Universidad Francisco Marroquín
Christian Medina Armas
Machine Learning Model

Proyecto Final -
Rain prediction model

Stephanie Grotewold
20210567
16 de mayo del 2024

Introducción

En este informe se analizan los datos relacionados con la lluvia, centrándose en variables meteorológicas. Estas variables incluyen información como temperaturas mínimas y máximas, cantidad de lluvia registrada, evaporación, horas de sol, dirección y velocidad del viento, humedad relativa, presión atmosférica, nubosidad y temperaturas a diferentes horas del día. También se incluyen variables binarias que indican si ha llovido hoy y si se espera lluvia para el día siguiente, que es la variable objetivo del problema de predicción. El objetivo de este proyecto es construir un modelo de aprendizaje automático que pueda predecir de manera confiable si lloverá o no al día siguiente en función de las condiciones climáticas históricas.

Data

El conjunto de datos "*RainTomorrow*" consta de múltiples instancias relacionadas con observaciones meteorológicas a lo largo de aproximadamente 10 años en varias ubicaciones de Australia. Incluye un total de 22 características. Entre las variables categóricas se incluyen la dirección del viento (WindGustDir, WindDir9am, WindDir3pm) y la variable binaria "RainToday", que indica si ha llovido en el día actual. Por otro lado, las variables numéricas abarcan la temperatura mínima y máxima (MinTemp, MaxTemp), la cantidad de lluvia registrada (Rainfall), la evaporación (Evaporation), las horas de sol (Sunshine), la velocidad del viento (WindGustSpeed, WindSpeed9am, WindSpeed3pm), la humedad relativa (Humidity9am, Humidity3pm), la presión atmosférica (Pressure9am, Pressure3pm), la nubosidad (Cloud9am, Cloud3pm) y las temperaturas en diferentes momentos del día (Temp9am, Temp3pm).

Métodos

Análisis de datos numéricos

Los datos revelan una amplia variabilidad en las variables numéricas del conjunto, evidenciada por extremos como temperaturas que van desde -8.2°C hasta 33.9°C, y cantidades de lluvia que oscilan entre 0 mm y 371 mm en un solo día. La velocidad máxima de ráfagas de viento varía entre 6 km/h y 135 km/h, con un promedio de 40 km/h. La presión atmosférica a las 9 am y 3 pm tiene promedios de 1017.6 hPa y 1015.3 hPa. La temperatura a las 3 pm suele ser más alta que a las 9 am debido al calentamiento diurno. En cuanto a la velocidad del viento, puede variar sin una tendencia clara de ser siempre más alta o más baja a esa hora. La humedad relativa tiende a ser más baja a las 3 pm en climas cálidos. Por último, la presión atmosférica puede variar según las condiciones locales, sin seguir una tendencia específica de ser más alta o más baja a las 3 pm.

Análisis de datos categóricos

El conjunto de datos abarca un período considerable desde el 1 de noviembre de 2007 hasta el 25 de junio de 2017, lo que ofrece una amplia perspectiva temporal para el análisis meteorológico. Al destacar que Cairns registró la mayor cantidad de lluvia con 16429.0 mm, y los meses más lluviosos fueron Marzo, Enero y Junio con 35669.8 mm, 34483.4 mm y 33275.2 mm respectivamente, se subraya la variabilidad estacional y geográfica de los patrones climáticos en los datos. Los cinco lugares más frecuentes en el conjunto de datos son Canberra, Sydney, Hobart, Brisbane y Melbourne. La dirección de viento predominante a las 9 am es norte (N), mientras que a las 3 pm es sureste (SE). Los tres años con mayor cantidad de lluvia son 2010, el año con el registro más alto de lluvia, seguido por 2011 y 2016.

Análisis de normalidad

Los resultados de las pruebas de normalidad muestran que la mayoría de las variables no siguen una distribución normal. Esto se evidencia por los valores significativamente bajos de p obtenidos en la prueba de D'Agostino's K^2 Test y las estadísticas de prueba de Anderson-Darling que son considerablemente mayores que los valores críticos. En contraste, las variables Pressure9am y Pressure3pm también muestran cierta falta de normalidad, aunque en menor medida en comparación con las demás variables.

