RAIN PREDICTION MODEL

MACHINE LEARNING MODELS

MARZO.20, 2021

ELABORADO POR
HANS WALTER

TABLA DE CONTENIDOS

INTRODUCTION

En este segundo proyecto, estuvimos trabajando con datos sobre el clima en la ciudad de Canberra que se encuentra en Australia. El dataset contaba con numerosas variables que incluían desde la dirección del viento, como también la humedad y presión registrada en cada estación. Como finalidad, queríamos desarrollar un modelo que fuera capaz de predecir si iba a llover al día siguiente o no. En el análisis, utilizamos tres modelos de clasificación. Estos incluyen: logistic regression, linear svc y decision tree classifier. Se logró crear un modelo pero con un rendimiento abajo de lo esperado. Pueden afectar numerosos factores como la falta de variables correlacionadas, la existencia de variables insignificantes en el análisis, como también la posibilidad de que no es posible poder predecir el clima.

DATA

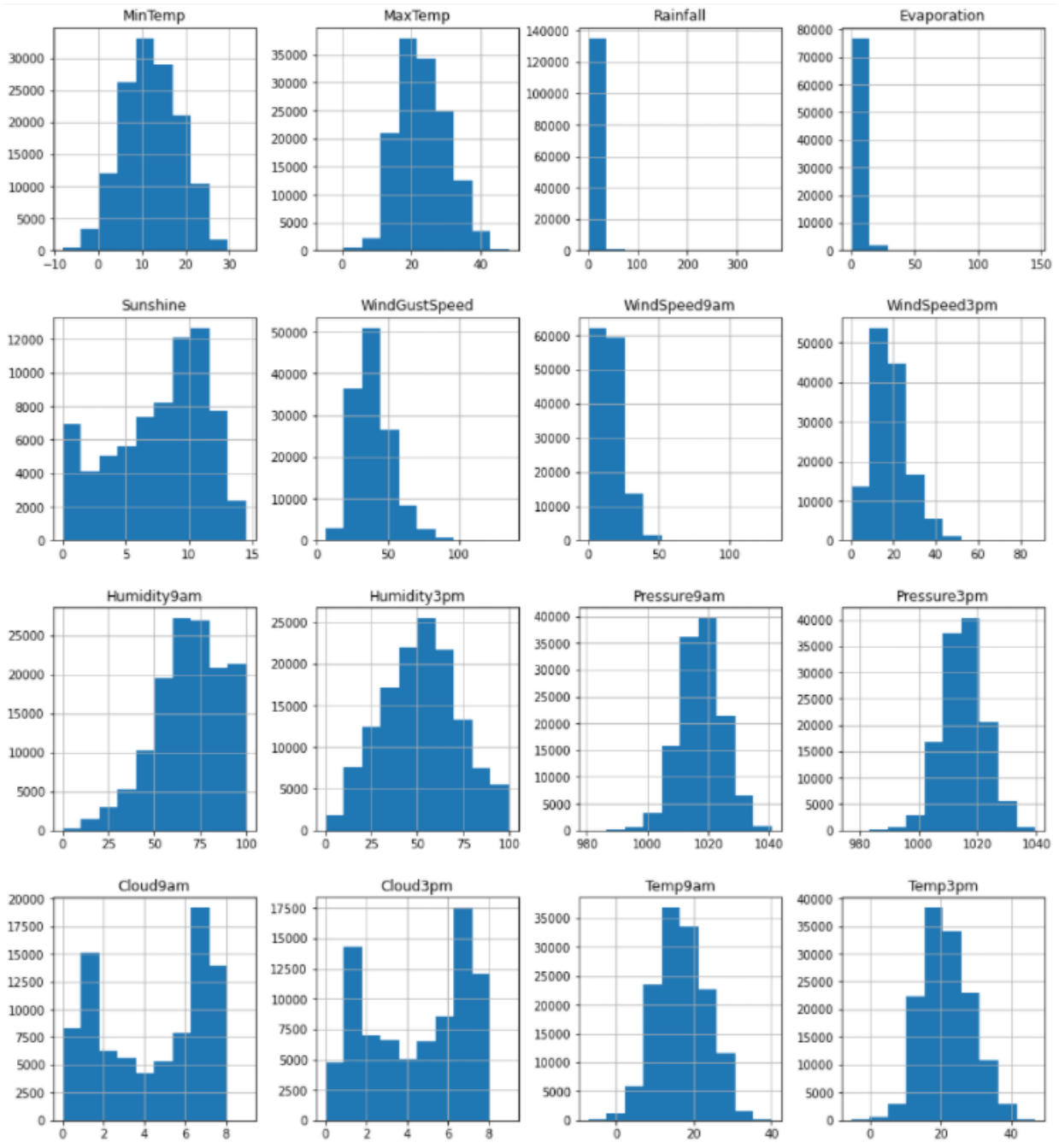
El dataset es compuesto por 22 columnas de features y 1 de label. 16 columnas son númericas mientras que solo 5 son categóricas. La distribución de los datos en su mayoría se encontraba sesgada, con unas pocas columnas con comportamiento bimodal y el resto una distribución normal. Notamos previo a cualquier elaboración del modelo que la variable objetivo era compuesta por un 77% de resultados 0 (No llueve) y un 23% de resultados 1 (Sí llueve). Es importante recalcar esto ya que cuando se tiene una variable objetivo sesgada, es probable que el modelo se aprenda de memoria el training set. Por lo mismo se utiliza cross-validation para asegurar su generalización pero de igual manera afecta el rendimiento del modelo.

