrían saber qué sucede con determinadas especies en cuanto crecimiento, decrecimiento, con la salud en general de ellas o con cualquier factor de estudio que la biología considere sin tener que invadir sus hábitats.

Para construir la base de datos de información de los cantos de anuros, los investigadores recolectaron información de audio de cantos de diferentes anuros provenientes de diversos lugares, dicha información se puede guardar en formatos digitales como lo son audio tipo WAV. Los audios fueron grabados en condiciones naturales, se procedió a eliminar el ruido ambiente, y a procesar los audios para hacer una clasificación de los sonidos característicos asignando a cada dato recolectado la especie reconocida por expertos, al audio se le hace un procesamiento con unas funciones de transformación determinada y finalmente se obtiene de cada fragmento recolectado y preclasificado un vector de coeficientes. El almacenamiento de dicha información se crece con cada nuevo espécimen y se almacena por medio de una matriz que contiene los coeficientes característicos de esos datos y la clasificación correspondiente.

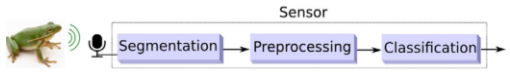


Figura 1. Pre-Clasificación - Pasos básicos en el marco de identificación de especies de anuros [1].

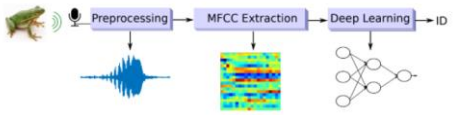


Figura 2. Esquema general de preclasificación propuesto por los autores que utiliza técnicas de machine leraning y deep learning para la extracción de los coeficientes MFCC [2].

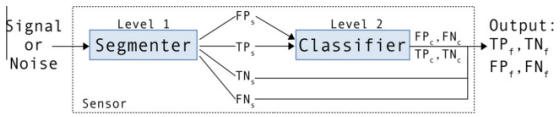


Figura 3. Preclasificación - Esquema de interacción entre el segmentador y el clasificador, solo reconoce segmentos enviados al clasificador de especies [3].

Documento enviado 13 de septiembre de 2022 para revisión.
Juan David Cano Gómez, estudiante de ingeniería de sistemas Universidad de Antioquia, Medellín, Colombia (e-mail: juan.cano3@udea.edu.co).

Julián Andrés Castaño Jiménez, estudiante de ingeniería de sistemas Universidad de Antioquia, Medellín, Colombia
(e-mail: julian.castanoj@udea.edu.co)

Para diferenciar mejor **preclasificación** y **clasificación se** establece para este trabajo que la **preclasificación** [1]–[3] tiene que ver con todo proceso previo a la obtención de la base de datos que contiene la matriz de MFCC con sus respectivas asignaciones de familia, especie, género como se aprecia en la ***Figuras 1,2 y 3***, es decir, todo lo previo a la obtención del archivo *.csv que se procesó para esta entrega de proyecto, mientras que la palabra clasificación será utilizada directamente para todo aquello que tenga que ver con procesos que se hayan realizado para este trabajo.

Acerca de la base de datos Anuran Calls de UCI:
Los registros fueron recolectados en ambientes reales con condiciones de ruido en el ambiente. Algunos anuros son de la Universidad Federal do amazonas, Manaus, Mata Atlántica do Brasil y desde Córdoba-Argentina. Las grabaciones se realizaron en formato WAV con 44.1 kHz de frecuencia de muestreo, 32 bit de resolución, esto permite analizar señales hasta de 22kHz. Se calcularon 22 MFCC, utilizaron 44 filtros triangulares normalizados entre -1 y 1 unidades. Los MFCC colectivamente forman MFC. Cada sílaba contiene una longitud diferente, se normaliza de acuerdo con ***MFCCs_i/(max(abs(MFCCs_i)))***[4]

1) Descripción del problema

El problema de predicción que se aborda consiste en realizar una clasificación de anuros basado en una base de datos de registros de audio de sus cantos. Las especies que fueron analizadas y estudiadas corresponden a especies en Brasil y Argentina. Este problema es aplicado en campos biológicos, más específicamente en campos bio-acústicos donde se desea conocer a qué clase pertenece un canto específico, en este caso, las clases son las especies de anuros, partiendo de las características que se pueden extraer de los sonidos que ellos emiten. Como se mencionó anteriormente, este problema corresponde a un problema de clasificación. Las variables a predecir son de tipo categórica, se hace una transformación de esas salidas para retornar un valor numérico.

2) Variables del sistema

Variables de entrada	*Variables de salida*
Datos de sonido (MFCC)	*Familia*
	Genero
	Especie

Variables de entrada: Datos de sonido en formato MFCC

Un MFCC se define por sus siglas en inglés como Mel Frequency Cepstral Coefficients, es decir, esta información contiene coeficientes de determinados espectros de sonido en términos de las frecuencias tipo Mel o en escala Mel.

Inicialmente los investigadores capturaron registros de audio en lugares donde encontrarían anuros, el espectro es convertido en coeficientes que contienen información relevante acerca de una comunicación oral determinada, cada sección puede corresponder a frases, cada frase descomponerse en palabras y cada palabra en sílabas si así se quiere, dependiendo el estudio que se esté realizando. Las aplicaciones de estos coeficientes podrían ser reconocimiento de voz, traducción de audio a texto o identificación de especies como es el caso de este proyecto.

En este problema, los MFCC contienen información seccionada que se introduce en un vector de características, cada característica permite la identificación de las especies, esas características las consideramos variables de entrada.[5]

Variables de salida: Familia, Género, Especie.

Las **familias** pueden ser:
Bufonidae 68, Dendrobatidae 542, Hylidae 2165, Leptodactylidae, 4420.

Los **géneros** pueden ser:
Adenomera 4150, Ameerega 542, Dendropsophus 310, Hypsiboas 1593, Leptodactylus 270, Osteocephalus 114, Rhinella 68, Scinax 148.

Las **especies** pueden ser:
Adenomera, Andre 672, Adenomera, Hylaedacta| 3478, Ameeregatrivittata 542, HylaMinuta 310, Hypsiboas, Cinerascens 472, Hypsiboas, Cordobae 1121, LeptodactylusFuscus 270, OsteocephalusOopha 114, Rhinellagranulosa 68, ScinaxRuber 148.

Según la categoría taxonómica se puede apreciar que. Especie es un subconjunto de género y género un subconjunto de familia, por lo cual se determinó que la clasificación se hace en este trabajo con **especie** como se aprecia en la ***Figura 5***.

