INFORME FINAL PREDICCIÓN DE TIPO DE CORTEZA FORESTAL

POR:

Santiago Ramírez Pérez
1017240851

Verónica Palacio Muñoz
1017247211

MATERIA:
Introducción a la inteligencia artificial

PROFESOR:
Raúl Ramos Pollan

UNIVERSIDAD DE ANTIOQUIA

FACULTAD DE INGENIERÍA

MEDELLÍN

2023

CONTENIDO

INTRODUCCIÓN

La naturaleza nos presenta su diversidad de diversas maneras, muchas veces no tenemos ideas de cuantas maneras estamos hablando, y esa pluralidad de formas, texturas, tamaños, etc, genera curiosidad por conocer sus características y de qué forma se podría representar esa diversidad. En este proyecto se debe predecir el tipo de cubierta forestal (el tipo predominante de cubierta arbórea) a partir de variables estrictamente cartográficas (a diferencia de los datos de detección remota). El dataset fue realizado y facilitado por el Sistema de Información de Recursos de la Región 2 del Servicio Forestal de EE. UU. (USFS) y Servicio Geológico de EE. UU. Los datos están en formato bruto (no escalados) y contienen columnas binarias de datos para variables independientes cualitativas, como áreas silvestres y tipo de suelo.

El área de estudio son cuatro áreas silvestres ubicadas en el Bosque Nacional Roosevelt del norte de Colorado. Cada observación es un parche de 30m x 30m. Se le pide que prediga una clasificación entera para el tipo de cubierta forestal. Los siete tipos son:

1. Picea/abeto.
2. Pino Lodgepole.
3. Pino Ponderosa.
4. Álamo/Sauce.
5. Álamo temblón.
6. Abeto de Douglas.
7. Krummholz.

En algunas gráficas aparecerán números en vez de los nombres por el tipo de librerías utilizadas, generalmente empezarán por el 0 y terminarán en 6. En los casos en que ocurra esta situación se mostrarán en el código los nombres para relacionarlos correctamente.

A lo largo del proyecto hemos seguido la misma línea y no hemos presentado cambios en el procesamiento o la forma de ver los resultados, esto permitió que tuviéramos varios modelos predictivos y analizar más a fondo algunos de ellos para comparar y concluir acertadamente.

EXPLORACIÓN DESCRIPTIVA

El dataset que usaremos es de una competencia de kaggle en la cual se proporcionan datos cartográficos para cada celda de 30m x 30m de corteza forestal, estos son:

- Elevation - Elevación en metros.
- Aspect - Aspecto en grados de acimut.
- Slope - Pendiente en grados.
- Horizontal_Distance_To_Hydrology - Horz Dist a las características de agua superficial más cercanas.
- Vertical_Distance_To_Hydrology - Vert Dist a las características de agua superficial más cercanas.
- Horizontal_Distance_To_Roadways - Horz Dist a la carretera más cercana.
- Hillshade_9am (índice 0 a 255) - Hillshade índice a las 9 a. m., solsticio de verano.
- Hillshade_Noon (índice de 0 a 255) - Índice de sombreado al mediodía, solsticio de verano.
- Hillshade_3pm (índice de 0 a 255) - Índice de sombreado a las 3 p. m., solsticio de verano.
- Horizontal_Distance_To_Fire_Points- Dist Horz a los puntos de ignición de incendios forestales más cercanos.
- Wilderness_Area (4 columnas binarias, 0 = ausencia o 1 = presencia) - Designación de área silvestre.
- Soil_Type (40 columnas binarias, 0 = ausencia o 1 = presencia) - Designación de tipo de suelo.
- Cover_Type (7 tipos, números enteros 1 a 7) - Designación del tipo de cubierta forestal.

Para conocer cuáles son las 4 áreas silvestres y los 40 tipos de suelo invitamos a revisar la página oficial de la competencia:

https://www.kaggle.com/competitions/forest-cover-type-prediction/overview/description
Los archivos que utilizamos son:

- **train.csv:** es un archivo con los datos de entrenamiento(con 15120 instancias), descritos anteriormente.

- **test.csv :** son los datos de prueba(con más de 500.000 instancias), que tiene la misma naturaleza que los datos de entrenamiento, en este caso hay que predecir la columna Cover_Type, con el tipo de corteza a la que pertenezca del 1 al 7.

Métrica: la métrica de evaluación principal para el modelo será el porcentaje de precisión multiclases.

accuracy (ACC)

$$\mathrm{ACC} = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{P} + \mathrm{N}} = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TP} + \mathrm{TN} + \mathrm{FP} + \mathrm{FN}}$$

Imagen 1. Métrica de evaluación para el modelo.

Donde:

P: Condición positiva. El número de casos positivos reales en los datos.

N: Condición negativa. El número de casos negativos reales en los datos.

TP: True positive. Un resultado de prueba que indica correctamente la presencia de una condición o característica.

TN: True negative. Un resultado de prueba que indica correctamente la ausencia de una condición o característica.

FP: False positive. Un resultado de prueba que indica erróneamente que una condición o atributo en particular está presente.

FN: False negative. Un resultado de prueba que indica erróneamente que una condición o atributo en particular está ausente.

ITERACIONES DE DESARROLLO

Luego de comprender la tarea que tenemos frente a nosotros, procedemos a revisar, depurar y analizar los datos contenidos en el archivo train.csv. Comenzamos descargando los archivos en GoogleColab desde Kaggle, y con ayuda de la librería pandas leemos y estudiamos los datos del archivo train. Donde primeramente supimos que en cada columna de train hay guardados 15.120 datos, todos de tipo entero, ningún flotante, sin datos faltantes y sin datos nulos, lo que es una buena señal hasta ahora porque nos da cuenta del buen trabajo que se hizo al recolectar los datos.