DATA

Dataset Columns

Date: The date of observation
Location: The common name of the location of the weather station
Location: The minimum temperature in degrees celsius
MinTemp: The maximum temperature in degrees celsius
MaxTemp: The amount of rainfall recorded for the day in mm
Rainfall: The amount of rainfall recorded for the day in mm
Evaporation: The so-called Class A pan evaporation (mm) in the 24 hours to 9am
Sunshine: The number of hours of bright sunshine in the day.
WindGustDir: The direction of the strongest wind gust in the 24 hours to midnight
WindGustSpeed: The speed (km/h) of the strongest wind gust in the 24 hours to midnight
WindDir9am: Direction of the wind at 9am
WindDir3pm: Direction of the wind at 3pm
WindSpeed9am: Wind speed (km/hr) averaged over 10 minutes prior to 9am
WindSpeed3pm: Wind speed (km/hr) averaged over 10 minutes prior to 3pm
Humidity9am: Humidity (percent) at 9am
Humidity3pm: Humidity (percent) at 3pm
Pressure9am: Atmospheric pressure (hpa) reduced to mean sea level at 9am
Pressure3pm: Atmospheric pressure (hpa) reduced to mean sea level at 3pm
Cloud9am: Fraction of sky obscured by cloud at 9am. This is measured in "oktas", which are a unit of eigths.
Cloud3pm: Fraction of sky obscured by cloud (in "oktas": eighths) at 3pm. See Cload9am for a **description** of the values
Temp9am: Temperature (degrees C) at 9am
Temp3pm: Temperature (degrees C) at 3pm
RainToday: Boolean: 1 if precipitation (mm) in the 24 hours to 9am exceeds 1mm, otherwise 0
RainTomorrow: The amount of next day rain in mm. Used to create response variable RainTomorrow. A kind of measure of the "risk". (Target variable)

DATA


MinTemp
MaxTemp
Rainfall
Evaporation
Sunshine
WindGustSpeed
WindSpeed9am
WindSpeed3pm
Humidity9am
Humidity3pm
Pressure9am
Pressure3pm
Cloud9am
Cloud3pm
Temp9am
Temp3pm

METHODS

NA'S EN VARIABLE OBJETIVO

Ya que nuestro dataset cuenta con 141,194 registros, aplicamos un listwise deletion de la columna de la variable objetivo Y: RainTomorrow. Esto se debe a que la columna tiene 3,160 datos faltantes, a comparación de la cantidad de datos que tiene el dataframe, hacer un listwise deletion va tener efectos insignificantes. Es importante eliminar estos NA's, ya que la columna de la variable objetivo Y: RainTomorrow tiene que estar completa para poder crear un modelo.

LABEL ENCODING Y

El dataset originalmente tenía los resultados de la varíable objetivo como "Yes" o "No". Codificamos los "Yes"=1, "No"=0.

