TRABAJO FINAL
Coder House

1 PREGUNTA INVESTIGACIÓN

COMO SE PODRIA **MEJORAR LAS PREDICCIONES DE LOS TIEMPOS Y CUMPLIMIENTO** DE LAS PROMESAS DE ENTREGA DEL REPARTO DE PEDIDOS.

2 PROBLEMA MODELO ACTUAL

PREDICCION ENTRE TIEMPO ACTUAL Y TIEMPO PROMETIDO DE ENTREGA CON UN **RANGO "AMPLIO"** Y CON POTENCIAL DE MEJORA EN PRECISIÓN ENTRE DICHOS TIEMPOS.

3 PROPUESTA DE SOLUCIÓN

MEJORA EN PRECISIÓN DEL MODELO PREDICTIVO E IMPLEMENTACIÓN DE ESTRATEGIA QUE AYUDE EN EL CUMPLIMIENTO DE LA PROMESA REALIZADA AL CLIENTE.

SE UTILIZA TOTAL MINUTES COMO VARIABLE OBJETIVO.

Estrategias analíticas de datos y preparación de DF para modelamiento (Tipos de datos, limpieza de BBDD, correlación, outliers)

Selección características relevantes para el modelo

Selección características relevantes para el modelo

Revisar métricas nuevo modelo

MEJORAN?

NO

Volver a analizar importancia de variables para seguir descartando irrelevantes para mejorar el modelo

SI

¿Se Puede seguir mejorando?

SI

Volver a analizar importancia de variables para seguir descartando irrelevantes para mejorar el modelo

NO

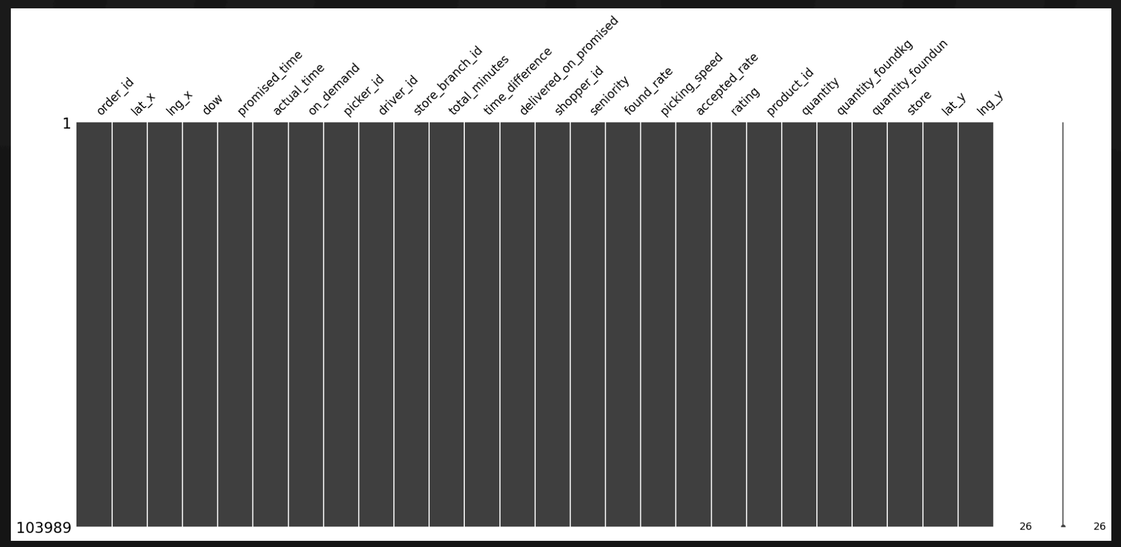
Modelo práctico y reducido con buenas métricas de desempeño