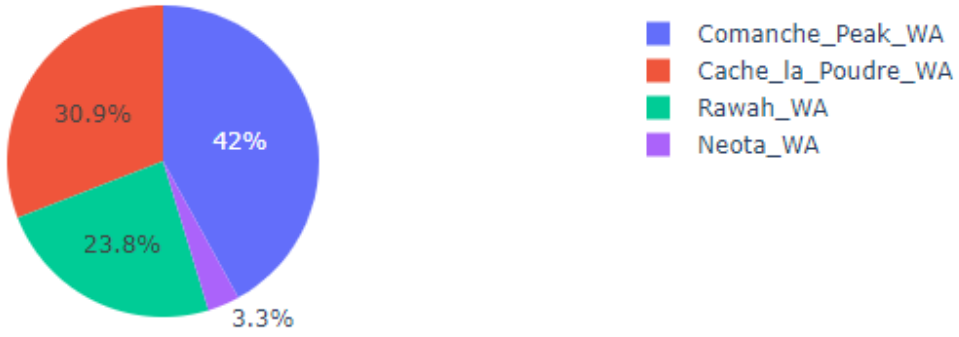
Descripción estadística.

Procedemos a revisar más a fondo el dataframe y le hacemos una descripción estadística, donde de cada columna de train obtendremos los siguientes datos:

- Conteo de datos almacenados.
- Media aritmética.
- Desviación estándar.
- Valor mínimo.
- Percentil 25%.
- Percentil 50%.
- Percentil 75%.
- Valor máximo.

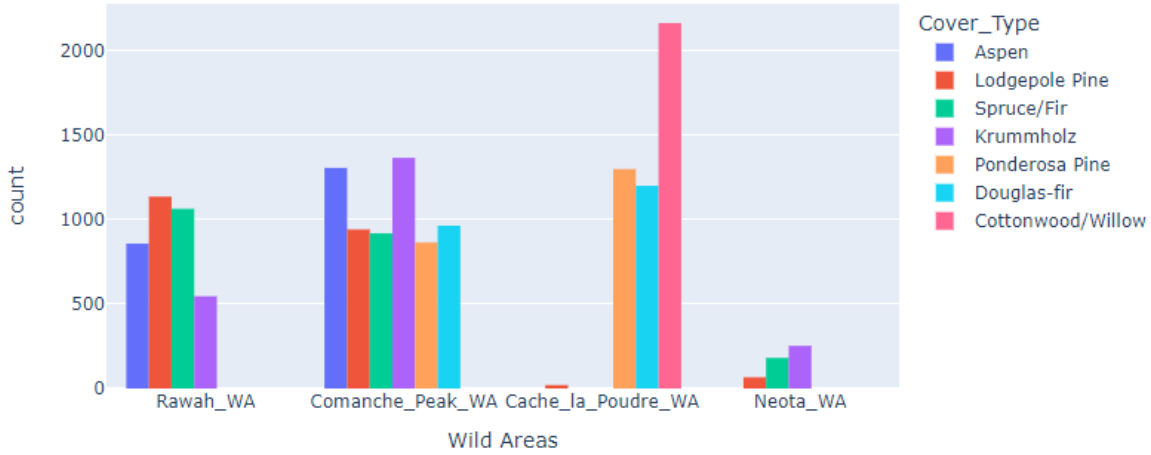
En este punto nos encontramos con algo curioso, en las columnas tipo de suelo #7 y tipo de suelo #15 nos devuelve valores de cero en todo el análisis estadístico. Esto nos deja una dificultad y es si estos valores igual a 0 son porque no se hizo una correcta medición o simplemente pusieron estos valores para no dejar los campos vacíos y tener inconvenientes con datos nulos. Esta es la mayor dificultad que nos hemos encontrado hasta ahora en el proyecto. Además, aunque tengamos incertidumbre en los datos almacenados en train debemos trabajar con lo que tenemos hasta el momento, así que verificamos el sesgo en cada columna para una posterior corrección. Verificamos que en cada tipo de corteza designada en el archivo train hay 2160 instancias para cada una de los 7 tipos de cobertura forestal.

Ahora procedemos a obtener los datos de las distintas áreas silvestres (4) y recordemos que estos datos son del Bosque Nacional Roosevelt del norte de Colorado (Estados Unidos). Obtuvimos un diagrama de tortas donde nos dice que el área silvestre con mayor cantidad de datos y la que menos datos tiene asignados son: Comanche_Peak_WA y Neota_WA respectivamente.



Gráfica 1. Porcentaje de las áreas silvestres con sus cantidades de datos obtenidos.

Realizando un histograma corroboramos esta información dándonos cuenta que en cada área silvestre no se encuentran presentes los 7 tipos de cortezas, y verificando que el área Comanche Peak WA tiene más datos, existiendo en esta área 6 tipos de corteza de 7.



Gráfica 2. Presencia de los diferentes tipos de corteza en cada una de las áreas silvestres.

Algoritmos de clasificación.

Con ayuda de la librería sklearn procedemos a generar modelos de clasificación. Habiendo 9 tipos de modelos predictivos y en la búsqueda del mejor optamos por conocer cuál método de clasificación se ajusta mejor a nuestro proyecto, en este paso obtuvimos la siguiente información luego de generar un bloque de código con un algoritmo de clasificación que probará cada método y nos mostrará su respectivo accuracy sobre el archivo train.

