TP Integrador AdD & IIA

Estudiante: Gabriel García Londoño

1. Análisis exploratorio inicial

Variables de entrada/salida

- Para el problema de clasificación:

  - Variables de entrada: WindGustSpeed, WindSpeed9am, WindSpeed3pm, Humidity9am, Humidity3pm, Cloud9am, Cloud3pm, MinTemp, MaxTemp, Rainfall, Evaporation, Sunshine, Pressure9am, Pressure3pm, Temp9am, Temp3pm, Location, WindGustDir, WindDir9am, WindDir3pm, RainToday.

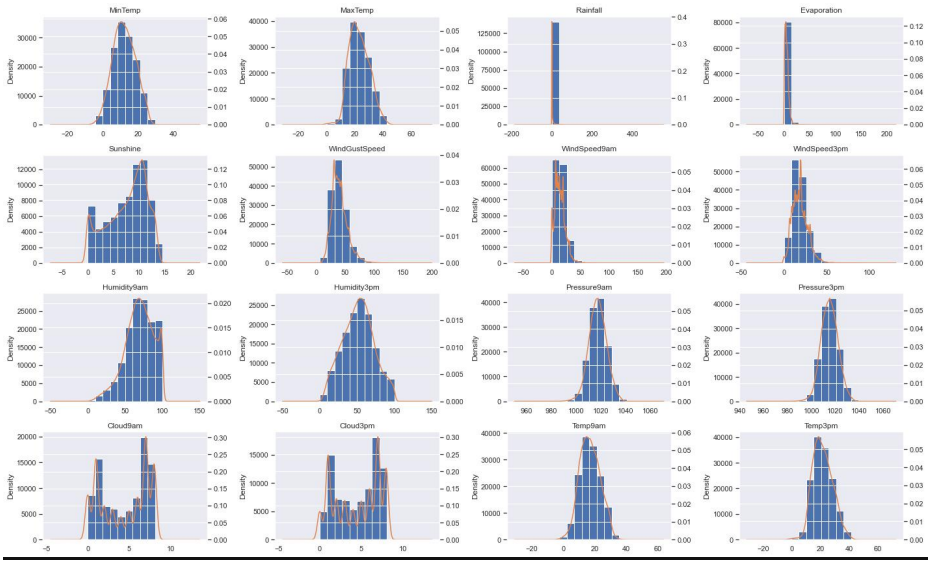
  - Variables de salida: RainTomorrow

- Para el problema de regresión:

  - Variables de entrada: WindGustSpeed, WindSpeed9am, WindSpeed3pm, Humidity9am, Humidity3pm, Cloud9am, Cloud3pm, MinTemp, MaxTemp, Evaporation, Sunshine, Pressure9am, Pressure3pm, Temp9am, Temp3pm, Location, WindGustDir, WindDir9am, WindDir3pm, RainToday, RainTomorrow.

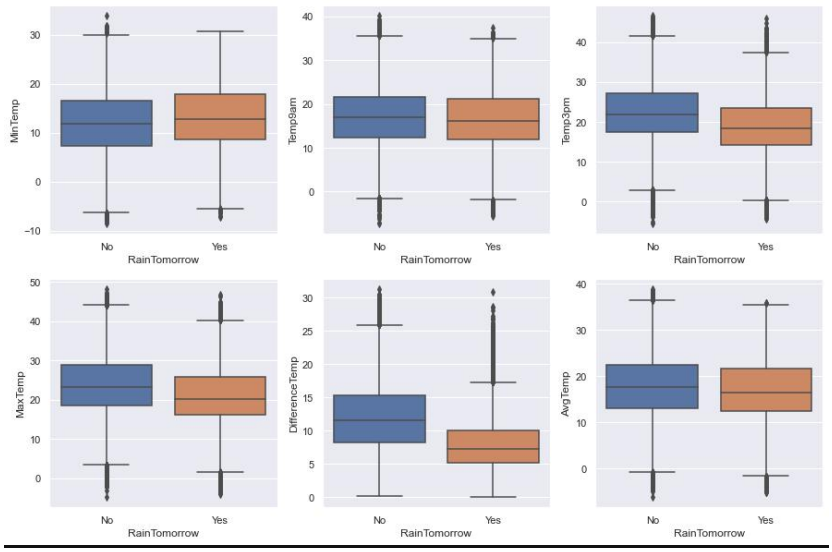
  - Variables de salida: RainfallTomorrow, la cual será creada a partir de la variable RainFall

Distribución variables numéricas:



