



Uso de Taxis Yellow Cab en USA en el año 2020

...

Autores

Domenje, Carlos R.

Fux, Santiago

Profesores

Canavesi, Tobías

Lopez, Yoel

Agosto 2022












¿Qué nos interesa conocer?



**¿Existe una manera de caracterizar
los lugares más recurrentes para
inicio / fin de viaje?**









Dataset: Yellow Cab

Descripción general de los features:

-  • **VendorID:** Proveedor de servicios de tecnologías en taxis (T-PEP)
-  • **tpep_pickup_datetime:** Fecha y hora en el cual el reloj fue activado al iniciar un viaje.
-  • **tpep_dropoff_datetime:** Fecha y hora en el cual el reloj fue desactivado al finalizar un viaje.
-  • **Passenger_count:** El número de pasajeros en el vehículo. (Es un dato que lo ingresa el conductor.
-  • **Trip_distance:** La distancia del viaje transcurrido en millas reportada por el taxímetro.
-  • **PULocationID:** TLC Zona en la que el taxímetro se activó.
-  • **DOLocationID:** TLC Zona en la que el taxímetro se desactivó.
-  • **RateCodeID:** El código de tarifa final vigente al final del viaje.
-  • **Store_and_fwd_flag:** almacenar y reenviar o envío continuo.
-  • **Payment_type:** Un código numérico que significa cómo el pasajero pagó por el viaje.
-  • **Fare_amount:** La tarifa de tiempo y distancia calculada por el taxímetro.

Dataset: Yellow Cab

Descripción general de los features:

-  • **Extra:** Varios extras y recargos. Actualmente, esto solo incluye los cargos de \$0.50 y \$1 por la hora pico y por la noche.
-  • **MTA_tax:** Impuesto MTA de \$0.50 que se activa automáticamente según la tasa de uso del medidor.
-  • **Improvement_surcharge:** Recargo de mejora de \$ 0.30 en viaje en el descenso de bandera.
-  • **Tip_amount:** Importe de la propina: Propinas de tarjetas de crédito. Las propinas en efectivo no están incluidas.
-  • **Tolls_amount:** Importe total de todos los peajes pagados en el viaje.
-  • **Total_amount:** El monto total cobrado a los pasajeros. No incluye propinas en efectivo.
-  • **Congestion_Surcharge:** Importe total recaudado en el viaje por el recargo por congestión del Estado de Nueva York.
-  • **Airport_fee:** \$1.25 para recoger solo en los aeropuertos LaGuardia y John F. Kennedy.

Consideraciones del Dataset

1. Se tomó el 10% de los datos del mes de Enero - Febrero y Marzo de 2020.

	VendorID	passenger_count	trip_distance	RatecodeID	PULocationID	DOLocationID	payment_type	fare_amount	extra
count	1571207.000	1555761.000	1571207.000	1555761.000	1571207.000	1571207.000	1571207.000	1571207.000	1571207.000
mean	1.673	1.501	3.040	1.057	164.392	162.203	1.252	12.615	1.423
std	0.470	1.143	215.275	0.767	65.721	70.006	0.486	11.756	398.892
min	1.000	0.000	-29.100	1.000	1.000	1.000	0.000	-320.000	-4.500
25%	1.000	1.000	0.970	1.000	125.000	113.000	1.000	6.500	0.000
50%	2.000	1.000	1.600	1.000	162.000	162.000	1.000	9.000	0.500
75%	2.000	2.000	2.930	1.000	234.000	234.000	2.000	14.000	2.500
max	6.000	9.000	269803.730	99.000	265.000	265.000	4.000	575.000	500000.800