Tratamiento de datos (conversiones, nan, valores únicos,etc…

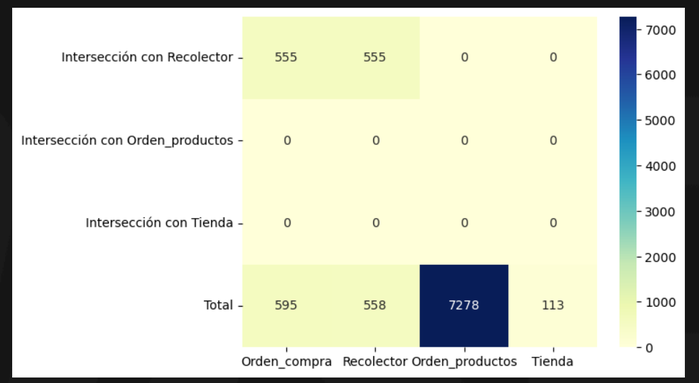
```python
def time_string_to_minutes(time_string):
    time_parts = [int(part) for part in time_string.split(':')]
    return time_parts[0] * 60 + time_parts[1] + time_parts[2] / 60


Orden_compra['promised_time'] = Orden_compra['promised_time'].apply(time_string_to_minutes)
Orden_compra['actual_time'] = Orden_compra['actual_time'].apply(time_string_to_minutes)


Orden_compra['on_demand'] = Orden_compra['on_demand'].astype(int)


Orden_compra['time_difference'] = Orden_compra['promised_time'] - Orden_compra['actual_time']

Orden_compra['delivered_on_promised'] = ((Orden_compra['on_demand'] & (Orden_compra['time_difference'] <= 90)) |
                                         (~Orden_compra['on_demand'] & (Orden_compra['time_difference'] >= 0))).astype(int)

Orden_compra = Orden_compra[Orden_compra['time_difference'] <= 90]
Orden_productos['buy_unit'] = Orden_productos['buy_unit'].replace({'KG': 0, 'UN': 1})
Orden_productos['quantity_foundkg'] = Orden_productos['quantity_found'].where(Orden_productos['buy_unit'] == 0, 0)
Orden_productos['quantity_foundun'] = Orden_productos['quantity_found'].where(Orden_productos['buy_unit'] == 1, 0)
Orden_productos.drop('quantity_found', axis=1, inplace=True)
Orden_productos.drop('buy_unit', axis=1, inplace=True)
```

```python
Recolector['seniority'] = Recolector['seniority'].replace({
    'BEGINNER': 0,
    'INTERMEDIATE': 1,
    'ADVANCED': 2,
    'REVIEW': 3
})
Orden_compra = (Orden_compra.dropna())
Orden_productos = (Orden_productos.dropna())
Recolector = (Recolector.dropna())
Tienda = Tienda.dropna()
```

```python
Orden_productos['quantity'] = pd.to_numeric(Orden_productos['quantity'], errors='coerce')
sorted_unique_values = sorted(Orden_productos['quantity'].unique().tolist())
valores_unicos = Recolector['seniority'].unique()
```

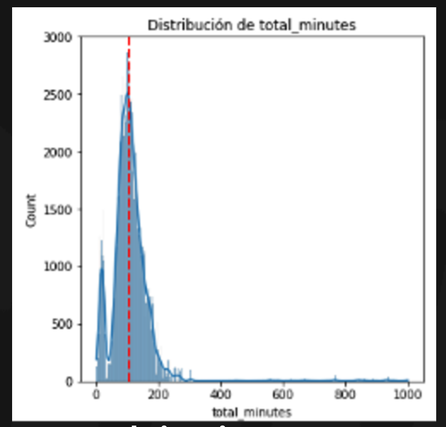


Murge



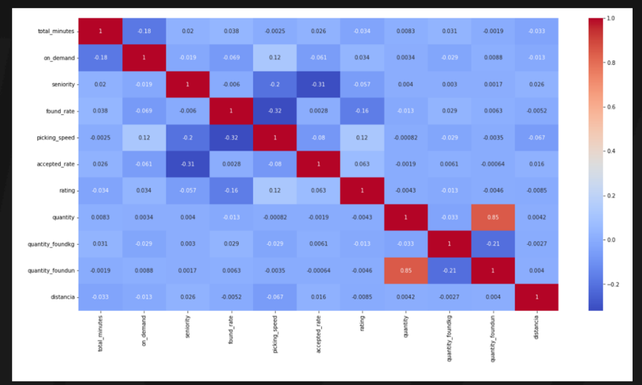
- El objetivo detrás esconseguir un df que cruce las id’s de las 4 tablas en común de cara al modelamiento de los datos

Distribuciones - Variable Objetivo



- Variable objetivo con sesgo negativo (no normal).
- Se presentaron variables sin distribuciones normales (variables numéricas).
- En datos categóricos se presentan frecuencias altas en ciertas variables (id_tienda, id_recolector, etc)

Correlaciones



- Variable objetivo (total_minutes) no presenta correlación alta con ninguna de las variables a utilizar.
- Correlaciones altas entre quantity y quantity_found (no relevantes en analisis)