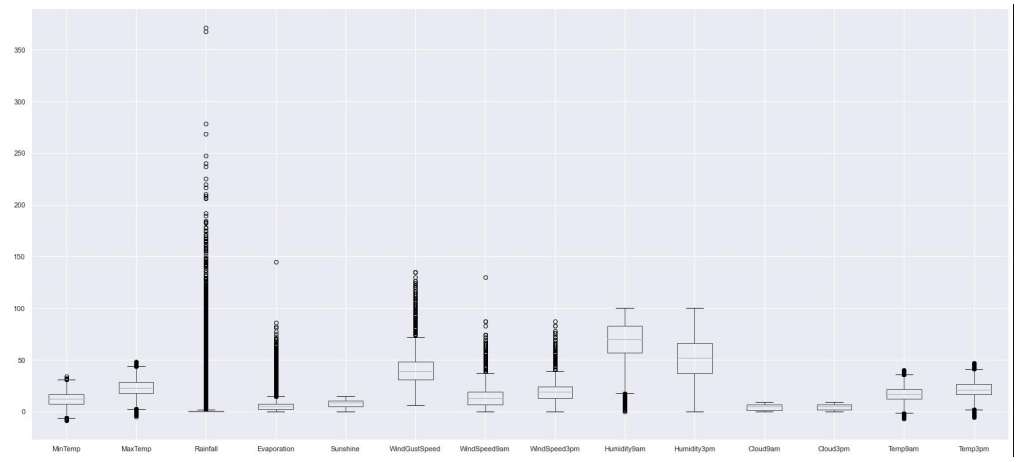
En general se observa que en las variables de temperatura se aprecia cierta normalidad, con un ligero sesgo, en sus respectivas distribuciones. Para Rainfall y Evaporation se observa una alta concentración de su distribución en valores de cero y cercanos a cero. Sunshine tiene sesgo negativo. Todas las variables de viento tienen un ligero sesgo positivo y al parecer con presencia de valores atípicos. Las variables de humedad presenta un cambio significativo dependiendo de la hora de medición, mientras que la distribución de las de presión no cambia con respecto a la hora de medición y finalmente la nubosidad, según lo visto en los histogramas varía muy poco dependiendo de la hora de medición. Además, su distribución parece ser bimodal, ya que sus valores se concentran sobre todo en nubosidades bajas y altas mientras que hay menos casos con nubosidades intermedias.

Análisis bivariado con variable objetivo:



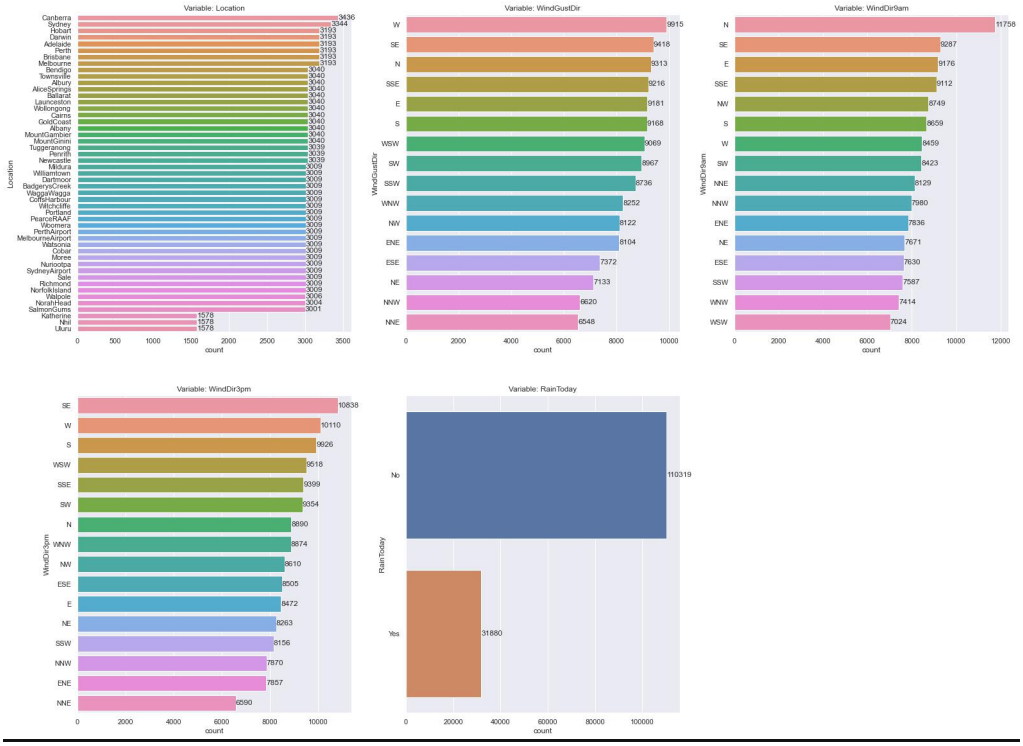
Se realizó un análisis bivariado de la variable objetivo con respecto a las distribuciones de las variables independientes y encontramos que algunas pueden llegar a ser mejores predictoras que otras.