PREPROCESSING PIPELINES

Para las columnas categóricas, utilizamos el método de "most_frequent" para reemplazar los valores nulos. Después utilizamos "OneHotEncoding" que primero convertía las variables categóricas en representaciones números, y por último las volvía dummy variables. Esto es necesario para poder hacer regresiones como también ayudar a que la máquina interprete estas variables. Para las columnas números, decidimos aplicar a todas el método de imputación por "median". Este método era el que mejor aplicaba para todos los tipos de distribuciones, ya que las normales no sufrían efecto (media y promedio son casi iguales en estas) y las distribuciones sesgadas era apropiado la media por la presencia de outliers.

METHODS

LOGISTIC REGRESSION

El modelo estima la probabilidad que la variable objetivo pertenezca a cierta clase. Como es un clasificador binario, estos rangos son de 0 a 1 (0% a 100%). También se puede denotar que se comporta como una función sigmoide. Normalmente el threshold con el que se trabaja si se quiere convertir las probabilidades en clases, es que si p >= 50% se clasifica como 1, de lo contrario es 0. A continuación hablaremos sobre los parámetros que utilizamos en el GridSearchCV, que dejamos que la máquina escogiera el que considerara más apropiado dependiendo del F1 score que tuviera.

- L1 Lasso Regression: Agarra el valor de la suma de los errores al cuadrado, y lo suma por el valor de lambda multiplcado por el valor al cuadrado de la pendiente. Lo incorporamos al param_grid, ya que normalmente descarta totalmente uno o más set de features que sean irrelevantes para el análisis.
- L2 Ridge Regression: Agarra el valor de la suma de los errores al cuadrado, y lo suma por el valor de lambda multiplicado por el valor absoluto de la pendiente. Este método a diferencia de L1, simplemente reduce el peso de variables insignificantes al análisis final.
- Valor C: Mientras más grande sea C, va a ser menos fuerte la regularización. Mientras más pequeño, va a resultar más rígido. Sí el valor de C es 1, significa que casi no tiene penalización presente. Sí el valor C se acerca a 0, tiene penalización más rígida. En nuestro análisis teníamos los rangos de [0.001 0.01, 0.1, 1]
- Algoritmos: newton-cg(L2), lbfgs(L2), sag(L2), saga(L1|L2), liblinear(L1|L2). Cada algoritmo tiene un modelo matématico complejo. Vemos también que algunos pueden utilizarse solo en un tipo de regresión y en pocos casos en ambos. A pesar de no ver a fondo los algoritmos, decidimos incluirlos para darle más flexibilidad al modelo al momento de escoger parámetros óptimos.

LINEAR SVC

El propósito de los márgenes es establecer una barrera/límite entre clases. Se destaca el modelo ya que utiliza clasificación con Soft Margin o Hard Margin. Puede elaborar clasificaciones lineales y no lineales. No proporciona probabilidades como el modelo de logistic regression.

- Soft Margin Classification: Más flexible y generalizable, pero puede ocasionar underfitting/bias.
- Hard Margin Classification: Sensible a la presencia de outliers, puede ocasionar overfitting/variation.

Parámetros:

- Valor C: Un valor bajo no penaliza de manera significativa la mala clasificación de la variable objetivo, a diferencia de un valor alto que ocasiona un margen más pequeño entre las dos clases que se están prediciendo. Usamos los mismos rangos mencionados en el modelo anterior.
- **Loss**: **Hinge -** Para generar el valor de loss, se resta el valor predecido con el valor real de y. **Squared_hinge -** Lo anteriormente definido pero simplemente se eleva al cuadrado. Esto significa que los errores que sean de mayor magnitud son penalizados más fuerte.

DECISION TREE CLASSIFIER

Tiene tres componentes principales las cuales son el root node (con el que comienza), los demás nodos que expanden el árbol, y finalmente los leaf nodes. El modelo funciona como un diagrama de flujo hasta llegar a un leaf node que determina que clase predice el modelo. El árbol termina de expanderse cuando el coeficiente Gini no tiene manera de reducirse. Los árboles de decisión son considerados supervised models, capaz de manejar datos númericos como también categóricos y no asumen ningún tipo de distribución de los datos. El modelo tiene como debilidad que fácilmente hace overfit. El modelo tiene los siguientes parámetros:

- Splitter: Es el método que utilizará para partir los nodos, donde best escoge el mejor por medio del valor de gini y en random de manera aleatoria. Anterior al correr al modelo creo que es mejor utilizar el método best pero igual manera podemos probar a ver que sucede.
- Max_depth: Que tan profundo puede llegar a ser el árbol en cuanto a niveles. En este caso lo dejamos establecido en entre 5, 10 y 15. Evitamos usar números más altos porque estaríamos arriesgándonos de cometer overfitting.
- Min_samples_leaf: Establecer la mínima cantidad de observaciones que tienen que haber para que se considere un leaf el grupo.