	AccuracyScore	PrecisionScore	RecallScore	f1_Score
RandomForest	0.866733	0.866733	0.866733	0.866733
KNeigbors	0.804894	0.804894	0.804894	0.804894
GradientBoosting	0.802249	0.802249	0.802249	0.802249
DecisionTree	0.793651	0.793651	0.793651	0.793651
SVC	0.625661	0.625661	0.625661	0.625661
Bernoulli	0.610780	0.610780	0.610780	0.610780
Gaussian	0.600860	0.600860	0.600860	0.600860
LogisticRegr	0.474537	0.474537	0.474537	0.474537
AdaBoost	0.346561	0.346561	0.346561	0.346561

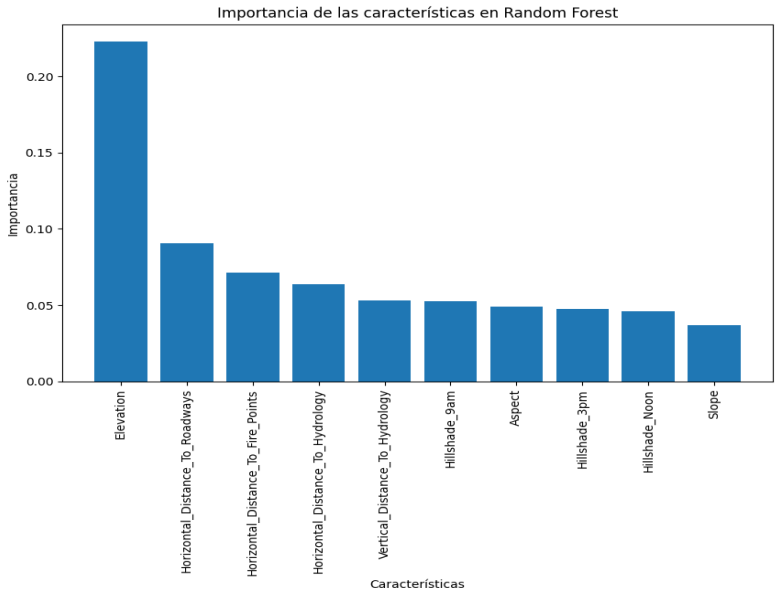
Tabla 1. Algoritmos de clasificación ordenados según su puntaje de predicción.

Esta tabla nos da una idea de los modelos más aproximados y los que no tanto, con base en ella escogemos diferentes modelos para ir más a fondo y descubrir el porqué de estos resultados.

RandomForest.

El primer clasificador analizado es RandomForest, clasificador que, según la puntuación, predice mejor para estos datos.

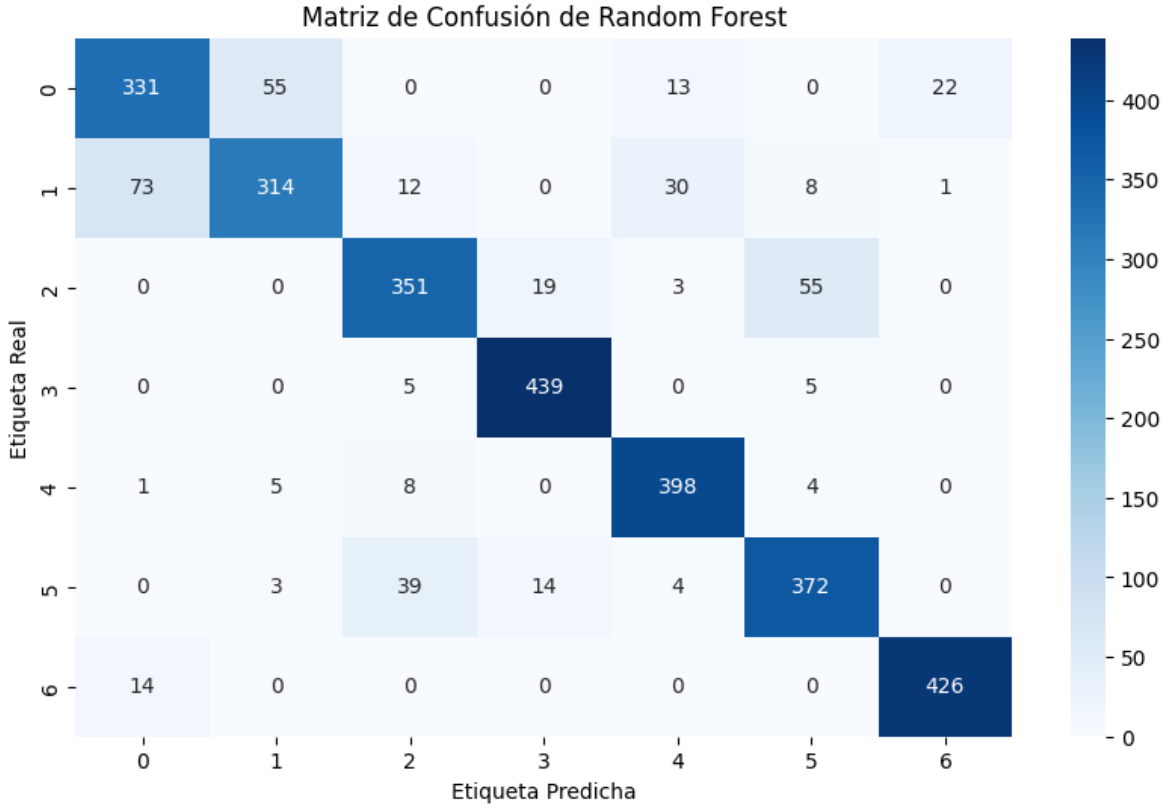
La primera gráfica realizada es acerca de qué tan importantes fueron las características (columnas) de los datos en cuanto a la predicción. Se utilizaron también las 10 primeras características según su importancia.



Gráfica 3. Top 10 características más importantes para RandomForest.

Luego, se grafica una matriz de dispersión que nos da una mejor visual de los datos, el clasificador para los 2 primeros tipos de corteza que son 0. Spruce/Fir y 1. Lodgepole Pine tuvo confusiones y esto se ve reflejado en la matriz.