Outliers:



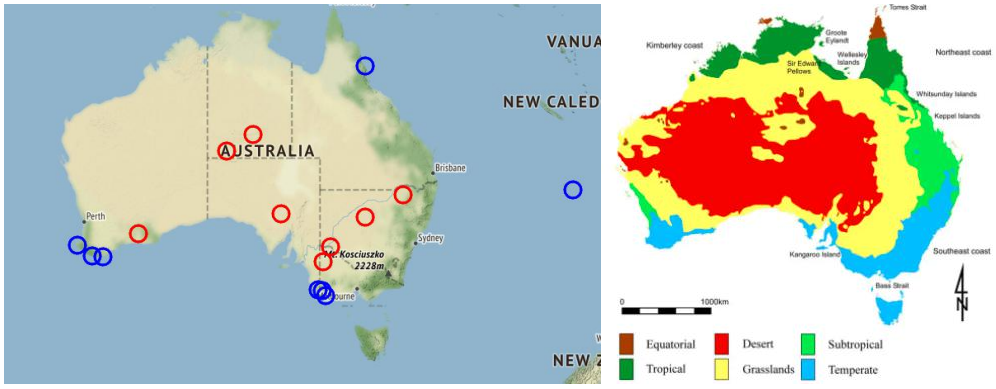
Se hace evidente que el dataset contiene muchos outliers, sobre todo en Rainfall. Pero hay que tener cuidado con los mismos puesto que en este caso un outlier se debe a que llovió con cierta intensidad, y al final es lo que queremos predecir.

Variables categóricas, representatividad de las clases:



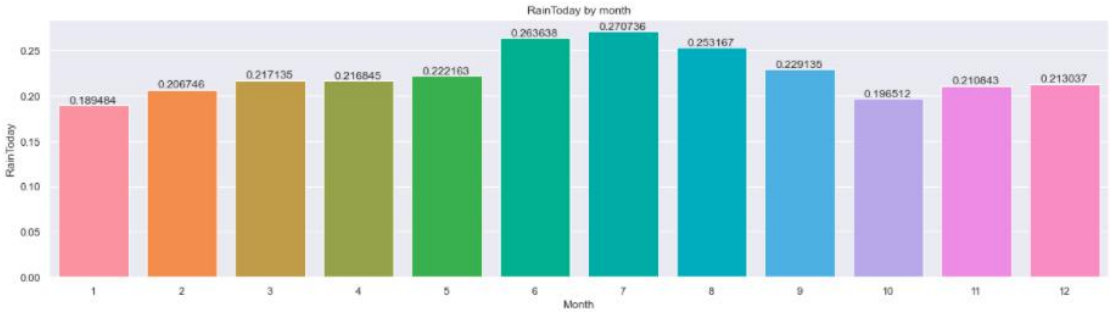
Las variables de dirección de viento muestran distintos valores de importancia en cada tipo de dirección, dependiendo de la hora y de las velocidades. Según la gráfica de la variable Location vemos que Canberra y Sydney (dos de las ciudades más importantes de Australia) tienen mayor representatividad dentro de esta categoría, pero todas las demás ubicaciones, exceptuando a Katherine, Nhil y Uluru, tienen una representatividad similar. Finalmente, para la variable RainToday nos damos cuenta que la variable está imbalanceada, con una relación aproximada de 4 a 1, comparando la cantidad de muestras donde no llovió con las que en donde sí llovió.

Distribución geográfica de lluvias:



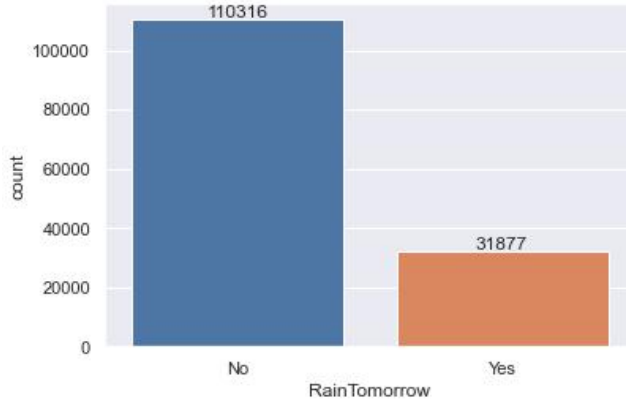
Al observar el mapa, se puede intuir que las zonas insulares y costeras son más propensas a la lluvia comparándolas con las zonas centrales. Al compararla con el mapa del clima en Australia, como era de esperarse, se hace evidente que las zonas con clima templado y tropical son más lluviosas, mientras que en las zonas desérticas, llanas y subtropicales llueve menos.