RESULTS

SCORE ON TESTING SET

Recordemos que el F1 Score es la ponderación del score que saca el modelo en los resultados de Recall y de Precision. Le ponemos énfasis a este resultado ya que estamos interesado en que el modelo sea ambos bueno en Recall (ej. clasificando armas y preferiendo tener errores positivos negativos.) y en Precision (ej. clasificando contenido apto para niños, prefiriendo tener errores falsos negativos)

LOGISTIC REGRESSION GENERAL F1 SCORE: 59%

- CON 0: F1 90%, RECALL: 94%, PRECISION: 87%
- CON 1: F1 60%, RECALL: 51%, PRECISION: 73%

Como establecimos al inicio, mencionamos el riesgo que el modelo iba a estar sesgado por la distribución de la variable objetivo. En el diagrama, el modelo de logistic tiene punteo alto en F1 Score 90% cuando no llueve a comparación de cuando sí llueve de tan solo 60%. Separandolo entre 0 y 1, vemos que tiene mayor fuerza en recall en 0 (es decir que tiene menos errores falsos negativos) y mejor precision en 1 (menor errores con falsos positivos). Es posible que por su amplio número de parámetros trabajados le permitió ser el modelo con mejor score de los tres.

LINEAR SVC GENERAL F1 SCORE: 58%

- CON 0: F1 90%, RECALL: 95%, PRECISION: 86%
- CON 1: F1 59%, RECALL: 49%, PRECISION: 74%

El comportamiento bastante similar al del modelo anterior. Sin embargo ahorita podemos ver que el Linear SVC tiene mejor rendimiento en recall con la categoría de 0(No Llueve) y un mejor precision en la categoría de 1(Llueve) que en el modelo de Logistic Regression.

DECISION TREE GENERAL F1 SCORE: 58%

- CON 0: F1 90%, RECALL: 94%, PRECISION: 86%
- CON 1: F1 58%, RECALL: 49%, PRECISION: 70%

Este modelo podemos establecer que es el menos óptimo de los tres, ya que los scores de recall y precision de cada categoría son menores a los anteriormente presentados.

CONCLUSION

Iniciamos el proyecto con el propósito de generar un modelo capaz de predecir si iba a haber lluvia al día siguiente. A pesar de lograrlo, observamos que su rendimiento está un poco arriba de lo que sería simplemente adivinar si llueve o no. El modelo más apropiado para utilizar para predecir la lluvia al día siguiente sería el de **Logistic Regression con un F1 Score de 59%.** Para mejorar el modelo, se requiere de más tiempo como también poder computacional, para poder procesar más parámetros y combinaciones. Otro aspecto el que puede mejorar el resultado del modelo es ya sea, tomar en consideración menos columnas para el análisis (puede ocasionar ruido y distorsión a los resultados), como también agregar otras que puedan tener correlación con el resultado si llueve al siguiente día. Podemos concluir entonces, que es posible predecir si llueve al siguiente día pero no con absoluta certeza.