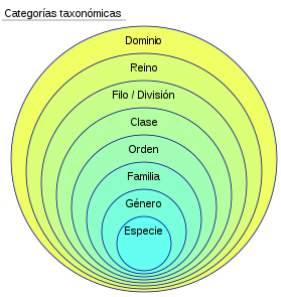


Figura 5. Categorías taxonómicas [6].

En cuanto a la codificación de los datos de entrada: en la base de datos existe una nota al pie de página que especifica que el audio se descompuso en un número determinado de partes, en cuyo caso, según la tabla de la base de datos, cada espectro se descompuso en 22 partes y estas partes fueron normalizadas.

En cuanto a la codificación de los datos de salida: el número que le sigue al nombre está asociado directamente con la clasificación.

La clasificación de las ranas en familia, género y especie corresponde a sus nombres científicos de la familia, el género y especie.

En este caso no existen datos faltantes, la base de datos específica en la sección de datos faltantes; N/A, esto significa que no aplica para este caso y ello puede deberse a que se procesó audios y fueron tomados por completo para la clasificación.

3) Estado del arte

Usualmente los problemas de clasificación de este tipo suelen ser abordados desde tres perspectivas, clasificar el audio completo sin segmentar, usar un tamaño de segmentación fija por tramos o hacer la clasificación por silabas como se muestra en la ***Figura 4***. Colonna y Colaboradores en su artículo de 2016 abordan el problema desde la clasificación por silabas del mismo tamaño, para ellos este tipo de clasificación es eficiente pues hacen una selección solo de segmentos que contienen información de audio y desechan tramos vacíos, [7]

Autor	#	ML	Acc	Autor	#	ML	Acc
Colonna	9	kNN,SVM	97%	Dayou	9	kNN	90%
Huang	5	kNN,SVM	100%	Han	9	kNN	100%
Jafar	28	kNN,SVM	98%	Vaca	20	kNN	91%
Xie	4	GMM	90%	Yuan	8	kNN	98%

Tabla 1. *Resumen de resultados de clasificación según los autores del estado del arte teniendo en cuenta # de especies, técnica utilizada y la exactitud* [7].

Los diferentes autores utilizaron máquinas de vectores de soporte (SVM), método de los k vecinos más cercanos (k-NN) y modelo de mezcla de gaussianas (GMM) para hacer sus respectivas clasificaciones como se aprecia en la ***Tabla 2***, en general se obtiene exactitudes (accuracy) por encima del 90%, cada autor toma decisiones respecto a qué muestras deben tomarse en cuenta para el entrenamiento y para la clasificación pero todos coinciden que al ser un problema supervisado, la introducción de nuevos datos al sistema puede hacer que se disminuya el porcentaje de exactitud de los modelos y se deba volver a entrenar el modelo.

Colonna y Colaboradores, sostienen que para mejorar la capacidad de generalización del sistema y evitar que suceda el problema de disminución de porcentaje de exactitud en la medida que se ingresen nuevas muestras al modelo se aplica la práctica de implementar la metodología de validación cruzada (k-CV), más no una validación cruzada clásica sino con una modificación, en la modificación se debe evitar que una división aleatoria contenga silabas del mismo espécimen dentro de dos subconjuntos diferentes para entrenamiento y prueba, proponen una evaluación Leave-one-out de validación cruzada por cada registro de individuo para medir el desempeño de los algoritmos de clasificación, con k igual al número de diferentes especímenes, así una silaba puede tener dos etiquetas una asignada al espécimen o registro de grabación y otra a su especie, se repite los pasos hasta que cada registro haya sido usado en un conjunto de prueba, de esta forma su pregunta de investigación se concentra en resolver en que tan bueno es su modelo para especímenes que aún no han sido ingresados al sistema [7]

Species	kNN $k=1$	kNN $k=3$	kNN $k=5$	Tree	QDA	SVM RBF	SVM $p=1$	SVM $p=2$	SVM $p=3$
Adenomera andreae	33.46	32.66	34.67	30.64	86.69	72.58	59.27	74.79	72.98
Ameerega trivittata	89.88	89.33	88.23	42.83	88.60	67.46	57.90	64.52	70.77
Adenomera hylaedactyla	98.68	99.37	99.50	94.29	98.29	99.77	99.77	99.73	99.60
Hyla minuta	61.57	53.71	53.27	34.49	52.40	55.02	25.32	55.02	65.06
Hypsiboas cinerascens	96.39	98.06	96.95	71.74	90.02	93.35	90.02	96.67	97.50
Hypsiboas cordobae	100.00	100.00	100.00	98.29	95.86	98.29	96.43	97.86	98.57
Leptodactylus fuscus	63.96	59.90	49.09	9.00	0.45	9.45	0.45	70.27	57.20
Osteocephalus oophagus	42.70	34.37	32.29	17.70	11.45	0.00	0.00	0.00	9.37
Rhinella granulosa	39.84	32.81	30.46	9.37	0.78	28.12	12.50	37.50	45.31
Scinax ruber	0.00	0.00	0.00	3.94	0.00	0.00	0.00	3.94	1.31
Micro-accuracy	86.21	85.80	85.36	73.52	85.38	84.34	80.09	86.93	**87.61**
Average-accuracy	**62.65**	**60.02**	**58.45**	41.23	**52.45**	**52.40**	**44.16**	**60.03**	**61.77**
Precision	0.62	0.60	0.59	0.53	0.65	0.66	0.63	0.72	0.70
Recall	0.63	0.60	0.58	0.41	0.52	0.52	0.44	0.60	0.62

Tabla 2. *Comparación de resultados utilizando la estrategia de descomposición una vs todas [7].*

Species	kNN $k=1$	kNN $k=3$	kNN $k=5$	Tree	QDA	SVM RBF	SVM $p=1$	SVM $p=2$	SVM $p=3$
Adenomera andreae	33.46	32.05	31.45	26.00	26.61	31.85	28.42	28.62	30.24
Ameerega trivittata	89.88	89.70	88.97	70.40	99.26	92.83	91.36	78.86	63.78
Adenomera hylaedactyla	98.68	99.37	99.50	98.19	98.49	99.86	99.96	99.77	99.34
Hyla minuta	61.57	53.71	53.27	58.07	84.27	61.57	62.44	66.81	68.99
Hypsiboas cinerascens	96.39	98.06	97.22	88.36	88.64	97.22	96.95	96.12	94.18
Hypsiboas cordobae	100.00	100.00	100.00	95.58	95.72	99.85	99.00	99.71	100.00
Leptodactylus fuscus	63.96	59.90	50.90	45.49	1.351	59.00	36.93	67.56	62.16
Osteocephalus oophagus	42.70	36.45	34.37	20.83	15.62	6.25	1.04	14.58	36.45
Rhinella granulosa	39.84	33.59	33.59	17.96	1.56	31.25	28.12	32.81	46.87
Scinax ruber	0.00	0.00	0.00	0.00	0.00	9.21	18.42	23.68	32.89
Micro-accuracy	**86.21**	85.83	85.34	80.85	82.66	86.14	84.82	85.32	84.43
Average-accuracy	**62.65**	**60.28**	**58.93**	52.09	**51.15**	**58.89**	**56.26**	**60.85**	**63.49**
Precision	0.62	0.61	0.60	0.57	0.53	0.67	0.63	0.68	0.70
Recall	0.63	0.60	0.59	0.52	0.51	0.59	0.56	0.60	0.63