Pruebas de Normalidad

```
lat_x:
  Estadística de Shapiro-Wilk: 0.2048131227493286
  p-value: 0.0

Los datos no provienen de una distribución normal
lng_x:
  Estadística de Shapiro-Wilk: 0.4140195846557617
  p-value: 0.0

Los datos no provienen de una distribución normal
dow:
  Estadística de Shapiro-Wilk: 0.9377398490905762
  p-value: 0.0

Los datos no provienen de una distribución normal
promised_time:
  Estadística de Shapiro-Wilk: 0.9477365016937256
  p-value: 0.0

Los datos no provienen de una distribución normal
actual_time:
  Estadística de Shapiro-Wilk: 0.9491007924079895
  p-value: 0.0

Los datos no provienen de una distribución normal
on_demand:
  Estadística de Shapiro-Wilk: 0.5806776285171509
  p-value: 0.0

Los datos no provienen de una distribución normal
total_minutes:
  Estadística de Shapiro-Wilk: 0.6914598941802979
  p-value: 0.0

Los datos no provienen de una distribución normal
seniority:
  Estadística de Shapiro-Wilk: 0.7356338500976562
  p-value: 0.0

Los datos no provienen de una distribución normal
found_rate:
  Estadística de Shapiro-Wilk: 0.9517794251441956
  p-value: 0.0

Los datos no provienen de una distribución normal
picking_speed:
  Estadística de Shapiro-Wilk: 0.886906087398529
  p-value: 0.0

Los datos no provienen de una distribución normal
accepted_rate:
  Estadística de Shapiro-Wilk: 0.6065037846565247
  p-value: 0.0

Los datos no provienen de una distribución normal
rating:
  Estadística de Shapiro-Wilk: 0.9365376830101013
  p-value: 0.0

Los datos no provienen de una distribución normal
quantity:
  Estadística de Shapiro-Wilk: 0.4435035586357116 7
  p-value: 0.0

Los datos no provienen de una distribución normal
quantity_found:
  Estadística de Shapiro-Wilk: 0.4995470643043518
  p-value: 0.0

Los datos no provienen de una distribución normal
buy_unit:
  Estadística de Shapiro-Wilk: 0.507822573184967
  p-value: 0.0

Los datos no provienen de una distribución normal
lat_y:
  Estadística de Shapiro-Wilk: 0.1412869771957397
  p-value: 0.0
```

- Se realizó test de Shapiro-wilk para determinar normalidad de variables.
- Ninguna presentó normalidad

1

DOW(Dia de la semana) : la media está en torno a 2.6 , lo que sugiere que la mayor parte de los datos se recopilaron a principios de semana(si 0 es domingo, 2.6 caería entre martes y miércoles)

2

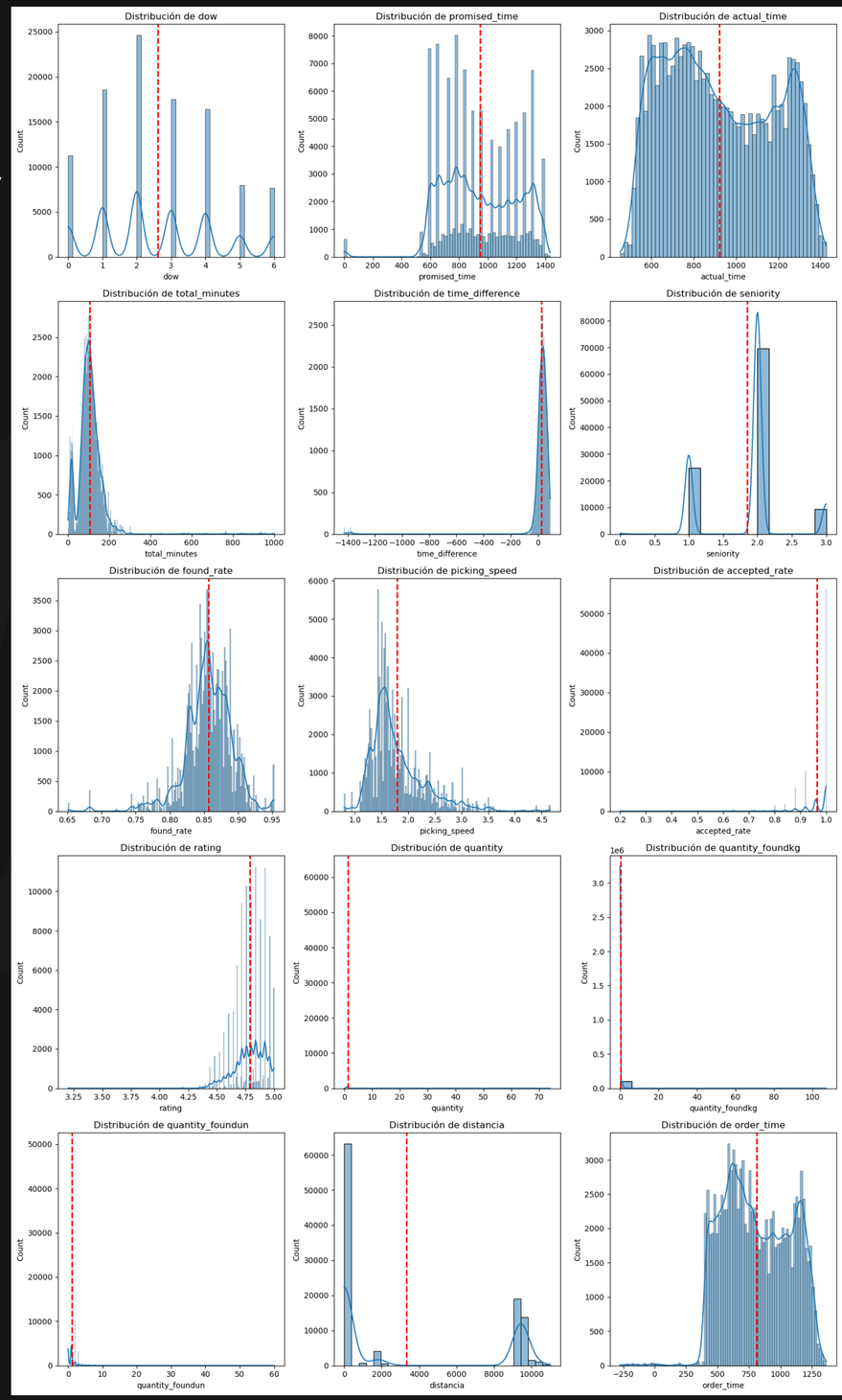
Promised_time y actual_time: la media y la mediana son bastante cercanas , lo que sugiere una distribución simétrica