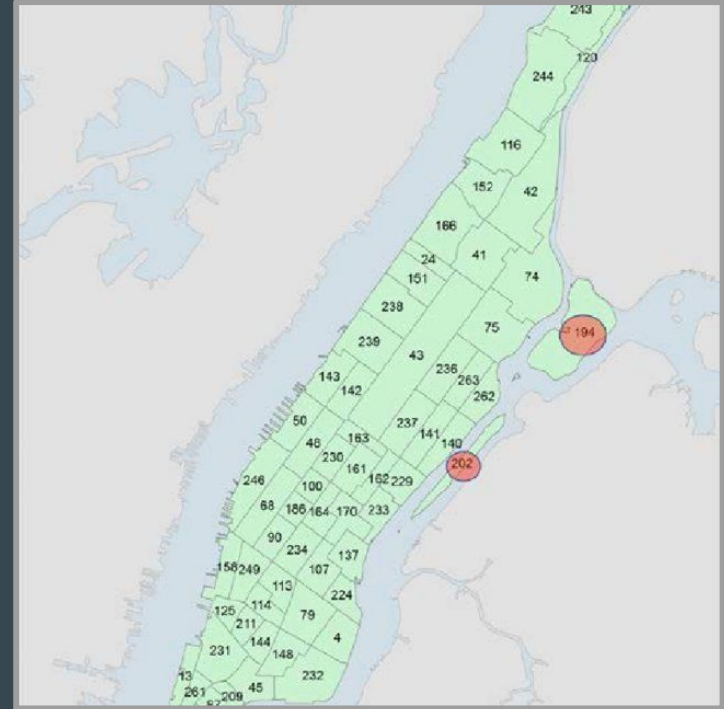


Consideraciones del Dataset

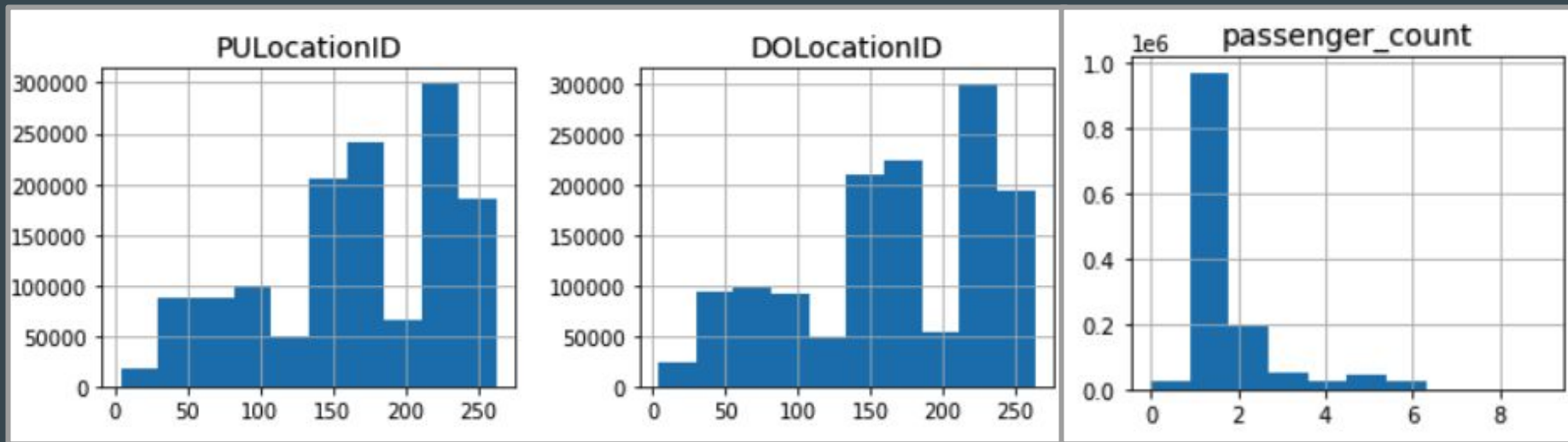
2. Se filtraron los datos con los códigos que pertenecen a Manhattan.