Tabla 3. *comparación de resultados utilizando la estrategia de descomposición uno a uno Haga clic o pulse aquí para escribir texto.*[7]

	kNN $k=1$	kNN $k=3$	kNN $k=5$	Tree	QDA	SVM RBF	SVM $p=1$	SVM $p=2$	SVM $p=3$
Gains	0.00	**+0.26**	**+0.48**	**+10.86**	−1.30	**+6.49**	**+12.10**	**+0.82**	**+1.72**

Tabla 4. *Comparación de 1a1 sobre 1vstodas[7].*

Las **tablas 3 a 4** arrojan los resultados de haber aplicado Cross Validación teniendo la consideración de clasificar por especies. Las tablas comparan los resultados teniendo en cuenta los KNN, el análisis de discriminante cuadrático (QDA), arboles de decisión, maquinas de soporte con kernel RBF y polinomiales de grados 1,2 y 3, en la tabla se aprecia un campo denominado Micro-accuracy que corresponde al promedio de las accuracy por especies. Se inicia con 52.40% para micro accuracy y en 10% para el accuracy medio. Según el modelo, el valor de referencia para un clasificador que siempre elige las especies más números micro y elige el medio en el caso que sea

aleatorio, utilizaron un estadístico t y el nivel de confianza es de p=0.05.

II. ENTRENAMIENTO Y EVALUACIÓN DE LOS MODELOS

4) Experimentos

4.1 Metodología de Validación

Debido a que el problema que se está abordando es un problema desbalanceado, porque existen clases que tienen mayor cantidad de muestra que otras clases en las diferentes salidas que son familia, género y especie, es importante utilizar una metodología de validación que tenga en cuenta todas las clases y que no se sesgue a la clase de mayor cantidad de muestras, es por esta razón que se utilizó la metodología de validación *Stratified k-fold*, que es una versión estratificada de la metodología de validación *Cross Validation k-fold*.

4.2 Base de datos para el proyecto

La base de datos para llevar a cabo el proyecto, es una base de datos del sitio web *UCI Machine Learning Repository* que tiene como nombre *Anuran Calls*. Como se mencionó anteriormente, los datos de la base de datos usada fueron extraídos de los hábitats de los anuros en condiciones reales de ruido y utilizando la bioacústica para tal fin, ya que el campo de estudio es la biología y se están procesando los cantos de los anuros con el fin de realizar posteriores clasificaciones acerca de la taxonomía de dichos animales. Con el fin de procesar los sonidos de las ranas, se usaron artefactos especiales para tal tarea y los resultados fueron consignados en dicha base de datos. [4]

4.3 Número de muestras y variables

El número de muestras es de 7195 y el número de variables es de 22 que corresponden a los MFCCs de los cantos de los anuros.

4.4 Distribución de muestras por clase

La base de datos de los cantos de ranas cuenta con 3 variables de salida, que son: familia, género y especie. La primera salida, familia, tiene un total de 4 clases que son *Bufonidae, Dendrobatidae, Hylidae, y Leptodactylidae*, con 68, 542, 2165, y 4420 muestras respectivamente. La salida género cuenta con 8 géneros o clases de género que son *Adenomera, Ammerega, Dendropsophus, Hypsiboas, Leptodactylus, Osteocephalus, Rhinella, y Scinax*, con 4150, 542, 310, 1593, 270, 114, 68 y 148 muestras respectivamente, y la salida de especies tiene 10 clases que son *AdenomeraAndre, AdenomeraHilaedatylus, Ameeregatrivittata, HylaMinuta, HypsiboasCinerascens, HypsiboasCordobae, LeptodactylusFuscus, OsteocephalusOophagus, Rhinellagranlosa, ScinaxRuber* con 672, 3478, 542, 310, 472, 1121, 270, 114, 68 y 148 muestras respectivamente. Acá se puede notar fácilmente el desbalance de clases por familia, género y especie. La distribución de clases vista gráficamente es:

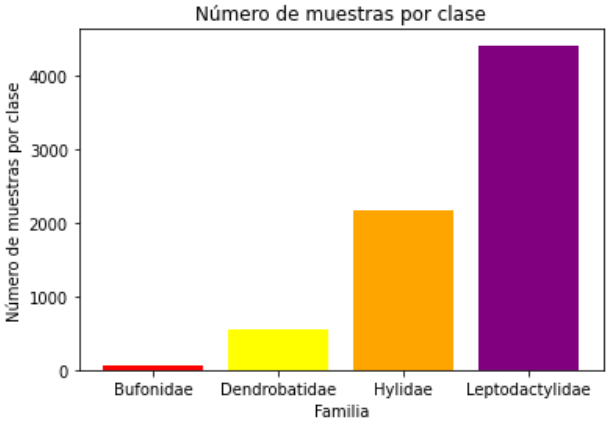


Figura 5. *Diagrama de barras clasificado por familia.*

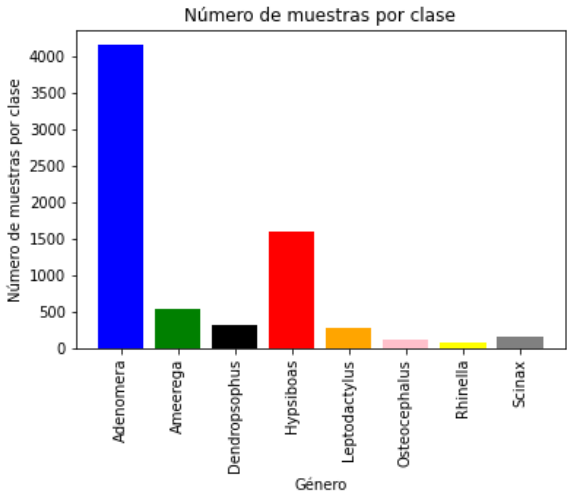


Figura 6. *Diagrama de barras clasificado por género.*

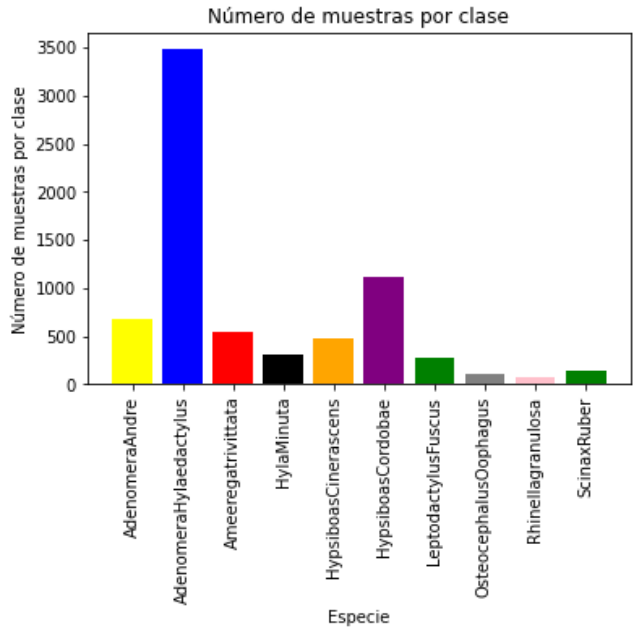


Figura 7. *Diagrama de barras clasificado por especie.*

Haciendo un análisis de las tres variables de salida implicadas, familia, género y especie, se tomó la decisión de trabajar con una sola de las variables, la variable especies, que es una variable que pertenece al campo de la biología y en este existen clasificaciones taxonómicas que permiten establecer una alta dependencia entre las tres variables, por lo que a través de la clasificación se encuentra en el nivel más bajo de jerarquía la variable especies, y se puede obtener de manera indirecta las restantes dos variables que se encuentran en los dos niveles superiores de la jerarquía. El orden de dicha jerarquía es el de género, familia y especie.

Como se mencionó anteriormente, la base datos del problema está desbalanceada, esto es un problema debido a que los modelos de *Machine Learning* no tienen en cuenta todas las clases. Es por esto que se recurre al uso de técnicas de submuestreo inteligente que son técnicas especiales para compensar este tipo de problemas. En este trabajo se utilizó la técnica de eliminación de muestras de la clase mayoritaria, ya que esta clase presenta un serio problema de desbalance. Para ello se utilizó el método de la librería *Imbalanced Learn* de *Python* que se llama *EditedNearestNeighbours* que es un método que está basado en el método de los vecinos más cercanos. Lo que hace esta función es que limpia la base de datos removiendo muestras cercanas a la frontera de decisión que se le establezca. Para cada muestra de la clase que se va a sub muestrear, se calculan los vecinos más cercanos y si no se cumple el criterio de selección se elimina la muestra, dicho criterio es que los vecinos más cercanos deben pertenecer a la misma clase que la muestra inspeccionada para que esta permanezca en el conjunto de datos.

Utilizando este método, *EditedNearestNeighbours*, se obtuvo un nuevo conjunto de datos previamente sobre muestreado que dio como resultado 4082 muestras y 22 características, como conjunto de datos, con un parámetro n_neighbours de 1000.

La estrategia de validación apropiada que se seleccionó es la de *Stratifed k-fold*, debido al problema de desbalance que existe entre las clases, y para evitar que las muestras de las clases minoritarias no se vieran mal representadas se recurrió a dicha estrategia de validación.

En cuanto a medidas o métricas de desempeño, como medida global o principal se utilizó la medida de *Matthews Correlation Coefficient (*coeficiente de correlación de Matthews) que es una medida que se utiliza en los problemas desbalanceados que tiene en cuenta todas las clases de las especies en cuestión. Como otras medidas se utilizaron las medidas de desempeño accuracy_score (exactitud), roc_auc_score (ROC AUC), área bajo la curva de características operativas del receptor a partir de las puntuaciones de predicción, y f1_score.

La variable de salida especies que es con la que se trabajó se codificó de la siguiente manera, usando el método *LabelEncoder* de *Sklearn*:

0 AdenomeraAndre
1 AdenomeraHylaedactylus
2 Ameeregatrivittata
3 HylaMinuta
4 HypsiboasCinerascens
5 HypsiboasCordobae
6 LeptodactylusFuscus
7 OsteocephalusOophagus
8 Rhinellagranulosa
9 ScinaxRuber

5) Modelos de predicción para clasificación

Para clasificar las muestras de los cantos de los anuros y hacer predicciones se implementaron 5 tipos de modelos de *Machine Learning* que fueron: *Análisis de Discriminante Cuadrático (QDA), Ventana de Parzen o Método Kernel, Gradient Boosting Tree, Redes Neuronales, y Máquina de Vectores de Soporte*. Cada uno de los modelos mencionados tiene parámetros o hiperpárametros que ajustar que se explicaran con más detalle en lo que sigue.

5.1 Análisis de Discriminante Cuadrático

Es un modelo y un clasificador con una frontera de decisión cuadrática que tiene como parámetro ajustar el parámetro de regularización, reg_param. Para este ajuste se consideraron los valores 0.1, 0.2, 0.3, 1.5, 2.5 , 3.7 y 10.5. Se utilizó también una estrategia One vs Rest (ovr) para realizar la implementación debido a que es un problema multi clase. Los resultados de dicha experimentación se muestran en la ***Tabla 5***. En esta tabla se puede observar el valor de los errores de entrenamiento y prueba para cada uno de los valores del parámetro con los que se evaluó, las medidas de rendimiento especificadas en la sección anterior.

	Reg_Param	Error entrenamiento	Error prueba	Accuracy	RAS	f1score
0	0.1	0.800124	0.705015	0.788742	0.969027	0.414208
1	0.2	0.533217	0.406515	0.610841	0.954356	0.221377
2	0.3	0.410358	0.301168	0.561501	0.946729	0.172730
3	1.5	0.000000	0.000000	0.483669	0.918579	0.065199
4	2.5	0.000000	0.000000	0.483669	0.913832	0.065199
5	3.7	0.000000	0.000000	0.483669	0.911361	0.065199
6	10.5	0.000000	0.000000	0.483669	0.907496	0.065199

Tabla 5*.. Resultados que arrojo el modelo QDA*

5.2 Ventana de Parzen o Método Kernel

Es un modelo que es no paramétrico, es decir que no asumen una forma parametrizada del modelo, se utilizó una función kernel de tipo gaussiana y se estimó la función de densidad de probabilidad. El hiperparámetro en este caso a ajustar es el del ancho de la ventana de parzen. Se experimentó con los valores de ancho de la ventana 1.0, 2.0, 3.0 y 10.0. Los resultados obtenidos se muestran en la ***Tabla 6***, en ella se muestran los diferentes valores de la ventana, los errores de prueba, y otras medidas de desempeño que se tuvieron en cuenta para este tipo de modelo.