3

On_demand: variable binaria. La media esta cercana a 0.3, lo que indica que alrededor del 30% de los datos están marcados como “on_demand”

4

Total_minutes, time_difference, order_time: algunas variaciones importantes, como se puede ver en la desviación estándar. Algunas diferencias negativas, sugiere que algunos pedidos se entregaron después de lo prometido

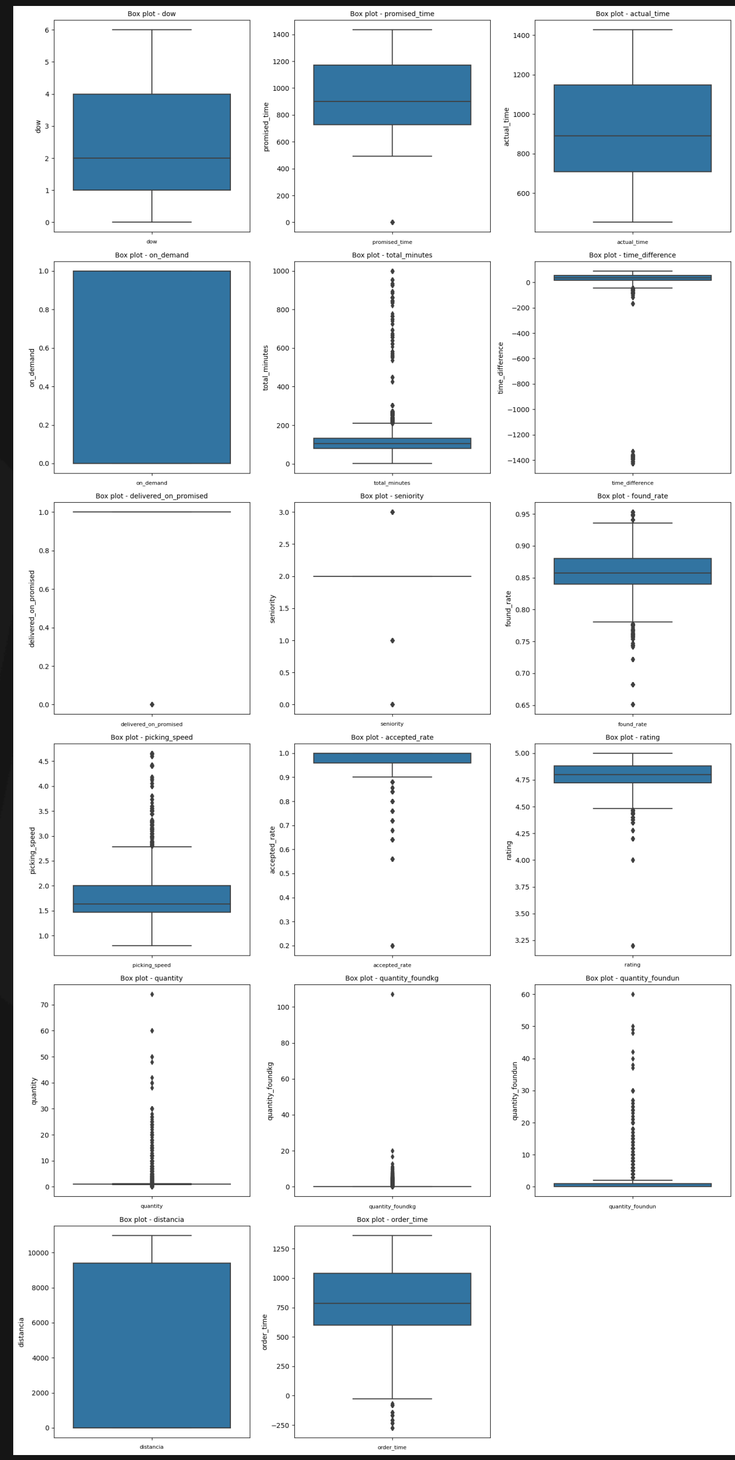


OUTLIERS

1 Las columnas 'seniority', 'quantity','quantity_found', y 'buy_unit' tienen una proporción alta de outliers, superando el 20%. En particular, 'quantity_found' tiene la mayor proporción de outliers, casi el 46.3% de los datos.

2 **'accepted_rate', 'picking_speed', 'found_rate', 'delivered_on_promised', 'total_minutes', ‘lat_x', tiene una cantidad de outliers que se podria considerar sin embargo de menor escala que las primeras.**

3 Algunas columnas como 'dow', 'on_demand', 'lng_y', y 'distancia' no presentan outliers



Ninguna de las columnas numéricas parece seguir una distribución normal, ya que todos los p-valores de la prueba de Shapiro-Wilk son 0

Test Normalidad

```
dow:
  Estadística de Shapiro-Wilk: 0.9378895163536072
  p-value: 0.0

Los datos no provienen de una distribución normal
promised_time:
  Estadística de Shapiro-Wilk: 0.947485089302063
  p-value: 0.0

Los datos no provienen de una distribución normal
actual_time:
  Estadística de Shapiro-Wilk: 0.9497385025024414
  p-value: 0.0

Los datos no provienen de una distribución normal
on_demand:
  Estadística de Shapiro-Wilk: 0.5741005539894104