Distribución temporal de lluvias:



Por mes parece que hay un aumento considerable en las precipitaciones justo a mitad de año.

Distribución Variable de salida:



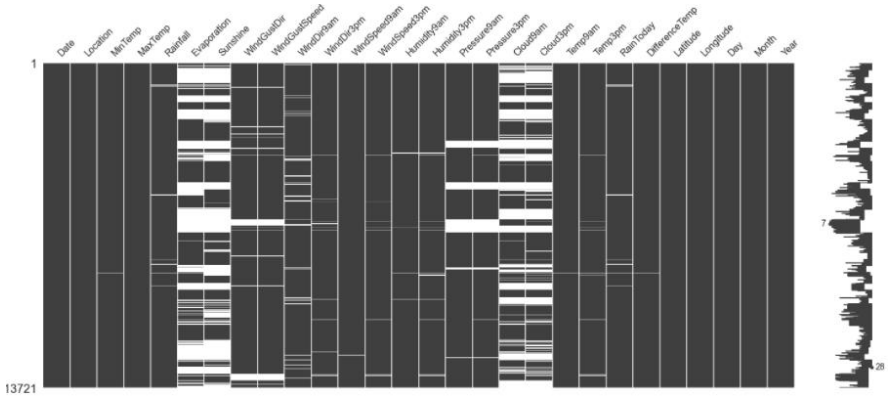
Hay un claro desbalance entre las clases. Considero que sería necesario más adelante emplear alguna técnica de Sobremuestreo de datos, como SMOTE, para balancear las clases y tener resultados mejores.

2. Esquema de validación de resultados.

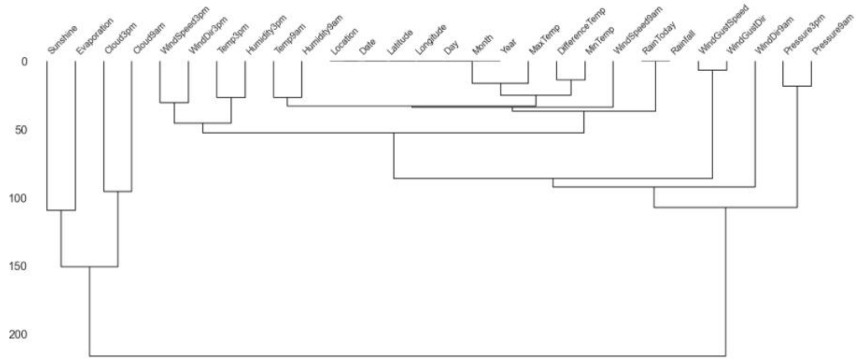
Se particiona el dataset en train/test con una relación 80/20 teniendo en cuenta que se debe pasar como parámetro el stratify para mantener el balance entre las clases. Vale recalcar que la limpieza y preparación de datos se hizo primero en el dataset de train, y luego se replicó en el de test, para evitar leakages.

3. Limpieza y preparación de datos / ingeniería de features

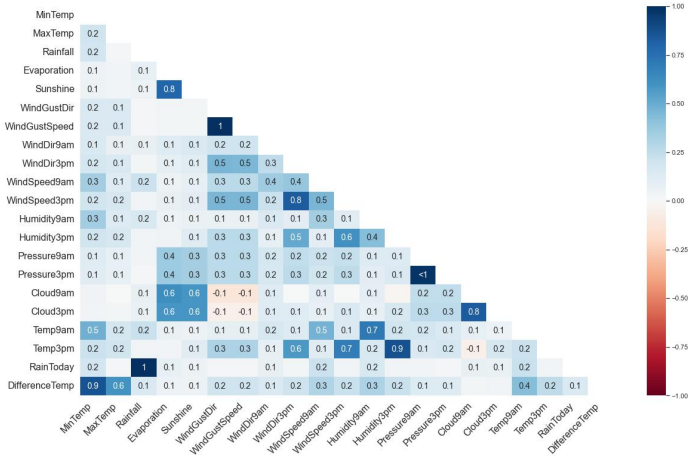
Distribución de valores faltantes:



Dendograma:



Heatmap:



De las gráficas anteriores se puede intuir que cuando la variable Sunshine falta, en un gran porcentaje de casos también faltan las variables Evaporation, Cloud9am y Cloud3pm. Y que luego en menor medida, también está ligado a una falta de las variables de presión. Además, siempre que falta un dato en WindGustSpeed, falta en WindGustDir y a su vez, cuando falta RainToday sí o sí falta Rainfall. Por lo que es muy probable que la falta de esos datos sea MNAR, y que tengan que ver más bien con la ubicación donde se miden estos datos.