	h	Error Prueba	Accuracy	f1score
0	1.0	0.871783	0.906880	0.712194
1	2.0	0.777943	0.833912	0.540078
2	3.0	0.754764	0.817234	0.495891
3	10.0	0.672799	0.749131	0.437167

Tabla 6. *Resultados para el modelo Ventana de Parzen.*

5.3 Gradient Boosting Tree

Este modelo es un modelo tipo Boosting. El algoritmo base es el de árboles de decisión. Se ajustaron los parámetros de n_estimators, el número de estimadores y max_depth, la máxima profundidad de los estimadores. Los valores de estos parámetros utilizados en la fase de validación fueron: para n_estimators: 1.0,2.0,3.0,4.0 y para max_depth: 5,10,15,20. Los resultados se muestran en la ***Tabla 7***, que se pueden ver los diferentes resultados para los parámetros, los errores de entrenamiento y de validación, y las demás medidas de desempeño. También se empleó la estrategia ovr para la implementación de este método.

	No Árboles	Profundidad	Error entrenamiento	Error validación	Accuracy	AUC	F1 Score
0	1.0	5.0	0.219971	0.163728	0.502432	0.903398	0.221236
1	1.0	10.0	0.266053	0.214931	0.516331	0.904856	0.291996
2	1.0	15.0	0.266053	0.211546	0.515636	0.901782	0.287112
3	1.0	20.0	0.266053	0.211813	0.515636	0.898243	0.290029
4	2.0	5.0	0.367276	0.262154	0.531619	0.920409	0.349451
5	2.0	10.0	0.364616	0.282419	0.539958	0.925198	0.355939
6	2.0	15.0	0.361222	0.283218	0.539263	0.906997	0.367188
7	2.0	20.0	0.360844	0.284207	0.539958	0.896042	0.366530
8	3.0	5.0	0.914340	0.757150	0.828353	0.931669	0.711305
9	3.0	10.0	0.989883	0.777511	0.842946	0.919848	0.718425
10	3.0	15.0	0.998311	0.799956	0.858235	0.914205	0.749354
11	3.0	20.0	1.000000	0.801958	0.859625	0.897296	0.741464
12	4.0	5.0	0.947078	0.777685	0.842946	0.950765	0.720773
13	4.0	10.0	0.995178	0.782503	0.846421	0.925483	0.717936
14	4.0	15.0	0.998793	0.799889	0.858235	0.913328	0.740283
15	4.0	20.0	1.000000	0.804941	0.861710	0.900478	0.750401

Tabla 7. *Resultados de modelo Gradient Boosting Tree.*

5.4 Redes Neuronales

Para el método de las redes neuronales. Se empleó la red neuronal MLP (Multi Layer Perceptron) que es una red neuronal con varias capas ocultas, y varios perceptrones que son la unidad básica de este tipo de modelos. Los parámetros que se ajustaron fueron el número de capas ocultas (hidden layer sizes) y a su vez la cantidad de neuronas que existen en las capas ocultas. Los parámetros evaluados en la fase de validación fueron hidden_layer_sizes: (10,) y la cantidad de neuronas fue de neurons: 3,4,5,6,7,8,9,10. Los resultados que se obtuvieron se muestran a continuación en la ***Tabla 8***. Se muestran las columnas con los diferentes resultados de los experimentos (parámetros, errores de entrenamiento y de prueba y demás medidas de desempeño).

	Capas ocultas	Neuronas	Error train	Error test	Accuracy	RAS	f1score
0	1.0	3.0	0.810594	0.715848	0.790132	0.950160	0.389161
1	1.0	4.0	0.882717	0.853819	0.894371	0.973387	0.583144
2	1.0	5.0	0.897011	0.826513	0.874218	0.929002	0.537922
3	1.0	6.0	0.930377	0.857949	0.897151	0.958454	0.609362
4	1.0	7.0	0.934352	0.898377	0.926338	0.986670	0.702554
5	1.0	8.0	0.953343	0.903728	0.930507	0.984627	0.802345
6	1.0	9.0	0.959636	0.908868	0.933982	0.990097	0.779412
7	1.0	10.0	0.955483	0.909744	0.934677	0.990404	0.839644

Tabla 8. *Resultados modelo Redes Neuronales (MLP)*

5.5 Máquina de Soporte de Vectores con kernel polinomial

En el modelo de la máquina de soporte de vectores, se pueden tener varios tipos de kernel según la frontera de decisión que se desee, bien sea para fronteras de decisión muy complejas o no tan complejas, El kernel polinomial es un kernel que es un polinomio que tiene cierto grado, se experimentó con grados (degree) de 1,2,3 y con términos de regularización C 0.1, 0.95,3.6 para hallar una buena frontera de decisión que separe los datos. Los resultados se muestran en la **Tabla 9.**, en donde se muestran los parámetros, errores de entrenamiento y prueba, y las medidas de desempeño:

	Par. Regularización	Degree	Error Entrenamiento	Error prueba	Accuracy	RAS	f1score
0	0.10	1.0	0.906165	0.853659	0.893676	0.975145	0.625877
1	0.10	2.0	0.874611	0.714222	0.798471	0.958685	0.497702
2	0.10	3.0	0.920081	0.856454	0.897846	0.985105	0.713562
3	0.95	1.0	0.935359	0.882080	0.914524	0.979704	0.773156
4	0.95	2.0	0.963199	0.887045	0.919388	0.961715	0.786312
5	0.95	3.0	0.985036	0.918310	0.940931	0.987589	0.852614
6	3.60	1.0	0.941953	0.866318	0.902710	0.981568	0.740520
7	3.60	2.0	0.980182	0.912554	0.937457	0.961485	0.852170
8	3.60	3.0	0.996137	0.920231	0.942321	0.990947	0.856129

Tabla 9. *Resultados del modelo SVC, kernel polinomial*

5.6 Máquina de Soporte de Vectores con kernel rbf

Para este experimento se utilizó un kerel de tipo rbf y se ajustaron los parámetros de regularización con valores 0.01, 0.10 y 10.5 y con parámetro gamma de 0.3,0.56,0.68. Los resultados que se obtuvieron están en la ***Tabla 10***, *e*n donde se puede apreciar los resultados para los diferentes parámetros, los errores de entrenamiento y de prueba y las diferentes medidas de desempeño evaluadas.