[4, 12, 13, 24, 41, 42, 43, 45, 48, 50, 68, 74, 75, 79, 87, 88, 90, 100, 103, 104, 105, 107, 113, 114, 116, 120, 125, 127, 128, 137, 140, 141, 142, 143, 144, 148, 151, 152, 153, 158, 161, 162, 163, 164, 166, 170, 186, 194, 202, 209, 211, 224, 229, 230, 231, 232, 233, 234, 236, 237, 238, 239, 243, 244, 246, 249, 261, 262, 263]




Distribución de las variables



Datos Inválidos y Nulos


Cantidad de Nulos

VendorID: 0/1571075
tpep_pickup_datetime: 0/1571075
tpep_dropoff_datetime: 0/1571075
passenger_count: 15093/1571075
trip_distance: 0/1571075
RatecodeID: 15093/1571075
store_and_fwd_flag: 15093/1571075
PULocationID: 0/1571075
DOLocationID: 0/1571075
payment_type: 0/1571075
fare_amount: 0/1571075
extra: 0/1571075
mta_tax: 0/1571075
tip_amount: 0/1571075
tolls_amount: 0/1571075
improvement_surcharge: 0/1571075
total_amount: 0/1571075
congestion_surcharge: 15093/1571075
airport_fee: 1571075/1571075



Cantidad de Inválidos (<0)

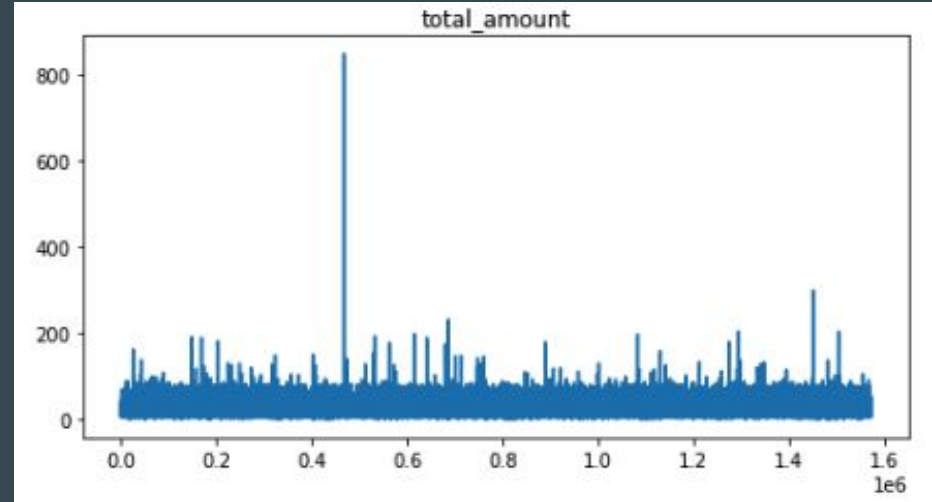
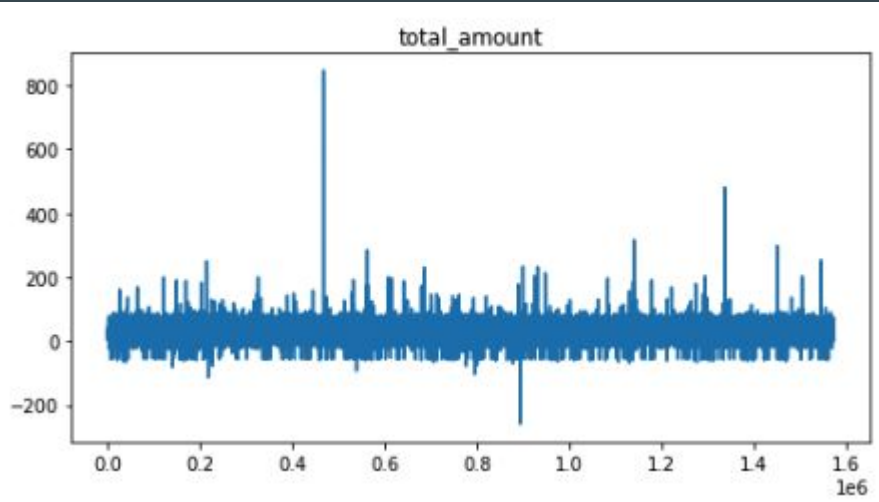
trip_distance: 0/1555982
fare_amount: 4974/1555982
extra: 2390/1555982
mta_tax: 4891/1555982
tip_amount: 55/1555982
tolls_amount: 93/1555982
improvement_surcharge: 4974/1555982
total_amount: 4974/1555982
congestion_surcharge: 3998/1555982