  p-value: 0.0

Los datos no provienen de una distribución normal
total_minutes:
  Estadística de Shapiro-Wilk: 0.6994078159332275
  p-value: 0.0

Los datos no provienen de una distribución normal
seniority:
  Estadística de Shapiro-Wilk: 0.7279052734375
  p-value: 0.0

Los datos no provienen de una distribución normal
found_rate:
  Estadística de Shapiro-Wilk: 0.9515535831451416
  p-value: 0.0

Los datos no provienen de una distribución normal
picking_speed:
  Estadística de Shapiro-Wilk: 0.8863444924354553
  p-value: 0.0

Los datos no provienen de una distribución normal
```

```
accepted_rate:
  Estadística de Shapiro-Wilk: 0.6083357334136963
  p-value: 0.0

Los datos no provienen de una distribución normal
rating:
  Estadística de Shapiro-Wilk: 0.9391042590141296
  p-value: 0.0

Los datos no provienen de una distribución normal
quantity:
  Estadística de Shapiro-Wilk: 0.44558823108673096
  p-value: 0.0

Los datos no provienen de una distribución normal
distancia:
  Estadística de Shapiro-Wilk: 0.633170485496521
  p-value: 0.0
```

Modelos

se evalúa distintos modelos para poder predecir la variable “total_time” la cual es la variable objetivo.

Random Forest

Se utilizan cuando se busca reducir el sobreajuste y mejorar la generalización. Cada árbol de decisión se entrena con diferentes muestras de datos y características seleccionadas aleatoriamente. Pueden manejar relaciones no lineales y capturar interacciones complejas entre las variables de entrada. También son resistentes al ruido y a valores atípicos en los datos.