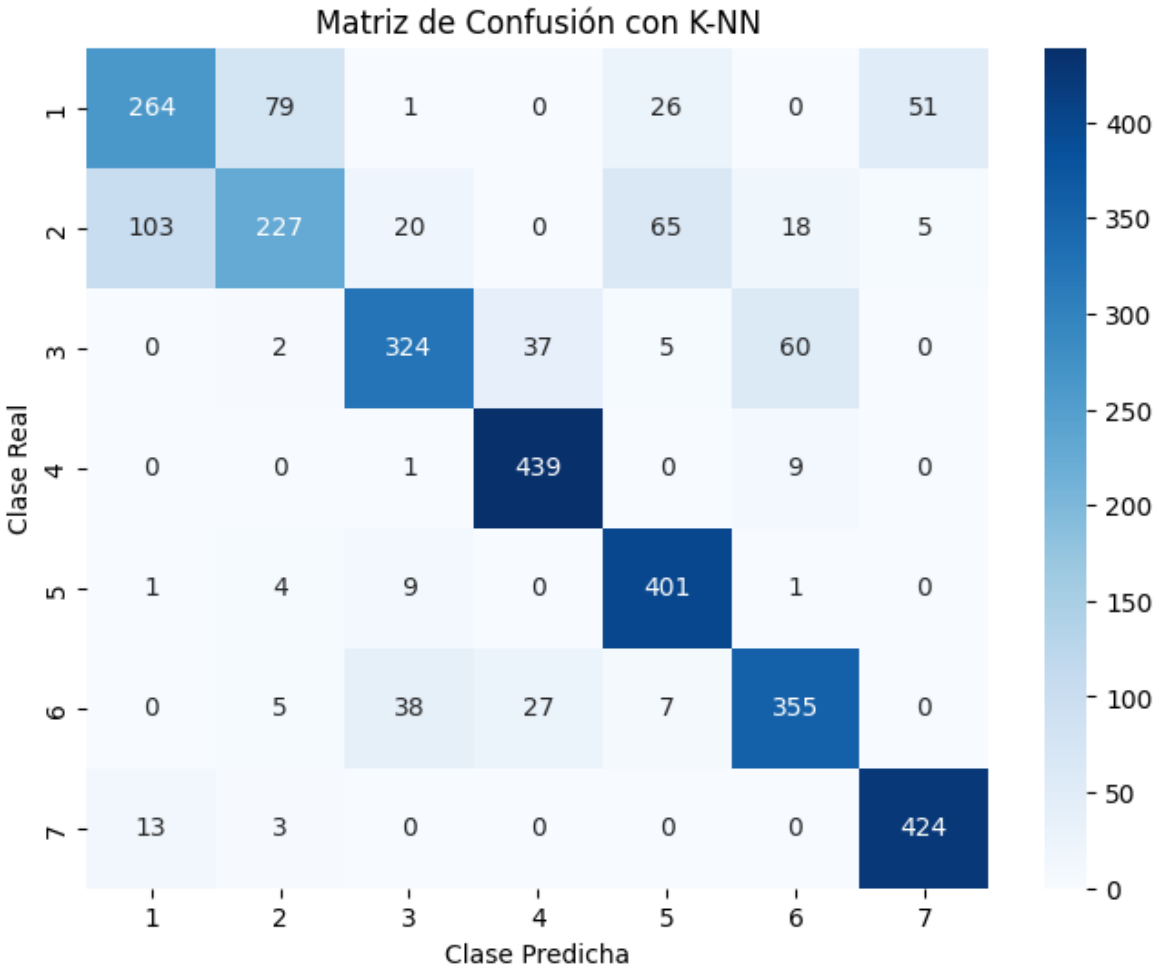
Cuando más oscuro está la cuadricula de la matriz significa que tuvo más datos en esa clasificación, así entonces la diagonal de esta matriz debería contener números diferentes a 0 y el resto de las cuadriculas deberían estar en 0, esto para una precisión “exacta”. Aquí es donde se van a evidenciar las predicciones acertadas vs las que no, entre más 0s haya en la gráfica será mejor para el análisis.



Gráfica 4. Matriz de confusión de RandomForest.

KNN.

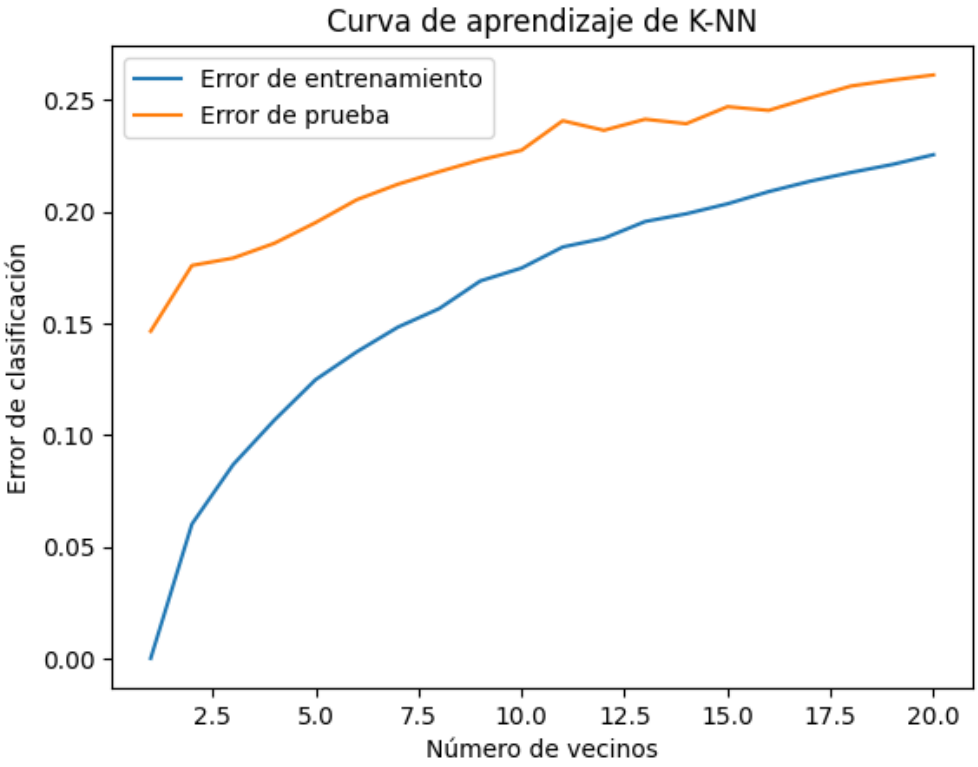
Continua el análisis con el clasificador KNeigbors**,** graficando para él la matriz de confusión y haciendo el análisis similar al mencionado anteriormente.