Exploración gráfica de datos inválidos

Datos con valores negativos

Datos con filtro aplicado



Variables tipo DateTime

- Se genera una nueva feature llamada 'duration' qué representa la duración del viaje.
- Se crean tres features más que representan los viajes diurnos, vespertinos y nocturnos (One hot encoding)

duration	morning	afternoon	evening
538.000	0	1	0
621.000	0	1	0
292.000	1	0	0
612.000	1	0	0
379.000	0	1	0



Morning

6-13hs



Afternoon

14-20hs



Evening

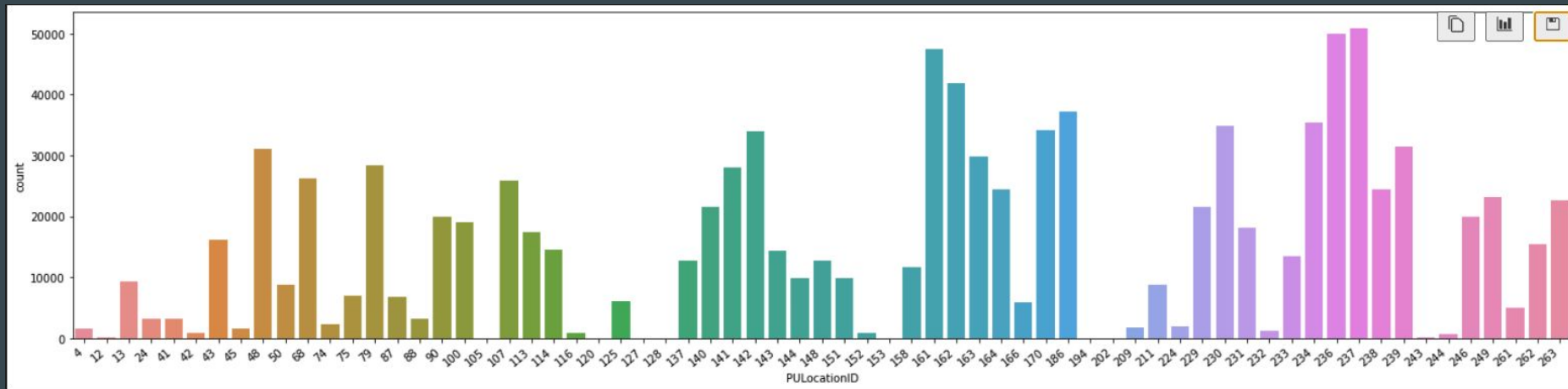
21-5hs



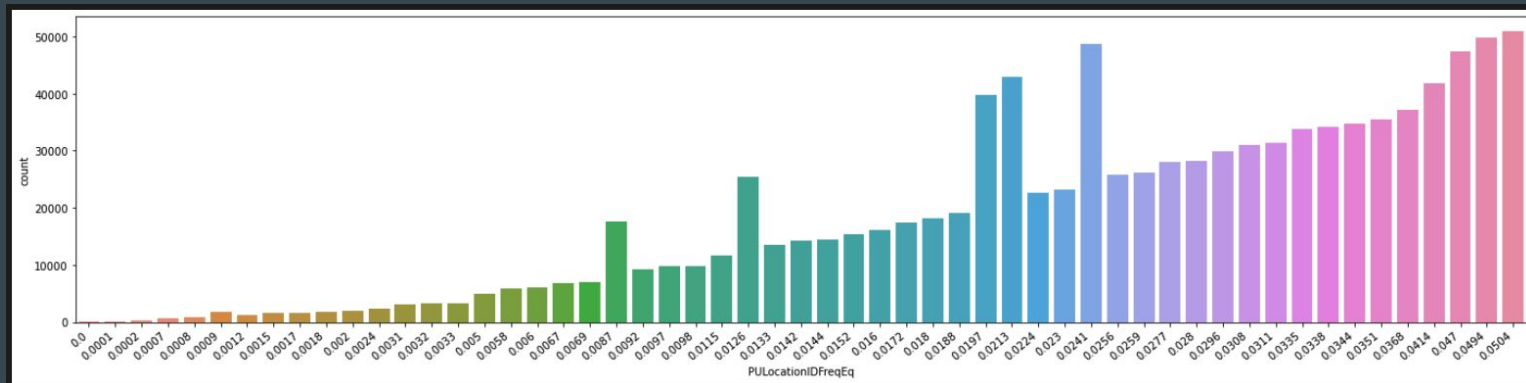
Duración

Fin - Inicio