	Par. Regularización	Gamma	Error Entrenamiento	Error prueba	Accuracy	RAS	f1score
0	0.01	0.03	0.825965	0.816081	0.867269	0.972900	0.522282
1	0.01	0.56	0.469640	0.455834	0.628909	0.971852	0.277074
2	0.01	0.68	0.409281	0.418152	0.608061	0.975577	0.241477
3	0.10	0.03	0.942582	0.892458	0.922168	0.978606	0.742123
4	0.10	0.56	0.986972	0.792836	0.851286	0.992074	0.703609
5	0.10	0.68	0.915582	0.735610	0.809590	0.990405	0.632189
6	10.50	0.03	0.999759	0.956877	0.968728	0.996365	0.935799
7	10.50	0.56	1.000000	0.868175	0.905490	0.994474	0.811857
8	10.50	0.68	1.000000	0.831561	0.879083	0.993993	0.778010

***Tabla 10**. Resultados modelo SVC para kernel rbf*

Los mejores modelos de los que se evaluaron anteriormente fueron:

Modelo	Hiperparamétros
QDA	reg_param:0.1
Parzen Window	h:1
GBT	n_estimators:4-max_depth:15
MLP	hidden_layer_sizes:(10,)
SVC	c:10.5-gamma:0.68

***Tabla 11.** Mejores modelos*

III. SELECCIÓN Y EXTRACCIÓN DE CARACTERÍSTICAS

7) Análisis ingenuo

Inicialmente calculamos la correlación que existe entre las variables con el coeficiente de Pearson, este es un valor que se encuentra entre -1 y 1. Las librerías calculan esto por medio de una matriz. Con la librería seaborn utilizamos un heatmap para poder hacer visible y más analizable la matriz obtenida con los coeficientes de Pearson, dejando el color cero en blanco y los extremos en color azul y café, ver **Figura 13.**

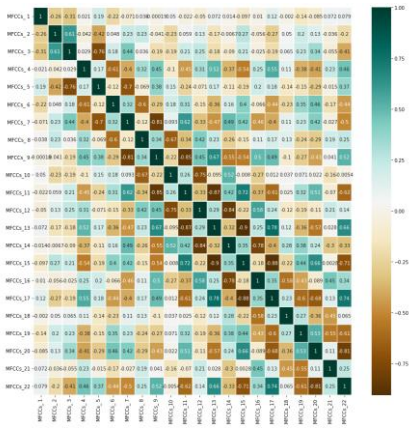


Figura 8. Heatmap de la matriz de correlación entre variables con los coeficientes de Pearson.

Adicional esto calculamos el coeficiente de Fisher dos a dos y de manera similar generamos un algortimo para poder localizar los valores en una matriz, la matriz la normalizamos y luego con heatmap donde el color blanco representa valores cercanos a cero y el morado cercanos a uno generamos la matriz para poder visualizar la capacidad discriminante de las variables, ver figura

***Figura 9**. Heatmap de los coeficientes de Fisher.*

8) Selección de características por el método de búsqueda secuencial ascendente y descendente

En este trabajo se implementó el método de búsqueda secuencial ascendente (Sequential Forward Selection). Para realizar esta búsqueda se utilizó una función criterio tipo wrapper que es un tipo de criterio de selección que, a pesar de ser costoso computacionalmente, es un buen criterio para seleccionar el mejor subconjunto de características, pues presenta buena exactitud, buena capacidad de generalización. La función criterio es la medida f1que se busca maximizar con el fin mencionado previamente, y que también busca la máxima capacidad predictiva del mejor subconjunto de características. La metodología de validación usada es la de *Stratified k-fold.*

Las medidas de desempeño utilizadas fueron accuracy, roc_auc_score y f1_score. Los resultados obtenidos realizando este proceso de selección de características se muestras en la ***Tabla 12*** y los resultados sin la selección de características se muestran en la ***Tabla 13*** para el modelo de Gradient Boosting Tree con sus parámetros.

	Num_features	Reducción	Error test	Accuracy	RAS	F1_score
0	2.0	9.090909	0.534568	0.607877	0.764235	0.439592
1	2.0	9.090909	0.697089	0.747856	0.842903	0.579186
2	2.0	9.090909	0.726706	0.773585	0.846357	0.627780
3	2.0	9.090909	0.665323	0.720412	0.864377	0.590664
4	2.0	9.090909	0.652384	0.711835	0.831281	0.551806
5	2.0	9.090909	0.632308	0.694683	0.851165	0.557957
6	2.0	9.090909	0.649901	0.711835	0.843079	0.521636

Tabla 12. *Resultados con selección para GBT*

	N_feat	Num_splits	Error prueba	Acc	Ras	F1
0	22.0	2.0	0.706904	0.750612	0.922239	0.678987
1	22.0	2.0	0.684726	0.735424	0.891157	0.590357
2	22.0	3.0	0.779057	0.814842	0.930632	0.698233
3	22.0	3.0	0.834995	0.860397	0.968821	0.840397
4	22.0	3.0	0.723094	0.767647	0.880646	0.627616

Tabla 13. *Resultados sin selección para GBT*

Los resultados obtenidos para el modelo MPL de redes neuronales con selección de características se muestran en la ***Tabla 14*** y los resultados para este mismo modelo sin selección de características se observan en la ***Tabla 15***.

	Num_features	Reducción	Error test	Accuracy	RAS	F1_score
0	2.0	9.090909	0.465957	0.565068	0.853269	0.268640
1	2.0	9.090909	0.661693	0.720412	0.870057	0.389848
2	2.0	9.090909	0.656440	0.708405	0.942392	0.358262
3	2.0	9.090909	0.576082	0.651801	0.917238	0.350497
4	2.0	9.090909	0.629503	0.692967	0.832535	0.368106
5	2.0	9.090909	0.549614	0.629503	0.829119	0.310428
6	2.0	9.090909	0.625664	0.686106	0.864524	0.326459

Tabla 14. *Resultados con selección para MLP*

	N_feat	Num_splits	Error prueba	Acc	Ras	F1
1	22.0	4.0	0.367080	0.427032	0.759782	0.197227
2	22.0	4.0	0.278857	0.368266	0.781802	0.150255
3	22.0	4.0	0.262463	0.361765	0.812915	0.142728
4	22.0	4.0	0.324266	0.401961	0.770781	0.169266
5	22.0	5.0	0.369658	0.430845	0.756310	0.203815
6	22.0	5.0	0.297226	0.402693	0.724762	0.121158
7	22.0	5.0	0.307849	0.393382	0.843613	0.171636
8	22.0	5.0	0.254465	0.355392	0.822837	0.133117
9	22.0	5.0	0.340697	0.441176	0.768521	0.164239

Tabla 15. *Resultados sin selección para MLP*

Para el modelo de Máquina de Vectores de Soporte, se muestran los resultados con selección y sin selección de características en ***Tabla 16***, ***Tabla 17***, respectivamente.