Gráfica 5. Matriz de confusión de KNN.

Al ser este el segundo clasificador mejor ubicado en nuestra tabla observamos cambios que pudieron afectar el resultado final, algunas predicciones ni difieren respecto a RandomForest y las que difieren lo hacen en cantidades bajas respecto a las mismas predicciones.

Para este clasificador se realiza la curva de aprendizaje donde observamos la variación del rendimiento del modelo respecto a cómo se va entrenando, es por esto que la gráfica tiene una tendencia exponencial.

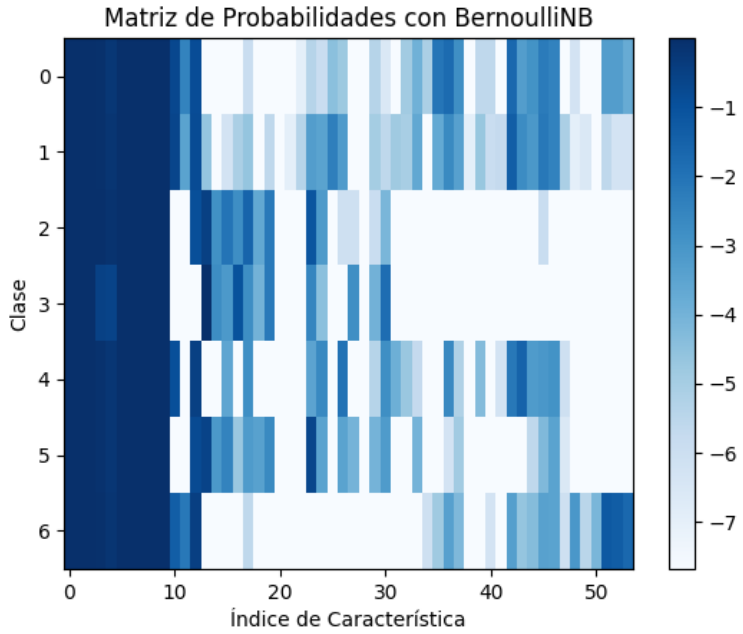


Gráfica 6. Curva de aprendizaje de KNN.

Bernoulli

Seguimos ahora con el clasificador Bernoulli, este clasificador está más debajo de los 2 anteriores y lo esperado es que ocurran más errores en las predicciones.

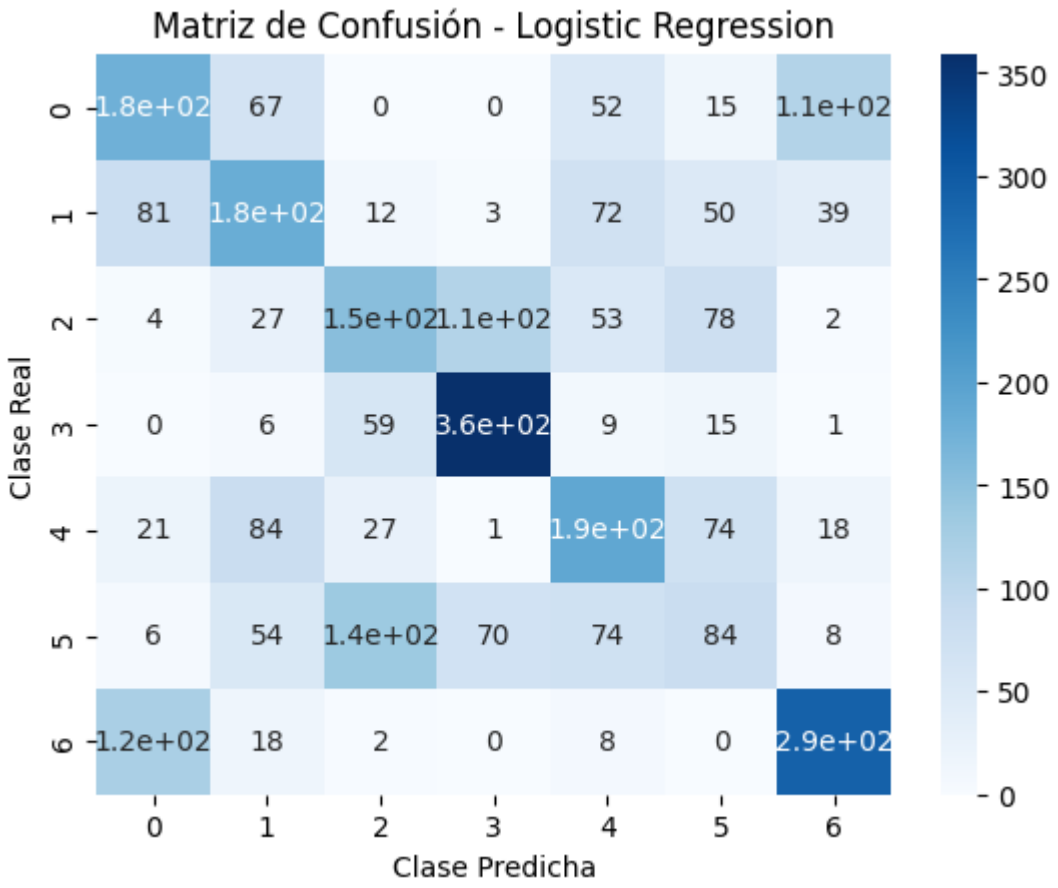
En este caso se realizó una matriz de probabilidades, donde al observar se puede concluir que las primeras 10 caracterísitcas no son relevantes para el clasificador ya que la mayoría de las clases son iguales, a medida de que cambian las características se puede observar la variedad de predicciones, recordemos que hay alrededor de 40 tipos de suelo.



