Universidad del Valle de Guatemala
Facultad de ingeniería

Data Science
Laboratorio 10
ChatGPT y Data Science

Andrés de la Roca
Jun Woo Lee

Guatemala, 2023

Exploración y procesamiento de datos

la base de datos de vivienda de la Encuesta Nacional de Condiciones de Vida (ENCOVI) 2014 incluye las siguientes variables:

- Región: Representa la región administrativa a la que pertenece la unidad de muestreo.
- Depto: Indica el departamento al que pertenece la unidad de muestreo.
- Area: Describe el tipo de área en la que se encuentra la unidad de muestreo (puede ser urbana o rural).
- UPM: Corresponde a la Unidad Primaria de Muestreo, una entidad utilizada en la metodología de la encuesta.
- NUMHOG: Número de hogar, que sirve como identificador único del hogar en la base de datos.
- Factor: Es el factor de expansión de hogares con agregado de consumo, utilizado para ajustar las cifras y hacer inferencias a nivel de población.
- Factor 3: Es el resultado de multiplicar el factor por el número de personas en el hogar.
- Pobreza: Indica la clasificación del hogar en términos de pobreza.
- THOGAR: Representa el tamaño del hogar, es decir, el número de personas que lo componen.

Estos datos proporcionan información detallada sobre la distribución geográfica, características del hogar y factores socioeconómicos relacionados con la pobreza en el contexto de la Encuesta Nacional de Condiciones de Vida realizada en 2014.

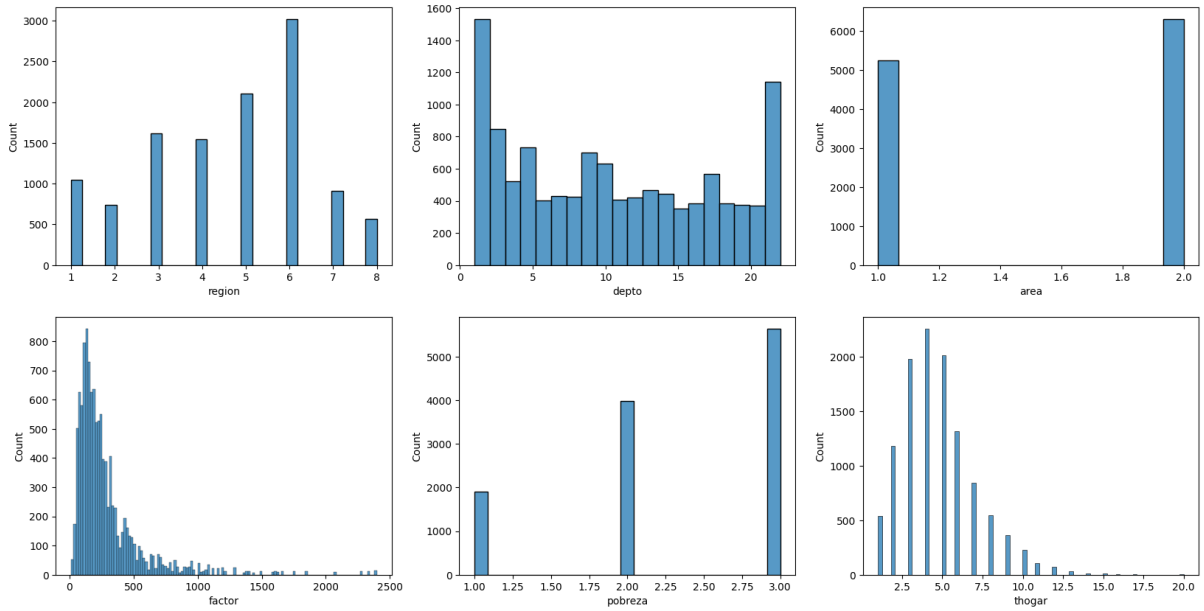
Análisis univariable

Tras realizar la limpieza de variables de la base de datos se encontraron las siguientes distribuciones estadisticas para cada una de las variables individualmente