Matriz de correlación

La matriz de correlación revela diversas relaciones entre las variables meteorológicas analizadas. Las mediciones de temperatura (MinTemp, MaxTemp, Temp9am, Temp3pm) muestran una fuerte correlación positiva entre sí, indicando que los cambios en una de estas temperaturas suelen reflejarse en las demás de manera consistente. Por otro lado, la precipitación (Rainfall) no presenta una correlación significativa con las temperaturas ni con otras variables estudiadas, sugiriendo que su comportamiento puede estar influenciado por factores distintos a las condiciones térmicas o de presión atmosférica. Las velocidades del viento (WindGustSpeed, WindSpeed9am, WindSpeed3pm) exhiben correlaciones moderadas a altas entre sí, indicando una cierta estabilidad en la intensidad del viento a lo largo del día. Asimismo, la humedad relativa (Humidity9am, Humidity3pm) y la presión atmosférica (Pressure9am, Pressure3pm) muestran correlaciones significativas entre sus mediciones matutinas y vespertinas, lo que sugiere patrones consistentes en estos parámetros durante el día. En cuanto a la nubosidad (Cloud9am, Cloud3pm), su correlación moderada indica que la fracción de cielo cubierto por nubes tiende a mantener cierta persistencia a lo largo de la jornada, aunque pueda haber variaciones según otros factores climáticos.

Variance Inflation Factor

El VIF es una medida que se utiliza para identificar la multicolinealidad en un modelo de regresión. En este caso, las variables MaxTemp, Temp3pm, Pressure9am, Pressure3pm, Temp9am y Humidity3pm tienen un VIF por encima de 5, llegando incluso hasta un valor de 57. Esto indica un problema de multicolinealidad, lo que puede hacer que los coeficientes de regresión sean inestables y difíciles de interpretar.

Modelos

Random Forest: Los bosques aleatorios son una colección de árboles de decisión. Cada árbol en el bosque aleatorio escupe una predicción de clase y la clase con más votos se convierte en la predicción del modelo. Los bosques aleatorios evitan el problema de *overfitting* que enfrentan los árboles de decisión. Maneja automáticamente la importancia de las características y reduce la tendencia de sobreajuste, sin embargo, es más complejo computacionalmente.

XGBoost: Funciona construyendo múltiples árboles de decisión secuenciales para corregir errores y optimiza una función de pérdida usando el algoritmo de descenso de gradiente. Sus ventajas incluyen manejo de datos faltantes, eficiencia computacional y prevención de sobreajuste. Sin embargo, su configuración de hiperparámetros puede ser compleja y su interpretación menos directa debido a su enfoque de ensamblaje de árboles.

Logistic Regression: La regresión logística predice la probabilidad de eventos binarios usando variables predictoras y una función logística. Durante el entrenamiento, ajusta coeficientes para minimizar las diferencias con los datos reales. Es fácil de interpretar, eficiente con grandes datos, maneja multicolinealidad y no requiere normalidad en los datos. Sin embargo, es limitada a clasificación binaria y puede ser sensible a valores atípicos.

SVM: En este algoritmo, se traza cada elemento de datos como un punto en el espacio n-dimensional, donde n es el número de características, con el valor de cada característica siendo el valor de una coordenada particular. Tiende a tener un buen rendimiento en conjunto de datos medianos, pero es sensible a la escala de las características y requiere hiperparámetros.

Pasos a seguir de cada algoritmo:

1. **Limpieza de datos**: Eliminar filas con valores nulos en la columna 'RainTomorrow'. Codificación de la variable objetivo: Utilizar LabelEncoder para convertir la variable objetivo en valores numéricos.
2. **Definición de características y objetivo**: Identificar características (X) y variable objetivo (y).
3. **Preprocesamiento de datos**:
    a. Para características numéricas:
        i. Imputación de valores faltantes utilizando SimpleImputer.
        ii. Escalamiento de características utilizando StandardScaler.
    b. Para características categóricas:
        i. Imputación de valores faltantes utilizando SimpleImputer.
        ii. Codificación one-hot de características utilizando OneHotEncoder.
4. **Pipeline de procesamiento:** Combinar todas las etapas de preprocesamiento en un pipeline utilizando Pipeline o ColumnTransformer.
   Incluir el clasificador específico para cada modelo en el pipeline:
    a. Regresión Logística: LogisticRegression
    b. XGBoost: XGBClassifier
    c. SVM: SVC
    d. Random Forest: RandomForestClassifier
5. **División de datos de entrenamiento y prueba**: Utilizar train_test_split para dividir los datos en conjuntos de entrenamiento y prueba.
6. **Entrenamiento del modelo**: Ajustar el pipeline de procesamiento y el modelo de clasificación en los datos de entrenamiento. Predecir las etiquetas de clase en los datos de prueba.
7. **Evaluación:** Evaluar el rendimiento del modelo utilizando métricas como accuracy y F1-score.