Decision Tree

Los árboles de decisión son modelos fáciles de interpretar y visualizar. El flujo de decisiones se representa de forma intuitiva mediante un árbol, lo que permite entender cómo se toman las decisiones y qué características son más importantes para hacerlo. pueden manejar grandes conjuntos de datos de manera eficiente y conjuntos de datos que contienen tanto variables categóricas como numéricas. Pueden realizar divisiones en función de diferentes tipos de atributos y, por lo tanto, son útiles para problemas con una mezcla de características.

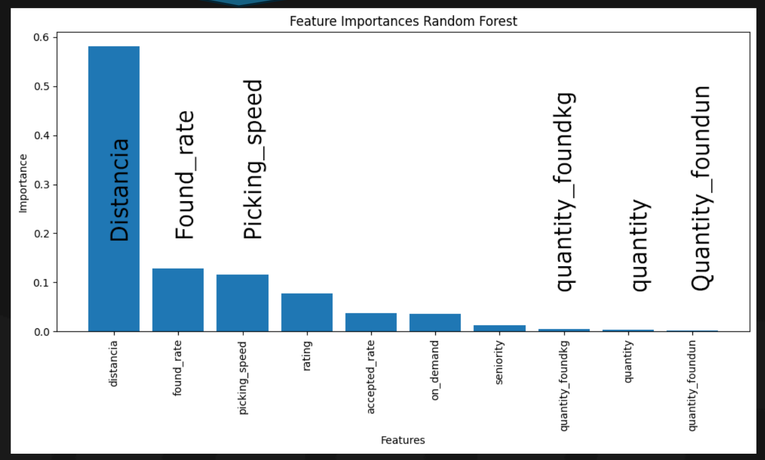
GAM

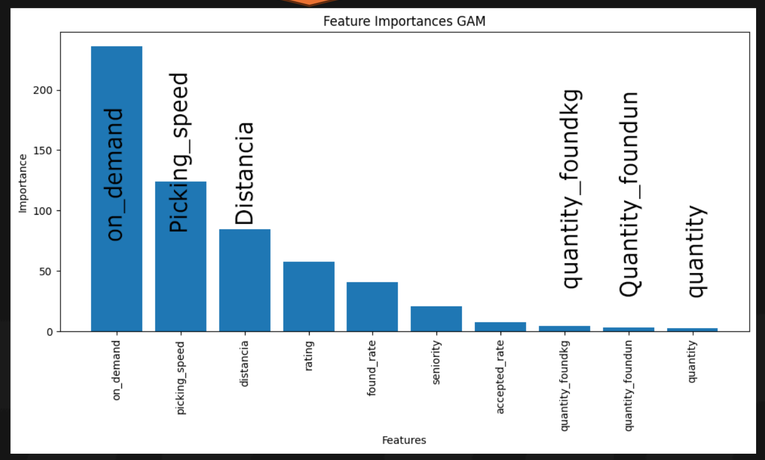
Son modelos flexibles que pueden modelar relaciones no lineales y capturar posibles interacciones entre variables predictoras. Pueden manejar diferentes tipos de variables y representar tendencias no lineales en los datos.