Variable categórica PULocationID

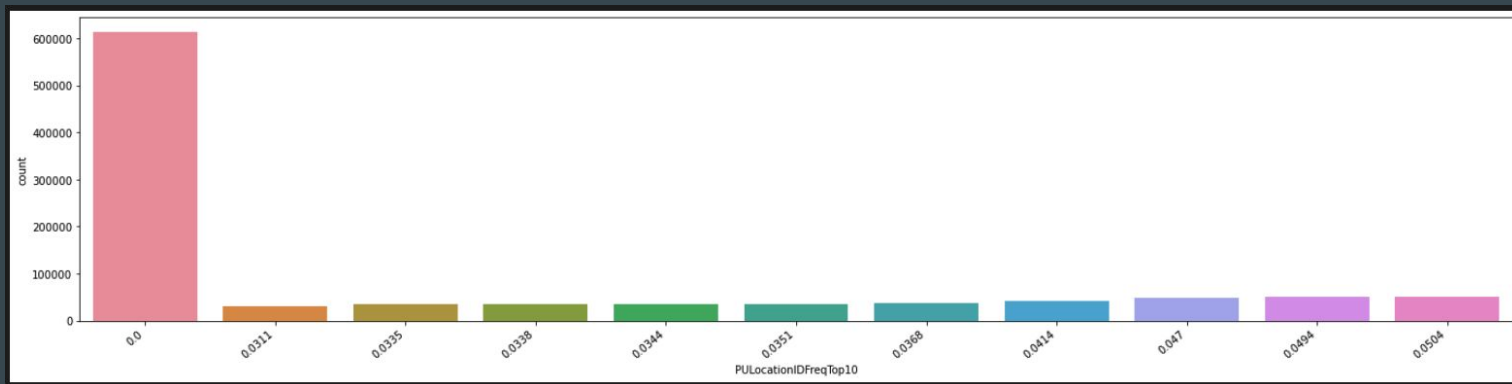


Variable Ecualizada

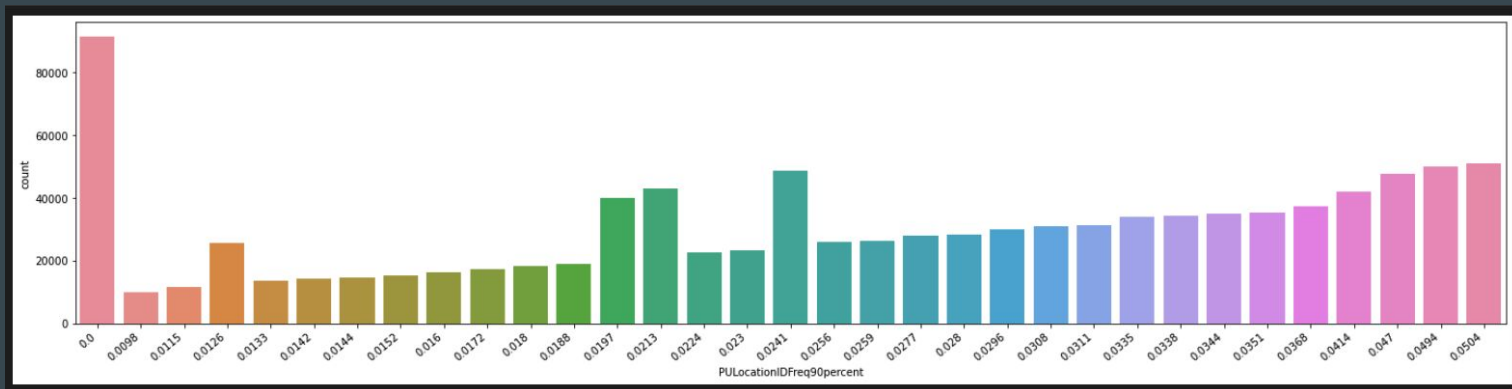


Variable categórica PULocationID

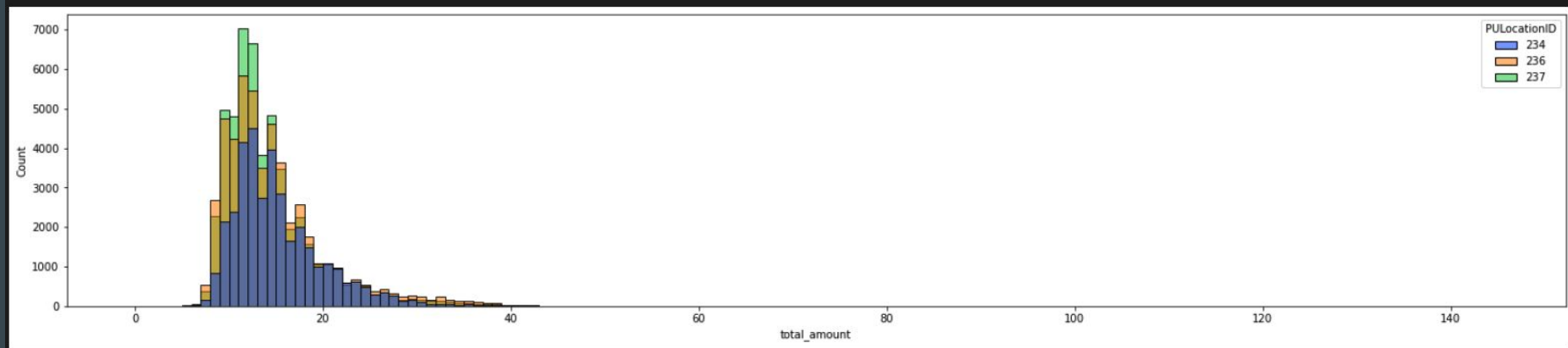
Variable Top 10



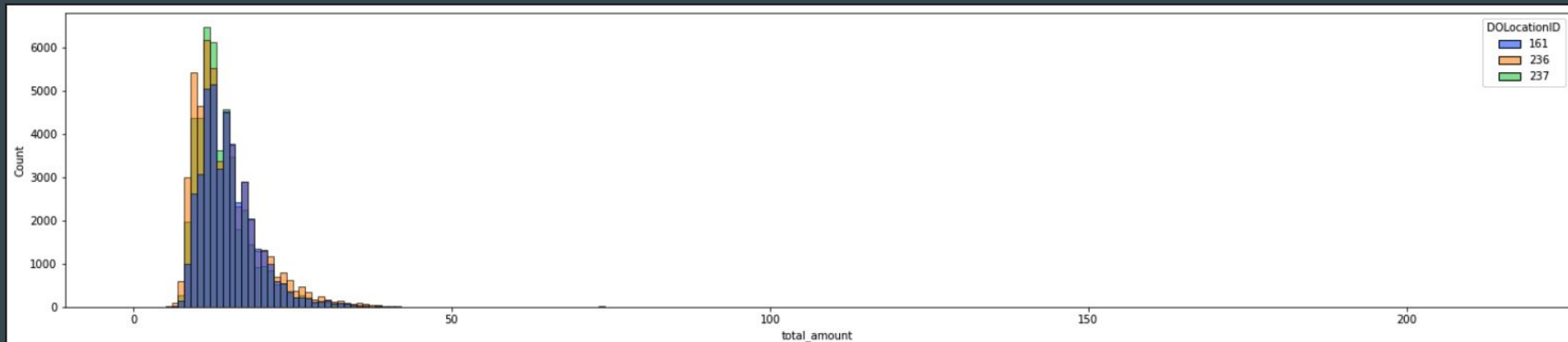
Variable 90%