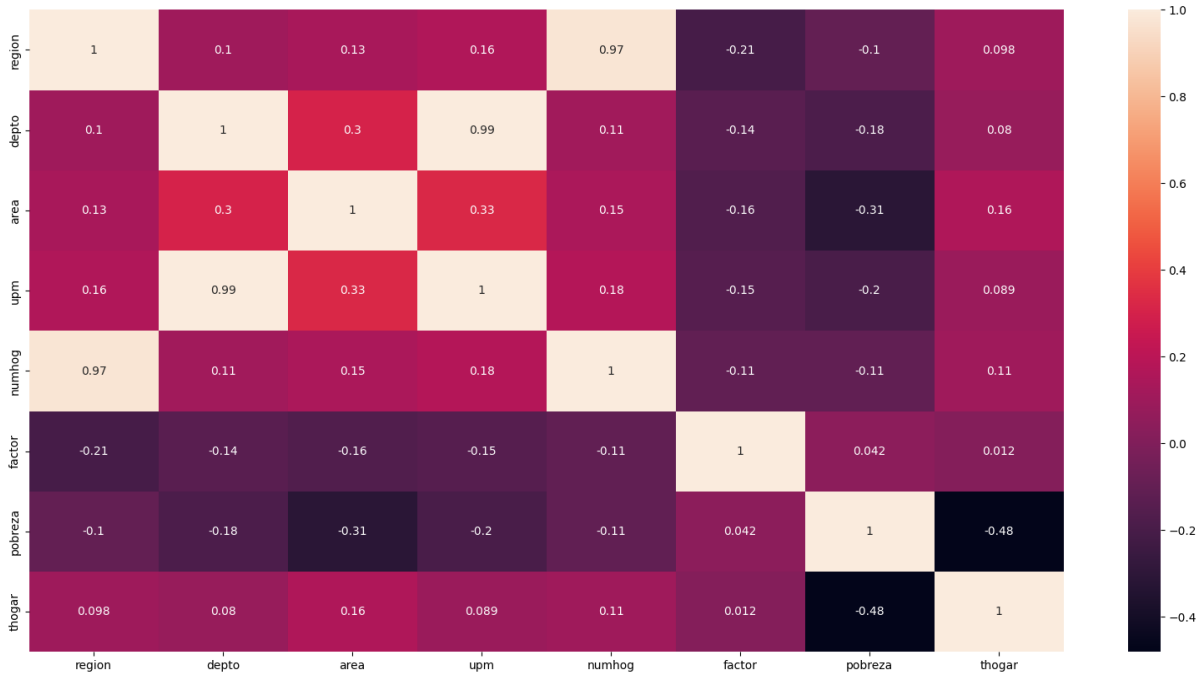
	region	depto	area	upm	numhog	factor	pobreza	thogar
count	54822	54822	54822	54822	54822	54822	54822	54822
mean	4.694849	10.708894	1.582412	531.359290	5948.701215	291.890190	2.142461	5.919631

std	1.883416	6.518859	0.493166	292.372677	3337.015133	281.990073	0.774229	2.638153
min	1.000000	1.000000	1.000000	1.000000	1.000000	13.000000	1.000000	1.000000
25%	3.000000	5.000000	1.000000	283.000000	3069.250000	130.000000	2.000000	4.000000
50%	5.000000	10.000000	2.000000	542.000000	6049.000000	209.000000	2.000000	5.000000
75%	6.000000	16.000000	2.000000	776.000000	8882.000000	341.000000	3.000000	7.000000
max	8.000000	22.000000	2.000000	1037.000000	11536.000000	2398.000000	3.000000	20.000000