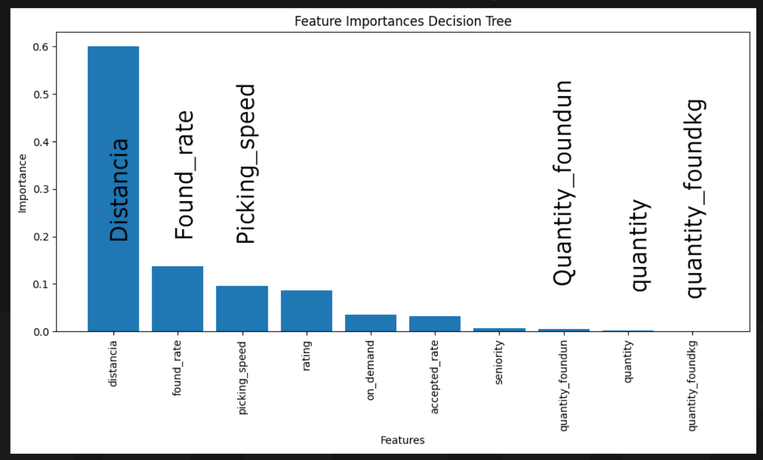
REDUCCION DE VARIABLES

Random Forest | **Decision Tree** | **GAM**

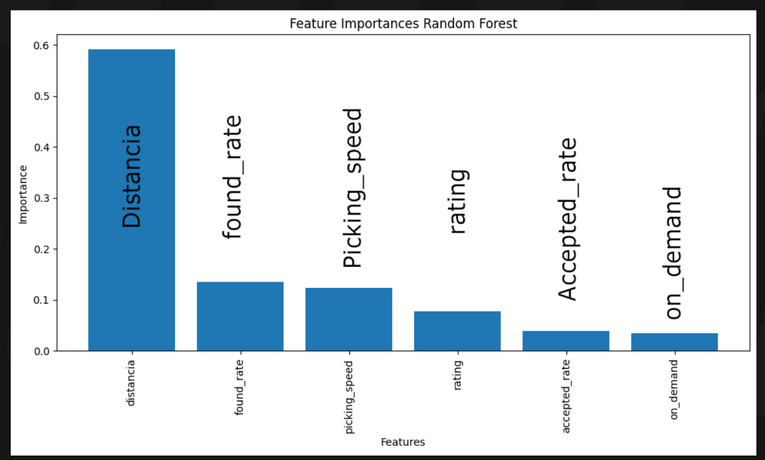
iteracion 1

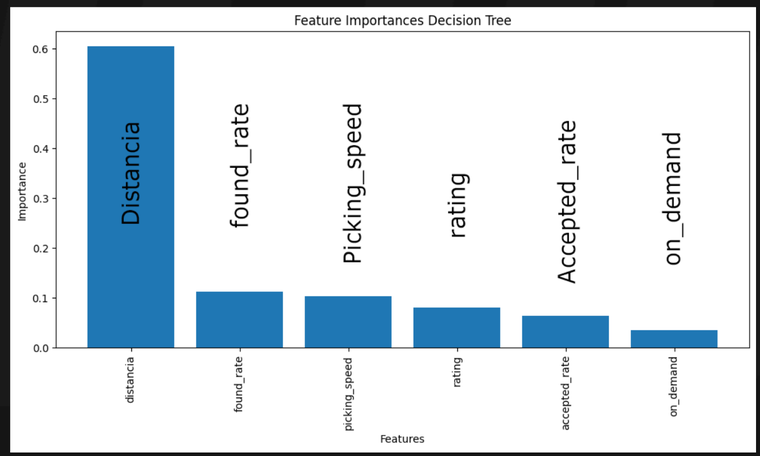


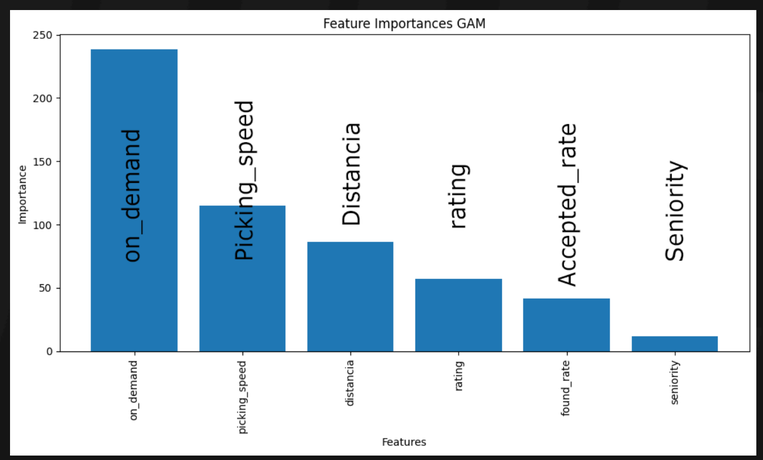




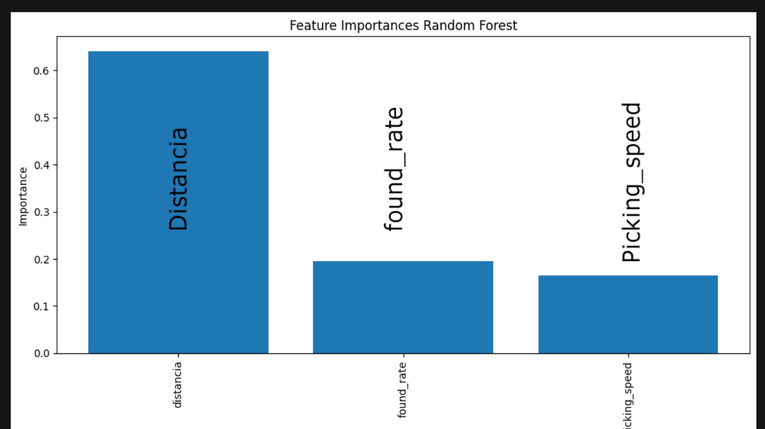
iteracion 2

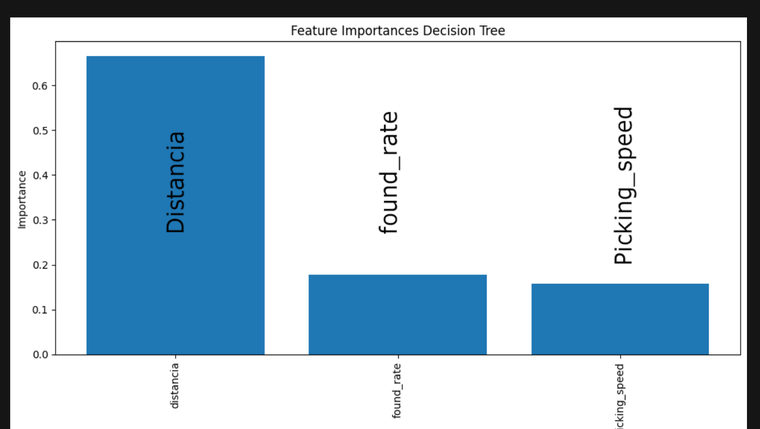


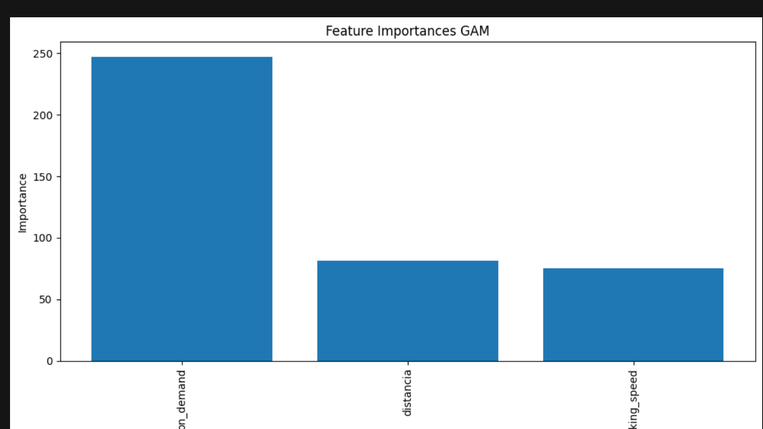




iteracion 3







ITERACIONES MODELOS

Random Forest

	MSE	MAE	R^2
iteracion1	42.012638	1.705303e-13	0.989834
iteracion2	25.020041	1.278977e-13	0.993946
iteracion3	27.565841	1.278977e-13	0.993329

Decision Tree

	MSE	MAE	R^2
iteracion1	93.199697	7.105427e-15	0.977447
iteracion2	54.981483	1.421085e-14	0.986695
iteracion3	19.462654	1.421085e-14	0.995290

GAM

	MSE	MAE	R^2
iteracion1	3883.326012	24.972518	0.060291
iteracion2	3883.206402	25.098819	0.060320
iteracion3	3930.696353	24.923066	0.048828

OPTIMIZACIÓN HIPERPARAMETROS

Random Forest

Mejores hiperparámetros: {
'max_depth': None,
'min_samples_leaf': 1,
'min_samples_split': 2,
'n_estimators': 200
}

MAE: 2.5579538487363607e-13
MSE: 26.286737828688917
R^2: 0.9936389862493383

Decision Tree

Mejores hiperparámetros = {
‘max_depth': None,
'min_samples_leaf': 1,
'min_samples_split’: 5
}

MAE: 1.4210854715202004e-14
MSE: 20.9373142510727
R^2: 0.9949334700744934

GAM

{
'MSE': 4013.2598708459905,
'MAE': 25.943303205426197,
'R^2': 0.02884863881554578
}

El modelo de Arbol de Decisión fue aquel que presentó un mejor rendimiento desde el punto de vista de las métricas obtenidas al momento de correr nuevamente los modelos teniendo el menor error cuadrático medio y el mejor R2 • Seguido ligeramente atras por Random Forest. • Finalmente el modelo GAM muestra el peor resultado de los 3

6) Conclusión

se ordenan los modelos de mejor a peor, para esto se puede utilizar la métrica MSE (Error Cuadrático Medio) como criterio principal, seguido por el MAE (Error Absoluto Medio) y finalmente el R^2 (Coeficiente de Determinación). A continuación, se mostrará un ranking con los modelos ordenados de mejor a peor y una breve descripción de las métricas en cada uno de ellos:

1. Decision Tree
2. Random Forest
3. GAM