Gráfica 7. Matriz de probabilidades de Bernoulli.

LogisticRegression.

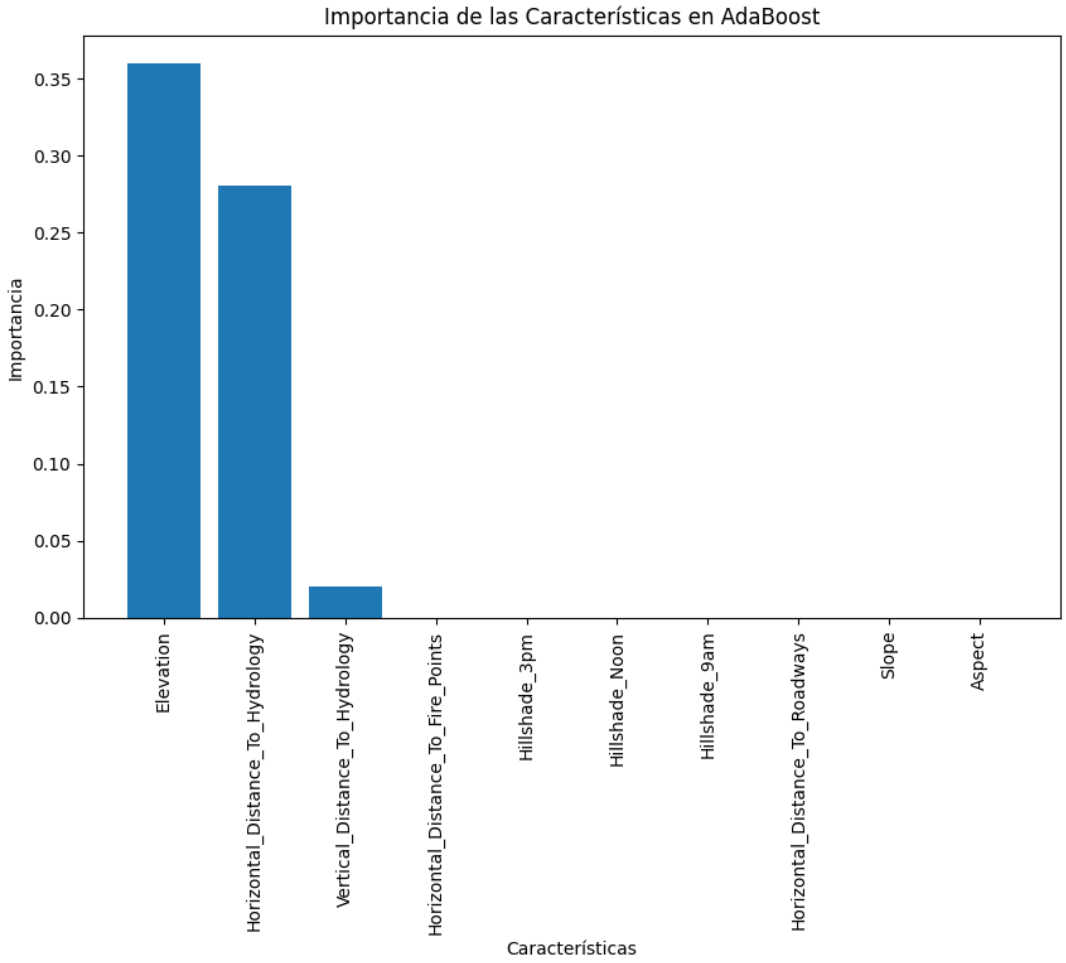
Continuando ahora con el 2 menos efectivo clasificador, para LogisticRegression se realizó una matriz de confusión, en este caso respecto a las otras 2 matrices realizadas se puede observar un gran cambio (solo visualmente sin constatar los números); los colores están más claros y repartidos a lo largo de toda la matriz, lo que indica una menor calidad de predicción y por ello es que ocupa el penúltimo lugar de nuestra tabla.



Gráfica 8. Matriz de confusión de Logistic Regression.

Adaboost.

Por último, analizamos el clasificador con menor puntaje, para Adaboost realizamos una gráfica de importancia para el modelo, en este caso se utilizaron también las 10 primeras características según su importancia.

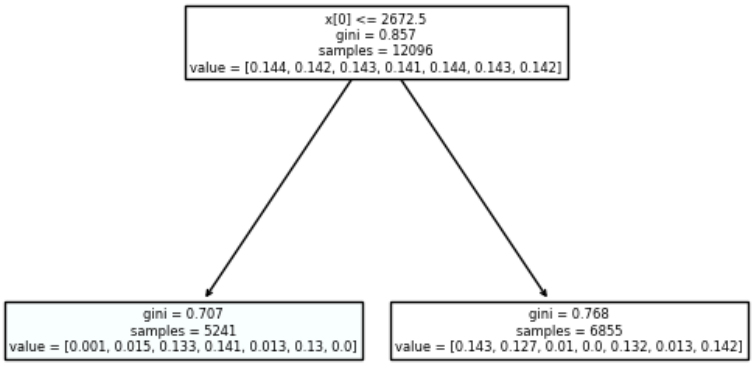


Gráfica 9. características más importantes para Adaboost.