Resultados

La métrica principal fue F1, que combina precisión y exhaustividad en una sola medida. Esto la hace útil para evaluar el equilibrio entre identificar correctamente casos positivos y evitar falsas alarmas. Una puntuación F1 más alta indica un mejor rendimiento del modelo en términos de equilibrio entre precisión y exhaustividad.

	Yes	**No**
XGBoost	59%	93%
SVC	58%	93%
Random Forest	71%	89%
Logistic Regression	52%	92%

Se intentó utilizar la búsqueda en cuadrícula para maximizar el F1 en tres modelos: SVC, Random Forest y XGBoost. En el caso de SVC, el proceso se interrumpió después de 1000 minutos sin lograr completarse. Sin embargo, al omitir la búsqueda en cuadrícula, se obtuvieron métricas similares a los modelos con hiperparámetros adicionales.

En el caso de Random Forest, se observó un F1 inicial del 64.37%, el cual experimentó una mejora significativa al alcanzar un valor del 65.65% mediante la optimización de hiperparámetros. Los parámetros ajustados durante este proceso fueron 'classifier__max_depth': None, 'classifier__min_samples_leaf': 4, 'classifier__min_samples_split': 2, 'classifier__n_estimators': 300. Este ajuste llevó un tiempo total de ejecución de 55 minutos y 37.5 segundos, pero resultó en una mejora notable en la métrica de evaluación del modelo.

En el caso de XGBoost, se logró mejorar el promedio del F1 del 66% al 67.16% mediante la optimización de hiperparámetros. Siendo 59% para los “Yes” y 93% para “No”.Durante este proceso, se exploraron diferentes configuraciones clave. Se ajustó el learning_rate a 0.2 para controlar la velocidad de aprendizaje del modelo. Además, se fijó la profundidad máxima en 7 para capturar relaciones complejas. Se determinó que el número óptimo de estimadores en el modelo era de 300. También se definió una fracción de muestras utilizadas para entrenar cada árbol (subsample) en 0.7. Estos cambios condujeron a una mejora significativa en la métrica de evaluación. El proceso completo de optimización tomó un tiempo de ejecución de 11 minutos y 56.7 segundos.

Conclusión

En este informe se analizan detalladamente los datos relacionados con la predicción de lluvia basados en variables meteorológicas, destacando que hubo bastantes observaciones faltantes en el conjunto de datos. Casi ninguna de las variables seguía una distribución normal, lo cual planteó desafíos adicionales en el proceso de análisis y modelado. Se exploraron múltiples características que abarcan desde temperaturas extremas hasta direcciones del viento y condiciones de humedad, con el objetivo de construir un modelo de aprendizaje automático confiable para predecir si lloverá al día siguiente.