	Num_features	Reducción	Error test	Accuracy	RAS	F1_score
0	2.0	9.090909	0.679588	0.729452	0.935272	0.441599
1	2.0	9.090909	0.658789	0.716981	0.873681	0.474512
2	2.0	9.090909	0.803280	0.835334	0.957216	0.579834
3	2.0	9.090909	0.726552	0.773585	0.946223	0.526178
4	2.0	9.090909	0.700141	0.753002	0.893834	0.472959
5	2.0	9.090909	0.630659	0.692967	0.902215	0.457150
6	2.0	9.090909	0.619749	0.684391	0.819148	0.398209

Tabla 16. *Resultados con selección SVC*

	N_feat	Num_splits	Error prueba	Acc	Ras	F1
1	22.0	4.0	0.694181	0.709109	0.987467	0.658938
2	22.0	4.0	0.848448	0.862880	0.998134	0.869800
3	22.0	4.0	0.802608	0.819608	0.991860	0.780057
4	22.0	4.0	0.715981	0.742157	0.989032	0.656289
5	22.0	5.0	0.676952	0.692778	0.986294	0.626373
6	22.0	5.0	0.795763	0.812729	0.993405	0.794136
7	22.0	5.0	0.776758	0.790441	0.994942	0.791007
8	22.0	5.0	0.860472	0.876225	0.996634	0.829049
9	22.0	5.0	0.755894	0.780637	0.990496	0.719007

Tabla 17. *Resultados sin selección SVC*

9) Selección de características PCA

En esta parte del trabajo se realiza una reducción de dimensión con extracción de características con el método Principle Component Analysis (PCA) que es un método que toma muy en cuenta la conservación de la información, ya que esta es variabilidad y se buscan las variables con mucha información. El criterio para seleccionar el número óptimo de componentes fue un criterio tipo wrapper que fue el máximo Coeficiente de Correlación de Matthews, debido a que el problema que se está abordando es un problema desbalanceado y se requiere de un criterio que pueda tener en cuenta todas las clases y no haya sesgo.

En el ***Figura 15*** se muestra la gráfica de varianza explicada para el conjunto de datos que permite visualizar el número de componentes y la varianza acumulada que se gana con el número de componentes asociado. Este permite elegir el número de componentes a partir del codo de la curva. En esta parte, el criterio para elegirlos es el número de componentes con los que se gane el 90% de la varianza de los datos. En este caso son 8 los componentes óptimos.

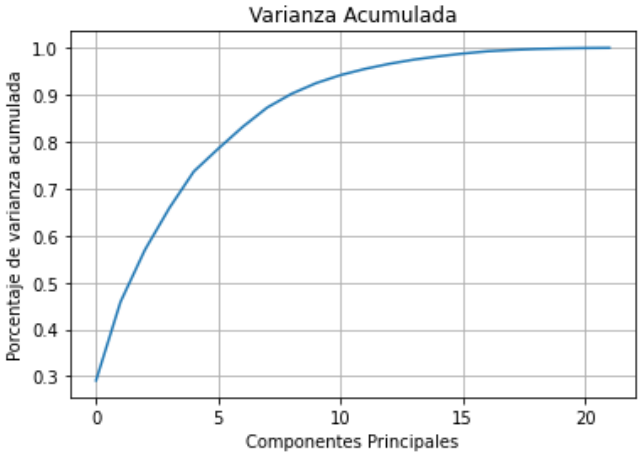


Figura 10. *Varianza explicada*

Los resultados para cada uno de los modelos con extracción de características se muestran en las tablas siguientes **18 a 20**:

	Variables	Reducción	Error prueba	Accuracy	RAS	F1
0	5.0	22.727273	0.699337	0.750306	0.893977	0.640166
1	5.0	22.727273	0.836993	0.864137	0.953155	0.750676
2	5.0	22.727273	0.807460	0.837010	0.946217	0.768904
3	5.0	22.727273	0.828356	0.856618	0.953805	0.792134
4	5.0	22.727273	0.811301	0.841912	0.933639	0.743656
5	6.0	27.272727	0.715108	0.761322	0.892817	0.646031
6	6.0	27.272727	0.846112	0.871481	0.953250	0.756160
7	6.0	27.272727	0.817486	0.845588	0.938890	0.774079
8	6.0	27.272727	0.802551	0.833333	0.944769	0.764184
9	6.0	27.272727	0.813513	0.844363	0.942527	0.747328
10	7.0	31.818182	0.708237	0.753978	0.903689	0.631080
11	7.0	31.818182	0.794187	0.826193	0.946010	0.718182
12	7.0	31.818182	0.812198	0.840686	0.932534	0.786458
13	7.0	31.818182	0.805422	0.835784	0.941023	0.760376
14	7.0	31.818182	0.790628	0.824755	0.934867	0.714496
15	8.0	36.363636	0.725133	0.768666	0.924052	0.669051
16	8.0	36.363636	0.842952	0.869033	0.957854	0.749443
17	8.0	36.363636	0.807666	0.835784	0.935328	0.776105
18	8.0	36.363636	0.833649	0.860294	0.945662	0.782896
19	8.0	36.363636	0.768067	0.806373	0.925927	0.678677

Tabla 18*. Resultados con extracción del modelo GBT*

	Componentes	Reducción	Error prueba	Accuracy	RAS	F1
0	5.0	22.727273	0.815104	0.844553	0.952362	0.676338
1	5.0	22.727273	0.798843	0.832313	0.972951	0.612984
2	5.0	22.727273	0.869985	0.890931	0.981244	0.730922
3	5.0	22.727273	0.857022	0.881127	0.973376	0.701518
4	5.0	22.727273	0.806452	0.838235	0.925611	0.581402
5	6.0	27.272727	0.827351	0.853121	0.957641	0.716029
6	6.0	27.272727	0.893681	0.910649	0.989355	0.798615
7	6.0	27.272727	0.883563	0.901961	0.969090	0.793497
8	6.0	27.272727	0.880713	0.900735	0.988179	0.708764
9	6.0	27.272727	0.857364	0.879902	0.940794	0.702489
10	7.0	31.818182	0.823364	0.850673	0.955530	0.677120
11	7.0	31.818182	0.891208	0.909425	0.982336	0.751054
12	7.0	31.818182	0.875264	0.894608	0.973834	0.736403
13	7.0	31.818182	0.890111	0.908088	0.984969	0.715289
14	7.0	31.818182	0.786090	0.822304	0.916904	0.568345
15	8.0	36.363636	0.736708	0.777234	0.941555	0.580327
16	8.0	36.363636	0.830343	0.856793	0.969339	0.663147