Transformación e imputación de variables

Variables categóricas:

Fecha

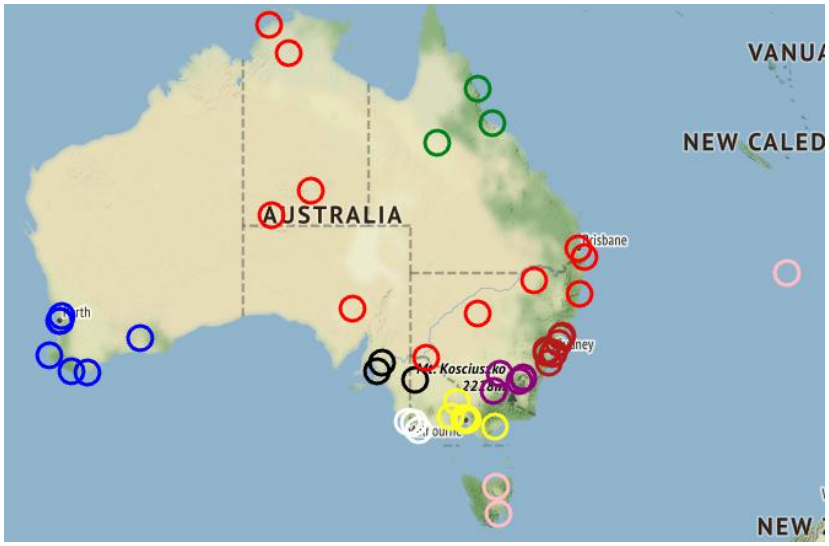
En la fecha no hay valores nulos. Dado que no se observó relación por día o por año con las precipitaciones, solo se tomará el mes y se le aplicará OHE. También se probará con la estrategia de transformación cíclica, ya que el período en meses vuelve a repetirse año tras año.

Direcciones de viento:

Las direcciones de viento faltantes (<7% de registros) se imputaron por el valor de la moda dependiendo de la ciudad, para no afectar tanto la representatividad por clase original de las variables. Para su codificación empleé una codificación numérica aprovechando que cada dirección del viento se puede reemplazar por su equivalente en grados y además, dada la naturaleza cíclica de las direcciones del viento, hice transformaciones cíclicas para cada variable.

Ubicación:

Se recuerda que en esta variable no se requiere imputación, debido a que tenemos el 100% de las ubicaciones con data. Para la codificación Inicialmente, se utilizó la API Nominatim del paquete Geopy que devuelve las coordenadas de Latitud y Longitud según el nombre de la ciudad. Luego pensé en clusterizar las coordenadas geográficas como en todo problema de clasificación no supervisada, tenía pensado aplicar k-means, pero luego de leer literatura al respecto entendí que k-means no es un buen approach para este tipo de problemas donde se involucran variables geográficas latitud y longitud. Dentro de la literatura vista recomiendan utilizar métodos de clusterización como Distancia definida con autoajuste HDBSCAN y/o Escala múltiple OPTICS. Por practicidad empleé únicamente HDBSCAN y se deja como trabajo futuro emplear otras técnicas para saber si mejora el clusterizado. Dado que el clustering consideró las islas dentro del cluster de outliers, creé un cluster adicional con las islas a parte.



Por último, apliqué OHE, que ahora con las ubicaciones sí será útil, puesto que solo tenemos 8 dimensiones más, en vez de 49.

Variables numéricas:

Imputación:

Debido al gran número de valores faltantes en las variables numéricas, para su imputación emplearé la técnica MICE. Pero teniendo en cuenta todas las demás columnas numéricas originales del dataset, debido a que no altera significativamente la distribución de los datos y ofrece mejores resultados comparándola con la imputación estadística. Se dejará como trabajo futuro emplear otra técnica como imputación por KNN.

Transformación:

Se creó una nueva variable basada en la diferencia entre la máxima y la mínima temperatura. No se decide hacer tratamiento de outliers ni transformación de variables numéricas debido a que considero que no se puede forzar la data a seguir una distribución normal, cuando originalmente no la tiene.