Al ver la gráfica, se observa que 10 clases tienen importancia 0 o mucho menor a 0.05, lo que nos acerca un poco a saber por qué este clasificador ocupa el último lugar de la tabla y la razón para sus predicciones erróneas.

Para mostrar más detallado el proceso de este caso en particular, en el notebook se grafican los 50 árboles de decisión que tiene el algoritmo, para este caso y por practicidad se decide solo mostrar el primero como parte del ejercicio de análisis. Para conocer más a fondo este algoritmo se puede dirigir al programa y ejecutarlo o simplemente observar lo ya graficado.

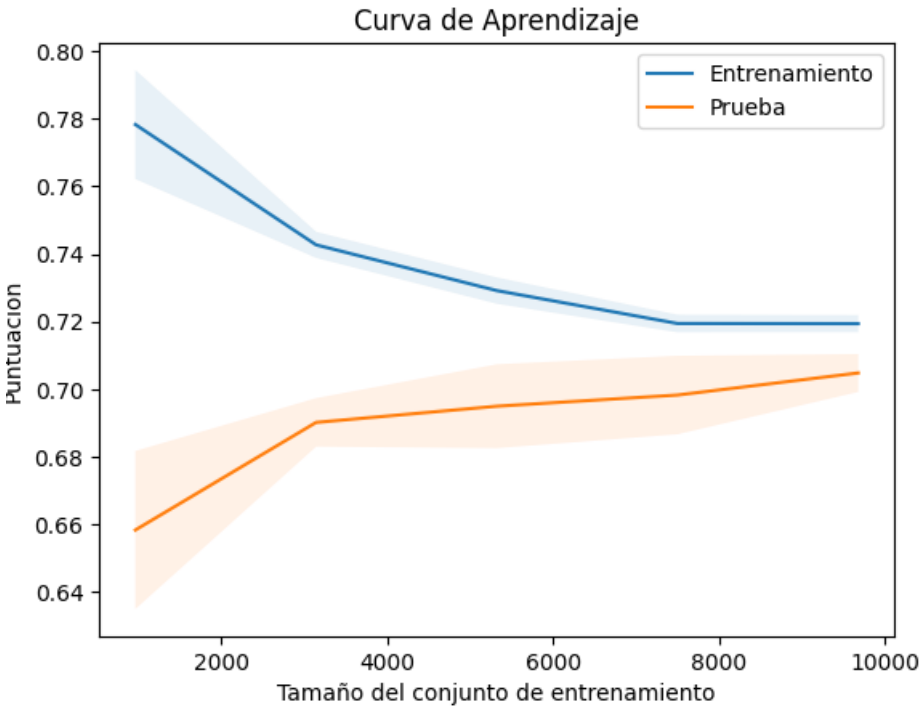
https://github.com/santiagoramirez10/ai4eng_project_ForestCoverType/blob/main/01%20Lectura%20y%20analisis%20de%20datos.ipynb



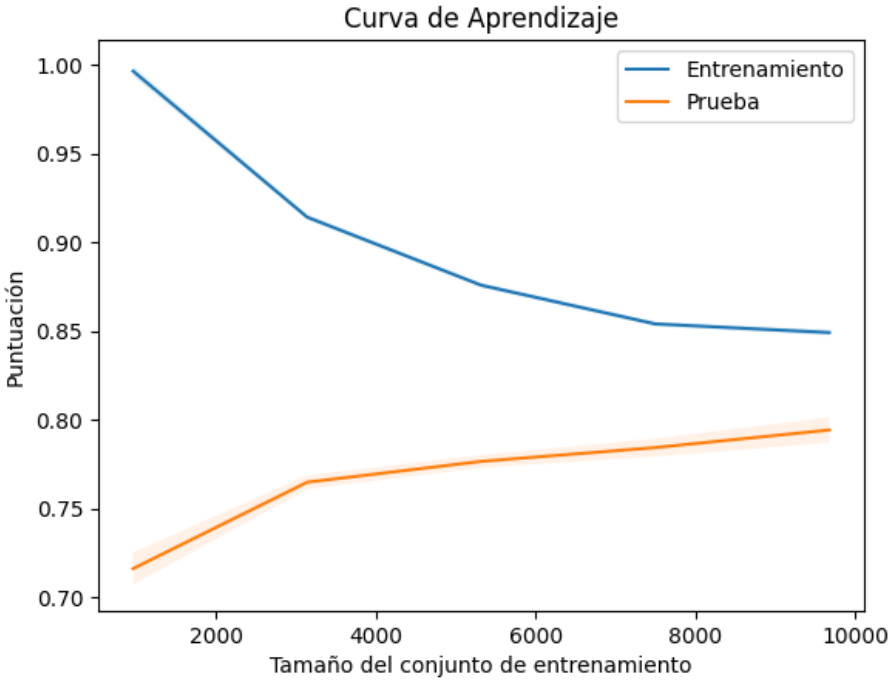
Gráfica 10. Primer árbol de decisión de Adaboost.