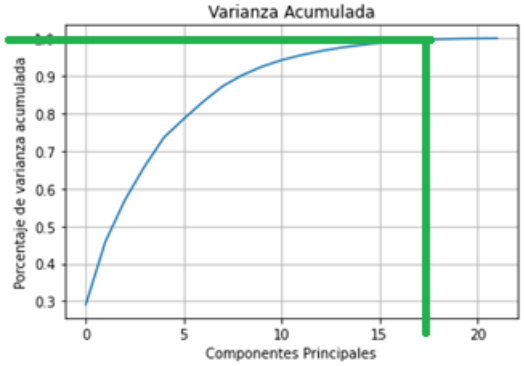
Tabla 19. *Resultados extracción con modelo MLP*

	Componentes	Reducción	Error prueba	Accuracy	RAS	F1
0	5.0	22.727273	0.836947	0.861689	0.981853	0.764406
1	5.0	22.727273	0.886238	0.903305	0.989695	0.865083
2	5.0	22.727273	0.890660	0.908088	0.989734	0.865536
3	5.0	22.727273	0.936907	0.947304	0.994281	0.893598
4	5.0	22.727273	0.887515	0.905637	0.986509	0.832267
5	6.0	27.272727	0.809497	0.837209	0.979882	0.742478
6	6.0	27.272727	0.893401	0.909425	0.992033	0.887157
7	6.0	27.272727	0.892115	0.908088	0.992779	0.868651
8	6.0	27.272727	0.941376	0.950980	0.994711	0.903875
9	6.0	27.272727	0.863251	0.883578	0.986946	0.808841
10	7.0	31.818182	0.821160	0.847001	0.985664	0.765445
11	7.0	31.818182	0.847993	0.867809	0.992935	0.850402
12	7.0	31.818182	0.893101	0.909314	0.994578	0.884373
13	7.0	31.818182	0.936951	0.947304	0.994949	0.904088
14	7.0	31.818182	0.824488	0.846814	0.989306	0.785967
15	8.0	36.363636	0.738029	0.764994	0.986920	0.720763
16	8.0	36.363636	0.826899	0.847001	0.992481	0.840107
17	8.0	36.363636	0.878321	0.893382	0.995180	0.862158
18	8.0	36.363636	0.931435	0.942402	0.996543	0.903083
19	8.0	36.363636	0.818040	0.840686	0.989079	0.775730

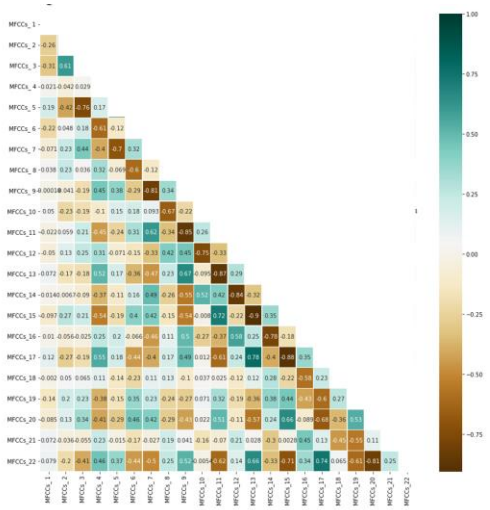
Tabla 20. *Resultados extracción con modelo SVC*

10) Análisis de resultados

- Varianza explicada, podemos seleccionar según este grafico 17 variables.



- Cuando estamos revisando los heatmaps con los coeficientes de correlación de Pearson y Fisher del análisis ingenuo tenemos matrices simétricas y una diagonal llena de unos pues la comparación es consigo misma, estos valores pueden ser anulados mediante el uso de un algoritmo para la visualización solo de los datos relevantes (encima o debajo de la diagonal principal sin la diagonal) o se puede implementar un algoritmo que seleccione en valor absoluto todos los coeficientes que se encuentren por encima del valor que determine el investigador. Los valores de coeficiente de Pearson con valores cercanos a uno, significa que las variables están altamente relacionadas de forma lineal.



- Haciendo un análisis ingenuo especie por especie para revisar la relación que existe entre las variables recomendamos utilizar todos los valores de la matriz para su visualización y análisis, en ella podemos observar a groso modo cuales tienen mayores relaciones entre sí. Las gráficas de color café y verde corresponden al índice de Pearson, y el morado corresponde a al índice de Fisher, la información se presenta en el orden como aparecen las especies en la se de datos de izquierda a derecha y de arriba hacia abajo:

Referencias

[1] J. G. Colonna, M. Cristo, M. Salvatierra, and E. F. Nakamura, “An incremental technique for real-time bioacoustic signal segmentation,” *Expert Systems with Applications*, vol. 42, no. 21, pp. 7367–7374, Jun. 2015, doi: 10.1016/j.eswa.2015.05.030.

[2] J. Colonna, T. Peet, C. A. Ferreira, A. M. Jorge, E. F. Gomes, and J. Gama, “Automatic classification of anuran sounds using convolutional neural networks,” in *ACM International Conference Proceeding Series*, Jul. 2016, vol. 20-22-July-2016, pp. 73–78. doi: 10.1145/2948992.2949016.

[3] J. G. Colonna, J. Gamma, and E. F. Nakamura, *Recognizing family, genus, and species of anuran using hierarchical classification approach*, vol. 9956. Cham: Springer International Publishing, 2016. doi: 10.1007/978-3-319-46307-0.

[4] J. G. Colonna, E. F. Nakamura, M. A. P. Cristo, and M. Gordo, “Base de datos de cantos de anuros de UCI - Machine Learning Repository.” https://archive.ics.uci.edu/ml/datasets/Anuran+Calls+%28MFCCs%29 (accessed Aug. 08, 2022).

[5] Wikipedia, “Definición MFCC.” https://es.wikipedia.org/wiki/MFCC (accessed Aug. 30, 2022).

[6] Wikipedia, “Categorías taxonómicas”, Accessed: Aug. 30, 2022. [Online]. Available: https://es.wikipedia.org/wiki/Categor%C3%ADa_taxon%C3%B3mica

[7] J. G. Colonna, J. Gamma, and E. F. Nakamura, *How to correctly evaluate an automatic bioacustics classificaction method*, vol. 9868. Cham: Springer International Publishing, 2016. doi: 10.1007/978-3-319-44636-3.