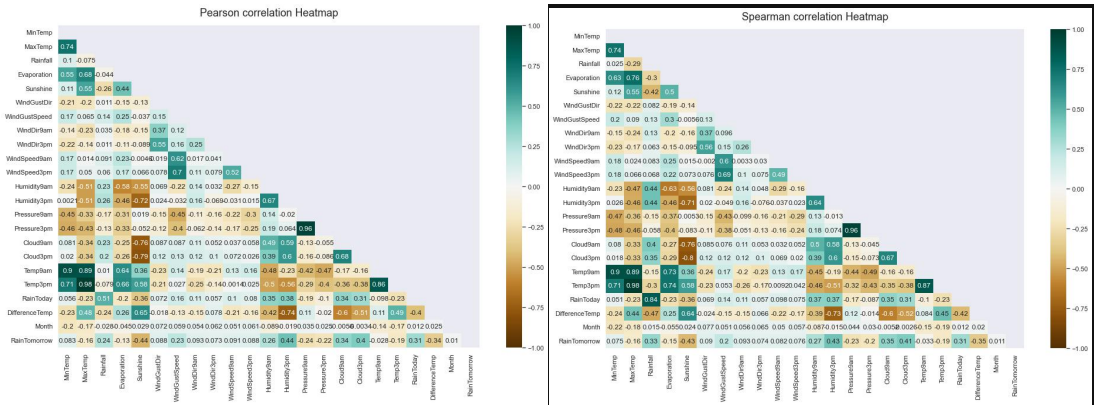
Al final el dataset de train queda sin valores faltantes:

```
# Comprobamos si hay nulos en el dataset de train
X_train.isna().sum().sum()

0
```

Variable finales:

```
Index(['MinTemp', 'Rainfall', 'Evaporation', 'Sunshine', 'WindGustDir',
       'WindGustSpeed', 'WindDir9am', 'WindDir3pm', 'WindSpeed9am',
       'WindSpeed3pm', 'Humidity9am', 'Humidity3pm', 'Pressure9am', 'Cloud9am',
       'Cloud3pm', 'Temp3pm', 'RainToday', 'DifferenceTemp', 'Enero',
       'Febrero', 'Marzo', 'Abril', 'Mayo', 'Junio', 'Julio', 'Agosto',
       'Septiembre', 'Octubre', 'Noviembre', 'Diciembre', 'month_sin',
       'month_cos', 'WindGustDir_sin', 'WindGustDir_cos', 'WindDir9am_sin',
       'WindDir9am_cos', 'WindDir3pm_sin', 'WindDir3pm_cos', 'cluster_-1',
       'cluster_0', 'cluster_1', 'cluster_2', 'cluster_3', 'cluster_4',
       'cluster_5', 'cluster_6', 'cluster_7'],
      dtype='object')
```

Análisis de correlaciones:

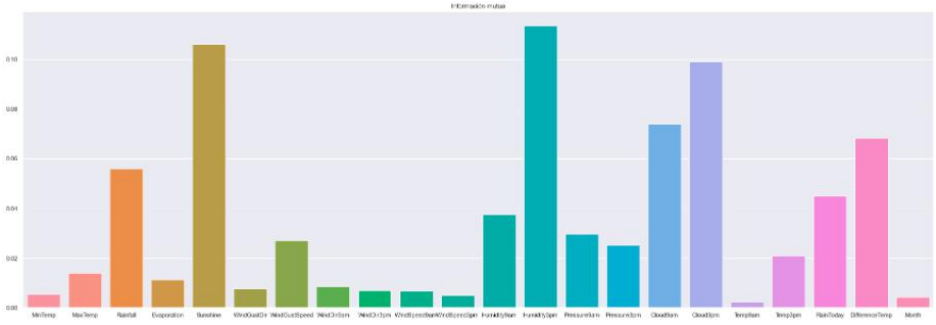


Al observar el mapa de correlaciones de person observamos que:

- MinTemp con Temp9am y MaxTemp con Temp3pm tienen una altísima correlación lineal positiva(>=.9)
- Pressure9am y Pressure3pm tienen una altísima correlación lineal positiva(.96)
- Nubosidad y Sunshine están correlacionados linealmente negativos (-.76)
- Humidity3pm y cloud3pm tienen una ligera correlación lineal positiva con nuestra variable objetivo (>=.4)
- Sunshine tiene una ligera correlación lineal negativa con nuestra variable objetivo (-.44)