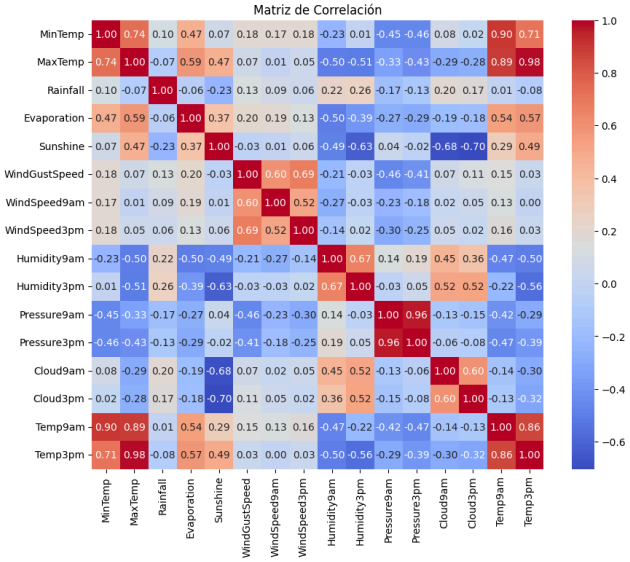
Durante el análisis, se utilizaron diferentes algoritmos de aprendizaje automático, destacando especialmente XGBoost como el mejor modelo para este problema. XGBoost logró mejorar la métrica F1 del 66% al 67.12% mediante la optimización de hiperparámetros. Las ventajas clave de XGBoost, como su capacidad para manejar datos faltantes, eficiencia computacional y prevención de sobreajuste, fueron decisivas en su elección como el modelo más efectivo para esta tarea.

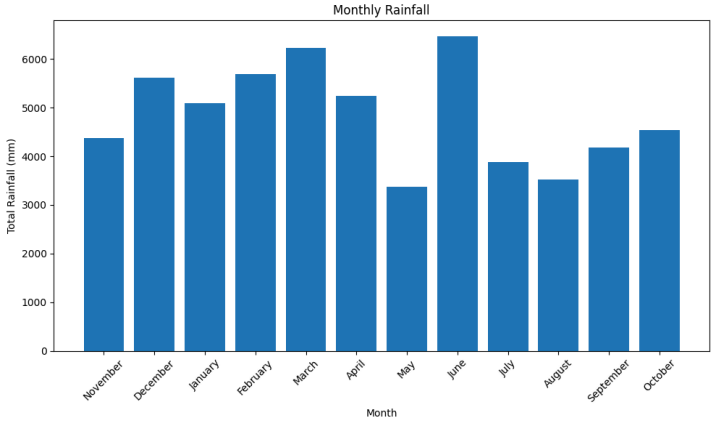
El proceso de optimización implicó ajustar parámetros importantes como la velocidad de aprendizaje (learning_rate), la profundidad máxima de los árboles (max_depth), el número de árboles en el modelo (n_estimators) y la fracción de muestras utilizadas para entrenar cada árbol (subsample). Estos ajustes condujeron a una mejora notable en la capacidad de predicción del modelo, validada mediante métricas como F1-score y accuracy.

Bibliografía

- *sklearn.model_selection.GridSearchCV*. (n.d.). Scikit-learn. https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html#sklearn.model_selection.GridSearchCV
- *sklearn.metrics.f1_score. (n.d.). Scikit-learn.* https://scikit-learn.org/stable/modules/generated/sklearn.metrics.f1_score.html
- *1.11. Ensembles: Gradient boosting, random forests, bagging, voting, stacking. (n.d.). Scikit-learn.* https://scikit-learn.org/stable/modules/ensemble.html

Anexos






```
✓ 55m 37.5s

/var/folders/57/c9zjxddn221c056sd800540w0000gn/T/ipykernel_2674/935648650.py:15: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  df_cleaned['RainTomorrow'] = label_encoder.fit_transform(df_cleaned['RainTomorrow'])
/Users/stephgrotewold/anaconda3/envs/ml/lib/python3.10/site-packages/joblib/externals/loky/process_executor.py:752: UserWarning: A worker stopped while so
  warnings.warn(
Best parameters: {'classifier__max_depth': None, 'classifier__min_samples_leaf': 4, 'classifier__min_samples_split': 2, 'classifier__n_estimators': 300}
Accuracy: 0.8429746078893035
F1-score: 0.6565792600808049
```

```python
    # Save the best model to a pickle file
    with open('Stephanie_Grotewold_fin.pkl', 'wb') as f:
        pickle.dump(best_model, f)

    print(f"Accuracy: {accuracy}")
    print(f"F1-score: {f1}")
```

✓ 11m 56.7s

```
Fitting 8 folds for each of 81 candidates, totalling 648 fits
Best parameters: {'classifier__learning_rate': 0.2, 'classifier__max_depth': 7, 'classifier__n_estimators': 300, 'classifier__subsample': 0.7}
Accuracy: 0.8305281205476805
F1-score: 0.6716264271008796
```