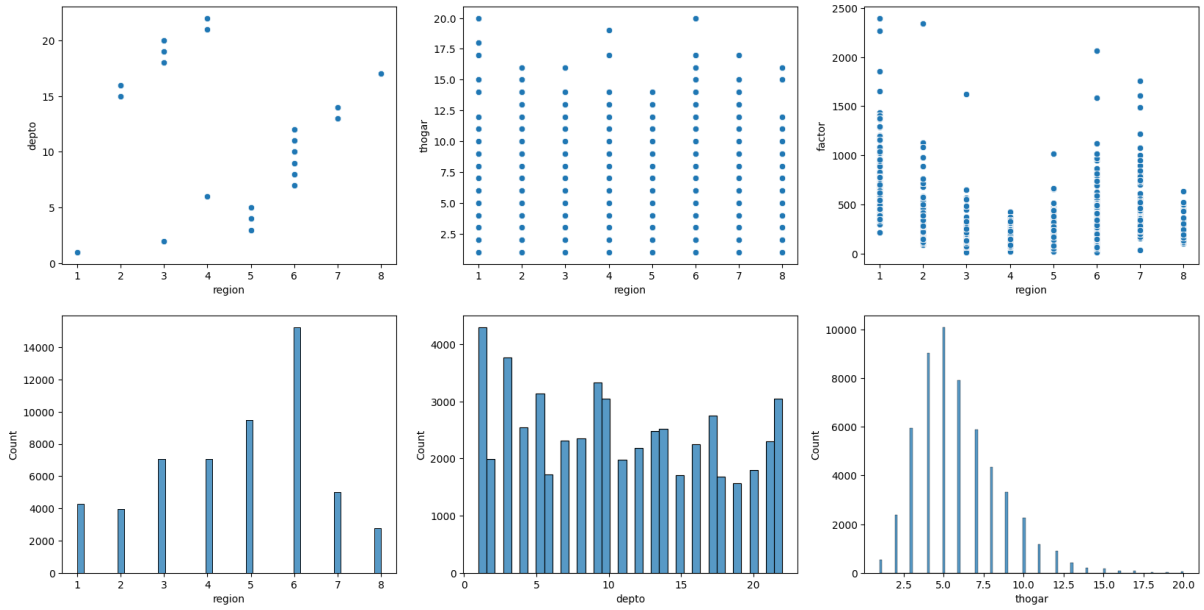
La base de datos de la Encuesta Nacional de Condiciones de Vida (ENCOVI) 2014 proporciona una visión detallada de diversas variables clave. En términos de distribución geográfica el factor de expansión (Factor) varía desde 13 hasta 2398, con una media de 291.89 y una desviación estándar de 281.99. La variable Pobreza clasifica los hogares en términos de pobreza, siendo mayormente de tipo 2 (media de 2.14, desviación estándar de 0.77). El tamaño del hogar (THOGAR) tiene un rango de 1 a 20, con una media de 5.92 y una desviación estándar de 2.64. Estos estadísticas resumen la variabilidad y distribución de las variables clave en la base de datos ENCOVI 2014, ofreciendo una comprensión detallada de la diversidad socioeconómica y demográfica de la muestra.



Análisis bivariable



Al realizar un análisis bivariable de la Encuesta Nacional de Condiciones de Vida (ENCOVI) 2014, se exploraron relaciones significativas entre las variables clave. Se observa que las regiones administrativas (Region) varían en promedio según el tamaño del hogar (THOGAR), sugiriendo posibles disparidades demográficas entre distintas áreas geográficas. Además, la clasificación de pobreza (Pobreza) parece tener una relación inversa con el factor de expansión (Factor), indicando que los hogares con mayores factores podrían tener una menor propensión a la pobreza. La distribución de áreas (Area) sugiere que las zonas urbanas tienden a tener un mayor número de departamentos (Depto), apuntando a una concentración demográfica en entornos urbanos. Estos hallazgos destacan la complejidad de las interacciones entre variables socioeconómicas y geográficas en el contexto de la ENCOVI 2014, proporcionando una base para análisis más profundos y políticas informadas que aborden las disparidades observadas en la encuesta.



Preparación de Datos

En la fase de preparación de los datos de nuestro análisis de los conjuntos de datos ENCOVI para hogares (df_hogar) e individuos (df_personas), primero aseguramos que nuestra variable objetivo 'pobreza' fuera adecuada para una tarea de clasificación. Nuestra investigación reveló que 'pobreza' contenía valores únicos [3, 2, 1], lo que sugiere una clasificación ordinal o categórica. Para simplificar el análisis, binarizamos la variable 'pobreza' utilizando un umbral de pobreza arbitrario de 2. Esto convirtió la variable en un formato binario donde los valores mayores a 2 indican 'pobreza' (codificado como 1) y los valores 2 o menos indican 'no pobreza' (codificado como 0). Esta binarización nos permitió preparar los datos para un modelo de clasificación binaria.

```
# Define a threshold for poverty, this is an arbitrary example
poverty_threshold = 2

# Binarize 'pobreza'
df_hogar['pobreza'] = (df_hogar['pobreza'] > poverty_threshold).astype(int)
df_personas['pobreza'] = (df_personas['pobreza'] > poverty_threshold).astype(int)

# Check the binarization
print(df_hogar['pobreza'].value_counts())
print(df_personas['pobreza'].value_counts())
```

✓ 0.0s

```
0    11536
Name: pobreza, dtype: int64
0    54822
Name: pobreza, dtype: int64
```

Tras la binarización, confirmamos la distribución de las clases en cada conjunto de datos: en df_hogar encontramos 5,897 instancias de 'no pobreza' y 5,639 de 'pobreza'; en df_personas contamos 33,930 instancias de 'no pobreza' y 20,892 de 'pobreza'. Esta distribución

relativamente equilibrada nos ayuda a minimizar el sesgo de clase desbalanceada en el entrenamiento de modelos predictivos.