Hiperparámetros.

```
Mejores hiperparámetros para Random Forest:
{'max_depth': 5, 'n_estimators': 100}
Mejores hiperparámetros para Gradient Boosting:
{'learning_rate': 0.1, 'n_estimators': 100}
```



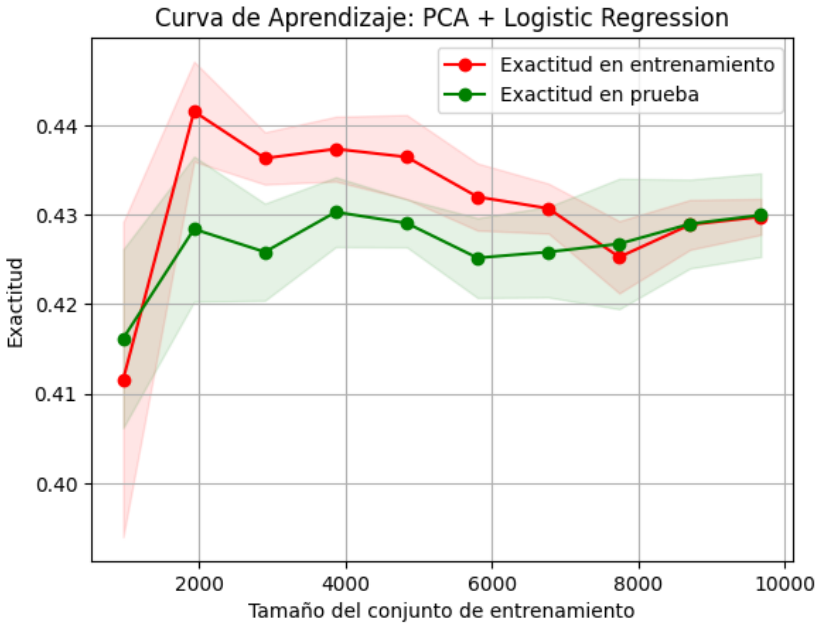
Gráfica 11. Curva de aprendizaje para RandomForest.



Gráfica 12. Curva de aprendizaje para GradientBoosting.

Se utiliza la combinación de PCA con LogisticRegression y se obtiene:

```
Mejor valor de 'n_clusters': 5
Mejor valor de 'C': 1
Exactitud en el conjunto de prueba: 0.05291005291005291
```



Gráfica 13. Curva de aprendizaje para PCA + LogisticRegression.

Para PCA con SVC:

```
Mejor valor de 'n_components': 3
Mejor valor de 'C': 1
Mejor valor de 'gamma': 0.1
Exactitud en el conjunto de prueba: 1.0


Exactitud del modelo SVM: 1.00
```

En general para cualquier proyecto es difícil dar una opinión acertada con tan poco tiempo de exploración. Como grupo sugerimos tomar muchos más datos que permitan conocer más a fondo los tipos de árboles que poseen estas características tan particulares, además, la obtención de más datos reduce el sesgo y nos da una visión más acertada de lo que debería ser.

Al seleccionar los modelos se pueden presentar similitudes que pueden indicar nada, como sugerencia está el utilizar muchos más modelos que permitan diferenciar mejor los datos. El tiempo es corto y se logró en él una cantidad de modelos que nos aproximan un poco a la realidad, consideramos que lo tratado en esta asignatura debería tener mucho más tiempo para experimentar y aportar a los proyectos.

El sesgo de los datos hace que algunos modelos presenten problemas en “aprender”, es por ello que es necesario una mejor máquina que permita el correcto funcionamiento de los programas más pesados; al ser un dataset tan pesado, en ocasiones el equipo responde lento, es por esto que recomendamos también realizar divisiones de los datos.

RETOS

A lo largo de la realización del proyecto encontramos retos que nos ayudan a mejorar nuestra capacidad de análisis y decisión. Uno de ellos y el más importante es por dónde empezar, qué datos elegir y porqué. Escoger los modelos de clasificación también es un reto importante ya que según estén los datos recogidos un clasificador puede funcionar o no. Afortunadamente, la primera parte del trabajo que era escoger el dataset funcionó de la mejor manera y no perdimos más tiempo en buscar otra competencia. Es importante elegir bien al comienzo para así a lo largo de un semestre lograr un trabajo como el realizado hasta hoy por el equipo.

Otro de los retos fue que al analizar el dataset por medio del análisis estadístico, nos encontramos con algo curioso, en las columnas tipo de suelo #7 y tipo de suelo #15 nos devuelve valores de cero en todas sus filas.

Esto nos deja una dificultad y es si estos valores igual a 0 son porque no se hizo una correcta medición o simplemente pusieron estos valores para no dejar los campos vacíos y tener inconvenientes con datos nulos. Además, aunque tengamos incertidumbre en los datos almacenados en train debemos trabajar con lo que tenemos hasta el momento. Solucionamos esto verificando que en cada tipo de corteza designada en el archivo train hay 2160 instancias para cada uno de los 7 tipos de cobertura forestal. Realmente no hay una razón verídica del porqué de estos datos, sin embargo, el equipo decidió trabajar sobre ellos y concluir que si están ahí los datos es porque son importantes para los encargados de recolectar los datos.

CONCLUSIONES

- Obteniendo una mayor cantidad de datos y conociendo la razón de por qué las 2 columnas mencionadas anteriormente contienen solo 0, podría reducirse el sesgo y obtener una mejor predicción.

- Los 9 modelos empleados nos muestran que a pesar de que haya diversidad en los datos, ninguno es más efectivo que otro en general, dependiendo del problema uno u otro nos ayudará a comprender mejor el problema y obtener mejores predicciones.

- El sesgo de los datos depende mucho de su naturaleza, al tener valores binarios y no datos más descriptivos puede crearse confusiones. Al tener estos datos, se optó por mostrar los valores correspondientes a los valores numéricos y así poder tener una mejor visión del problema.

- Al graficar algunos de los datos proporcionados por los clasificadores, podemos obtener una mejor perspectiva visual y obtener un mejor criterio que viendo los datos solo plasmados en matrices o listas.

BIBLIOGRAFÍA

- FOREST COVER TYPE PREDICTION – Use cartographic variables to classify forest categories | Kaggle. (2022). Retrieved 4 July 2022, from https://www.kaggle.com/competitions/forest-cover-typeprediction/overview/description