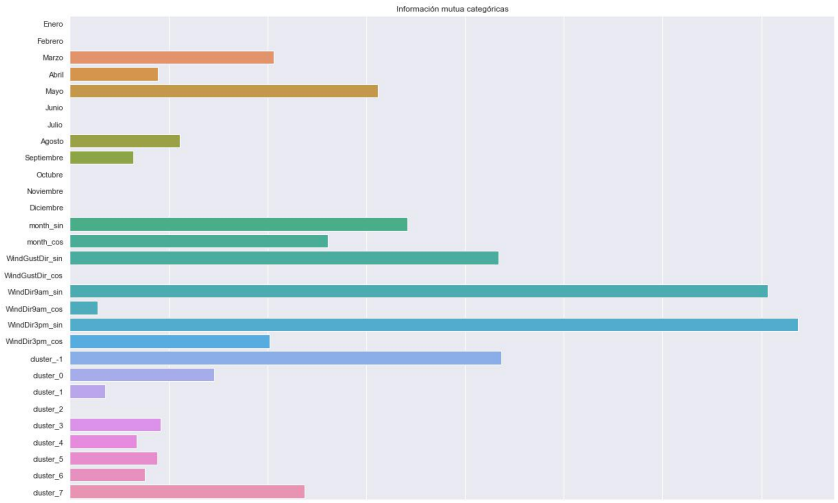
Dado que Pearson solo mide correlaciones lineales, es conveniente mirar otros enfoques como Spearman, que mide relaciones monotónicas y también el criterio de información mutua que incluso llega a medir relaciones no lineales.

Información mutua variables numéricas:



Con este criterio podemos observar que variables como Rainfall y DifferenceTemp se suman a las variables correlacionadas con RainTomorrow.

Información mutua variables categóricas:

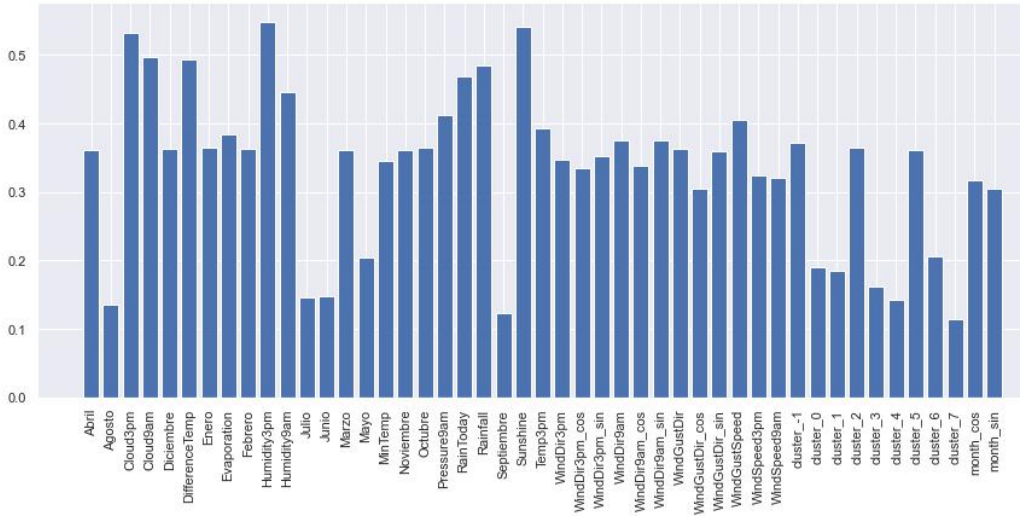


De la gráfica anterior se infirió que:

- La transformación cíclica seno del mes tiene mejor score de información mutua compara con el OHE del mes.
- Las transformaciónes cíclicas seno de las direcciones de viento son mejores predictoras que las coseno.
- El cluster_-1 que contenía outliers (ubicaciones centrales y desérticas en australia) es el mejor predictor entre todos los clústeres.
- El cluster_7 que contiene las ubicaciones insulares, también es un buen predictor para la variable objetivo.

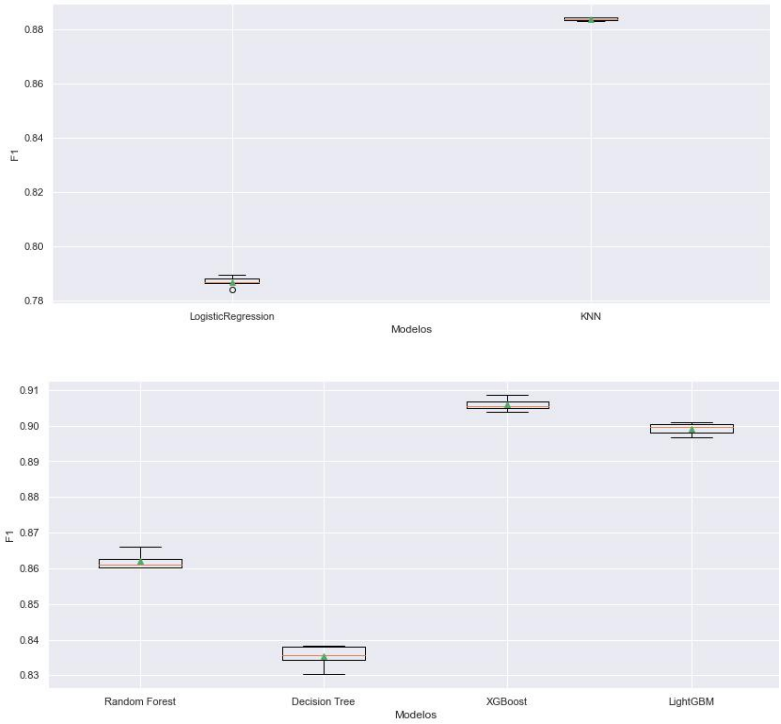
Selección de features:

Se empleó la librería Feature Engine y se eliminaron las variables altamente correlacionadas, seleccionando la que mejor se desempeñaba en un modelo de RandomForest. Además, se empleó una selección de características por desempeño unitario, en donde se evaluaba cada feature en un modelo de ML, para determinar su importancia. Los resultados fueron los siguientes:



4. Entrenamiento de modelos:

Se eliminaron las variables que se habían codificado y las que tenían alta correlación. Se entrenaron modelos lineales, de clustering y basados en árboles. Teniendo mejor desempeño los modelos tipo boosting como XGBoost y LightGBM.



Para evaluar el modelo empleé LightGBM debido a que es más eficiente computacionalmente que XGBoost.

5. Evaluación de resultados y conclusiones.

Primero se replicó el proceso de transformación y limpieza de datos en el dataset de test. Luego se entrenaron modelos de LightGBM con las siguientes consideraciones:

1. LightGBM con datos balanceados empleando SMOTE.

2. LightGBM con datos sin balancear, pero cambiando parámetro class_weight = 'balanced' y max_depth a 12

3. LightGBM haciendo selección de features según criterio de SingleFeaturePerformance

4. LightGBM empleando utilizando Imbalanced Pipeline y SMOTE.

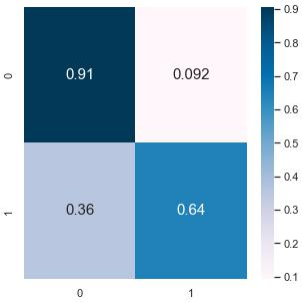
Al final quedan dos candidatos como ganadores, dependiendo de la métrica con la que evaluemos el modelo.

Si nos basamos en recall el ganador sería el modelo 2, ya que tiene un valor de .86:

```
Confusion matrix
 [[16089  5967]
 [  865  5510]]
              precision    recall  f1-score   support

         0.0       0.95      0.73      0.82     22056
         1.0       0.48      0.86      0.62      6375

    accuracy                           0.76     28431
   macro avg       0.71      0.80      0.72     28431
weighted avg       0.84      0.76      0.78     28431
```

Si nos basamos en F1 score, el ganador sería, con un score de 0.66.

```
Confusion matrix
 [[20020  2036]
 [ 2270  4105]]
              precision    recall  f1-score   support

         0.0       0.90      0.91      0.90     22056
         1.0       0.67      0.64      0.66      6375

    accuracy                           0.85     28431
   macro avg       0.78      0.78      0.78     28431
weighted avg       0.85      0.85      0.85     28431
```



Conclusiones finales:

- Si queremos predecir si lloverá el mejor desempeño del modelo se da empleando el dataset total sin aplicar SMOTE basándonos en el Recall, utilizando como modelo LightGMB, con máxima profundidad de 12 y class_weight = True.

- Si queremos una prediccion más precisa entre si lloverá o no, es mejor seleccionar el último modelo en donde se empleó un Pipeline imbalanceado, SMOTE y LightGBM con máxima profundidad de 12 y class_weight = True.

6 Predicción de RainfallTomorrow

Para esta predicción emplearemos las mismas variables que en el ejercicio anterior, con la única diferencia de que agregamos RainfallTomorrow como variable explicativa. También cambiaremos los modelos y las métricas para evaluarlo. Primero probé con una regresión lineal y luego con el caballo de troya :) LightGBM en este caso pasa un problema de regresión.

Resultados:

Regresión lineal:

```
MSE: 51.47043795419655
RMSE: 7.174290066215371
R2 Square 0.32213223212527153
```

LightGBM Regressor con optimización de hiperparámetros basado en métrica de evaluación L1:

```
MSE: 39.91817028589919
RMSE: 6.318082801443741
R2 Square 0.47427606865467586
```

Comparando las métricas entre los modelos utilizados se ve claramente que el MSE Y RMSE bajó considerablemente empleando LightGBM, mientras que el R2 mejoró. Por lo que el modelo ganador en este caso es LightGBM regressor.