Finalmente, dividimos los datos en conjuntos de entrenamiento y prueba con una proporción de 80-20, reservando el 20% de los datos para la evaluación del modelo. Este paso es crucial para validar la capacidad de generalización del modelo y asegurar que su rendimiento es robusto a nuevos datos no vistos durante la fase de entrenamiento. La aleatoriedad en la división se controló estableciendo una semilla (random_state) para asegurar la reproducibilidad de los resultados.

```
# Splitting the data for df_hogar
X_hogar = df_hogar.drop('pobreza', axis=1)
y_hogar = df_hogar['pobreza']
X_train_hogar, X_test_hogar, y_train_hogar, y_test_hogar = train_test_split(X_hogar, y_hogar, test_size=0.2, random_state=42)

# Splitting the data for df_personas
X_personas = df_personas.drop('pobreza', axis=1)
y_personas = df_personas['pobreza']
X_train_personas, X_test_personas, y_train_personas, y_test_personas = train_test_split(X_personas, y_personas, test_size=0.2, random_state=42)
```
✓ 0.0s

Construcción del modelo

Para la construcción de los modelos predictivos, se seleccionó el algoritmo RandomForestClassifier. Se eligió este modelo de ensamble debido a que opera construyendo una multitud de árboles de decisión durante el entrenamiento y generando la clase que es el modo de las clasificaciones de los árboles individuales para la clasificación. Esta técnica es particularmente efectiva para conjuntos de datos complejos y heterogéneos, como es el caso de los datos socioeconómicos que estamos analizando.

Se inicializaron dos clasificadores de bosque aleatorio independientes: rf_classifier_hogar para el conjunto de datos de los hogares y rf_classifier_personas para el conjunto de datos de las personas. Ambos clasificadores se configuraron con 100 árboles (n_estimators=100), una cantidad estándar que ofrece un buen equilibrio entre rendimiento y tiempo de computación, y se estableció una semilla aleatoria (random_state=42) para garantizar la reproducibilidad de los resultados.

Una vez inicializados, entrenamos cada clasificador con sus respectivos conjuntos de entrenamiento. El modelo rf_classifier_hogar se ajustó utilizando las características y etiquetas del conjunto de datos de hogares, mientras que el modelo rf_classifier_personas se ajustó de manera similar con los datos de individuos. Este proceso de entrenamiento permite que los modelos aprendan las complejas relaciones entre las características y la variable objetivo 'pobreza', con el fin de hacer predicciones precisas sobre nuevos datos.

```python
# Initialize the RandomForestClassifier
rf_classifier_hogar = RandomForestClassifier(n_estimators=100, random_state=42)
rf_classifier_personas = RandomForestClassifier(n_estimators=100, random_state=42)

# Fit the model to the training data
rf_classifier_hogar.fit(X_train_hogar, y_train_hogar)
rf_classifier_personas.fit(X_train_personas, y_train_personas)
```

✓ 3.8s

```
RandomForestClassifier
RandomForestClassifier(random_state=42)
```

Evaluación del modelo

Una vez entrenados los modelos RandomForestClassifier para los conjuntos de datos de hogares (df_hogar) y de personas (df_personas), procedimos a la etapa de evaluación. La predicción sobre los datos de prueba nos permite estimar cómo los modelos funcionarán en la práctica, al ser expuestos a información que no han visto durante el entrenamiento.

Para el conjunto de datos de hogares, el informe de clasificación muestra una precisión y un recall balanceados entre las clases 0 (no pobreza) y 1 (pobreza), con una precisión ligeramente mayor para la clase 0. La precisión general del modelo (accuracy) fue del 73%, lo cual es un indicador de un rendimiento aceptable, pero con margen de mejora. Es particularmente importante destacar que en problemas sociales complejos como la predicción de niveles de pobreza, incluso pequeñas mejoras en la precisión y el recall pueden tener impactos significativos en la aplicación práctica de estos modelos. Para el conjunto de datos de personas, los resultados fueron excepcionalmente buenos, con una precisión y un recall de 99% para ambas clases.

```
Classification Report for df_hogar:
              precision    recall  f1-score   support

           0       0.76      0.74      0.75      1239
           1       0.71      0.72      0.71      1069

    accuracy                           0.73      2308
   macro avg       0.73      0.73      0.73      2308
weighted avg       0.73      0.73      0.73      2308

Classification Report for df_personas:
              precision    recall  f1-score   support

           0       0.99      0.99      0.99      6829
           1       0.99      0.99      0.99      4136

    accuracy                           0.99     10965
   macro avg       0.99      0.99      0.99     10965
weighted avg       0.99      0.99      0.99     10965
```

Optimización y ajuste del modelo

En nuestro caso, hemos llevado a cabo un proceso de búsqueda en cuadrícula (Grid Search) con validación cruzada para el clasificador RandomForestClassifier en ambos conjuntos de datos, df_hogar y df_personas.