REFERENCES

- 1.4. support Vector Machines¶. (n.d.). Retrieved March 21, 2021, from https://scikit-learn.org/stable/modules/svm.html
- Amballa, A. (2020, August 18). Feature engineering part-1 mean/ median imputation. Retrieved March 21, 2021, from https://medium.com/analytics-vidhya/feature-engineering-part-1-mean-median-imputation-761043b95379
- Badr, W. (2019, January 12). 6 different ways to compensate for missing data (data Imputation with examples). Retrieved March 21, 2021, from https://towardsdatascience.com/6-different-ways-to-compensate-for-missing-values-data-imputation-with-examples-6022d9ca0779
- Chowdhury, K. (2019, December 20). Understanding loss functions : Hinge loss. Retrieved March 21, 2021, from https://medium.com/analytics-vidhya/understanding-loss-functions-hinge-loss-a0ff112b40a1
- Chris, & User, M. (2021, February 08). How to use hinge & squared hinge loss with TensorFlow 2 and Keras? Retrieved March 21, 2021, from https://www.machinecurve.com/index.php/2019/10/15/how-to-use-hinge-squared-hinge-loss-with-keras/#what-is-hinge-loss
- A complete guide to histograms. (n.d.). Retrieved March 21, 2021, from https://chartio.com/learn/charts/histogram-complete-guide/
- Kassambara, & Chandrakumaran. (2018, March 11). Penalized regression Essentials: Ridge, LASSO & elastic net. Retrieved March 21, 2021, from http://www.sthda.com/english/articles/37-model-selection-essentials-in-r/153-penalized-regression-essentials-ridge-lasso-elastic-net/#lasso-regression
- Logistic regression optimization & parameters. (2021, January 26). Retrieved March 21, 2021, from https://holypython.com/log-reg/logistic-regression-optimization-parameters/
- Nagpal, A. (2017, October 14). L1 and l2 regularization methods. Retrieved March 21, 2021, from https://towardsdatascience.com/l1-and-l2-regularization-methods-ce25e7fc831c
- Parulpandey. (2020, July 11). A guide to handling missing values in python. Retrieved March 21, 2021, from https://www.kaggle.com/parulpandey/a-guide-to-handling-missing-values-in-python
- Qiao, F. (2019, January 08). Logistic regression model tuning with scikit-learn - part 1. Retrieved March 21, 2021, from https://towardsdatascience.com/logistic-regression-model-tuning-with-scikit-learn-part-1-425142e01af5
- Residentmario. (2018, April 28). Simple techniques for missing data imputation. Retrieved March 21, 2021, from https://www.kaggle.com/residentmario/simple-techniques-for-missing-data-imputation
- Science, O. (2020, February 25). Data imputation: Beyond mean, median, and mode. Retrieved March 21, 2021, from https://medium.com/@ODSC/data-imputation-beyond-mean-median-and-mode-6c798f3212e3
- Shashankasubrahmanya. (2018, June 07). Missing data imputation using regression. Retrieved March 21, 2021, from https://www.kaggle.com/shashankasubrahmanya/missing-data-imputation-using-regression
- Sklearn.linear_model.LogisticRegression¶. (n.d.). Retrieved March 21, 2021, from https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html
- Sklearn.model_selection.gridsearchcv¶. (n.d.). Retrieved March 21, 2021, from https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html
- Sklearn.svm.linearsvc¶. (n.d.). Retrieved March 21, 2021, from https://scikit-learn.org/stable/modules/generated/sklearn.svm.LinearSVC.html
- Sklearn.svm.linearsvc¶. (n.d.). Retrieved March 21, 2021, from https://scikit-learn.org/stable/modules/generated/sklearn.svm.LinearSVC.html#sklearn.svm.LinearSVC
- Sklearn.svm.svc¶. (n.d.). Retrieved March 21, 2021, from https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html
- Sklearn.tree.decisiontreeclassifier¶. (n.d.). Retrieved March 21, 2021, from https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html
- Stef Van Buuren. (n.d.). Retrieved March 21, 2021, from https://stefvanbuuren.name/fimd/sec-nonnormal.html
- Stephanie. (2020, December 14). Bimodal distribution: What is it? Retrieved March 21, 2021, from https://www.statisticshowto.com/what-is-a-bimodal-distribution/#:~:text=Bimodal%20Distribution%3A%20Two%20Peaks.&text=The%20bimodal%20distribution%20has%20two%20peaks.&text=However%2C%20if%20you%20think%20about,stop%20increasing%20and%20start%20decreasing.
- Stephanie. (2021, January 26). Multimodal distribution definition and examples. Retrieved March 21, 2021, from https://www.statisticshowto.com/multimodal-distribution/
- Tyagi, N. (2020, September 30). Understanding the gini index and information gain in decision trees. Retrieved March 21, 2021, from https://medium.com/analytics-steps/understanding-the-gini-index-and-information-gain-in-decision-trees-ab4720518ba8