El grid search evalúa y compara el rendimiento del clasificador bajo diferentes combinaciones de hiperparámetros definidos en param_grid. Los hiperparámetros que hemos considerado incluyen:

- n_estimators: El número de árboles en el bosque.
- max_depth: La profundidad máxima de los árboles.
- min_samples_split: El número mínimo de muestras requeridas para dividir un nodo interno.

```python
# Define the parameter grid
param_grid = {
    'n_estimators': [100, 200],
    'max_depth': [10, 20, None],
    'min_samples_split': [2, 5, 10]
}

# Initialize the RandomForestClassifier
rf_classifier = RandomForestClassifier(random_state=42)

# Initialize the GridSearchCV object
grid_search = GridSearchCV(estimator=rf_classifier, param_grid=param_grid,
                           cv=5, n_jobs=-1, scoring='accuracy', verbose=2)

# Fit the GridSearchCV object to the data
grid_search.fit(X_train_hogar, y_train_hogar)

# Predict on the test data using the best model
y_pred_hogar = grid_search.best_estimator_.predict(X_test_hogar)

# Generate classification report
report_hogar = classification_report(y_test_hogar, y_pred_hogar)


# Repeat the process for df_personas
grid_search.fit(X_train_personas, y_train_personas)
y_pred_personas = grid_search.best_estimator_.predict(X_test_personas)
report_personas = classification_report(y_test_personas, y_pred_personas)
```

Para el conjunto de datos df_hogar, el ajuste de hiperparámetros resultó en una mejora en todas las métricas de la precisión, el recall y el F1-score para ambas clases, así como en la precisión general del modelo. La precisión y el recall aumentaron ligeramente, lo que indica que el modelo ajustado es más equilibrado y eficaz para identificar y clasificar las instancias de pobreza correctamente.

En el caso de df_personas, el ajuste de hiperparámetros confirmó que el modelo ya estaba realizando predicciones con alta precisión, lo que se mantuvo después del ajuste. Con una precisión y un recall de 0.99, el modelo ajustado continúa siendo extremadamente preciso en la clasificación de las instancias.

```
Classification Report for df_hogar after hyperparameter tuning:
              precision    recall  f1-score   support

           0       0.78      0.74      0.76      1239
           1       0.72      0.76      0.74      1069

    accuracy                           0.75      2308
   macro avg       0.75      0.75      0.75      2308
weighted avg       0.75      0.75      0.75      2308

Classification Report for df_personas after hyperparameter tuning:
              precision    recall  f1-score   support

           0       0.99      0.99      0.99      6829
           1       0.99      0.99      0.99      4136

    accuracy                           0.99     10965
   macro avg       0.99      0.99      0.99     10965
weighted avg       0.99      0.99      0.99     10965
```

Reflexión

Uno de los desafíos principales fue garantizar la correcta preparación y procesamiento de los datos. La variable objetivo 'pobreza' requería una cuidadosa consideración para determinar si se trataba de una característica categórica o continua. La decisión de binarizar esta variable se tomó con base en un umbral definido, lo cual es una práctica común en problemas donde se necesita categorizar datos continuos para tareas de clasificación. Este paso fue crucial para la adecuada implementación y evaluación de los modelos de clasificación.

Las lecciones aprendidas son numerosas. Se ha reforzado la comprensión de que una meticulosa exploración de datos es fundamental antes de cualquier análisis. La importancia de una selección adecuada de modelos y la afinación de hiperparámetros ha quedado clara, así como la necesidad de interpretar los resultados con un enfoque crítico y considerando el contexto del problema.

En términos profesionales, el potencial de aplicar estas habilidades es considerable. La capacidad de transformar y preparar datos de forma efectiva es esencial para cualquier Científico de Datos. La habilidad de seleccionar y optimizar modelos tiene aplicaciones directas en la creación de sistemas predictivos en diversas industrias. Por ejemplo, en el sector financiero, esto podría traducirse en la predicción de riesgo crediticio, mientras que en la salud pública podría aplicarse a la predicción de brotes epidemiológicos.

Por último, este laboratorio ha subrayado el valor de la comunicación efectiva de resultados analíticos. Ser capaz de traducir análisis complejos en decisiones informadas es una competencia clave que impulsa el impacto de un Científico de Datos en cualquier organización. Estas habilidades, junto con un enfoque ético y considerado, no solo mejoran la toma de decisiones basada en datos, sino que también contribuyen a la construcción de tecnologías responsables que benefician a la sociedad.