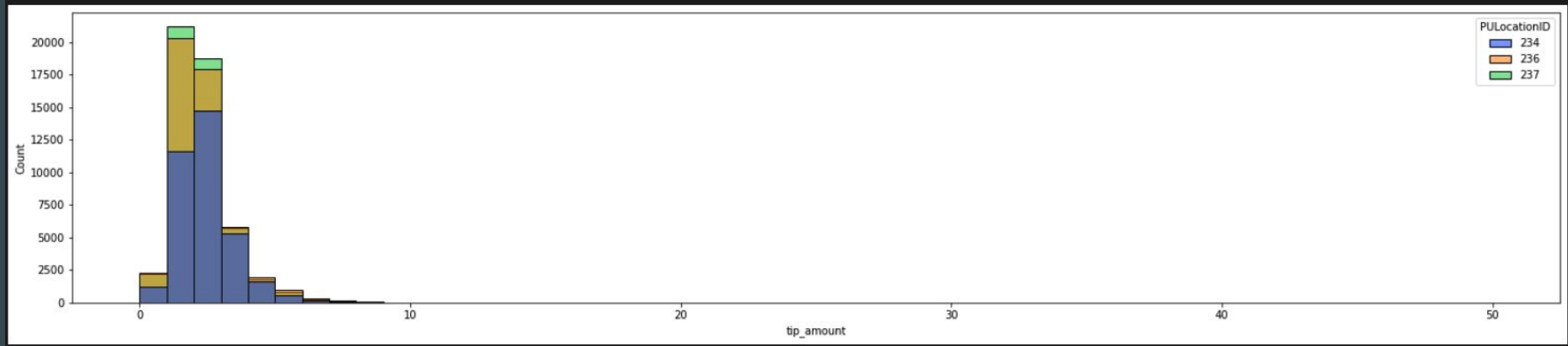
Análisis Gráfico - Inicio Viaje en zona 237 - 236 - 234 vs Total Amount



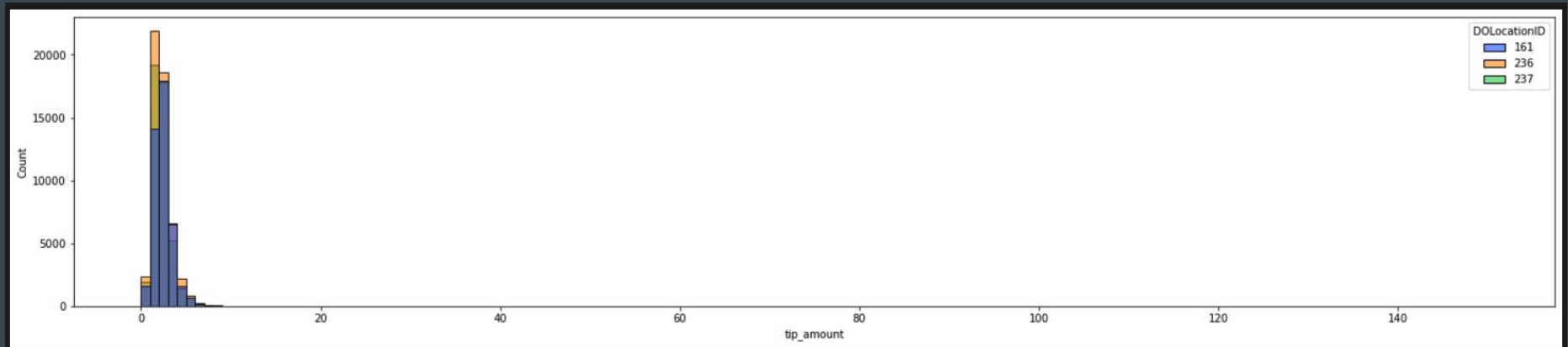
Análisis Gráfico - Fin Viaje en zona 237 - 236 - 161 vs Total Amount



Análisis Gráfico - Inicio Viaje en zona 237 - 236 - 234 vs TIP Amount



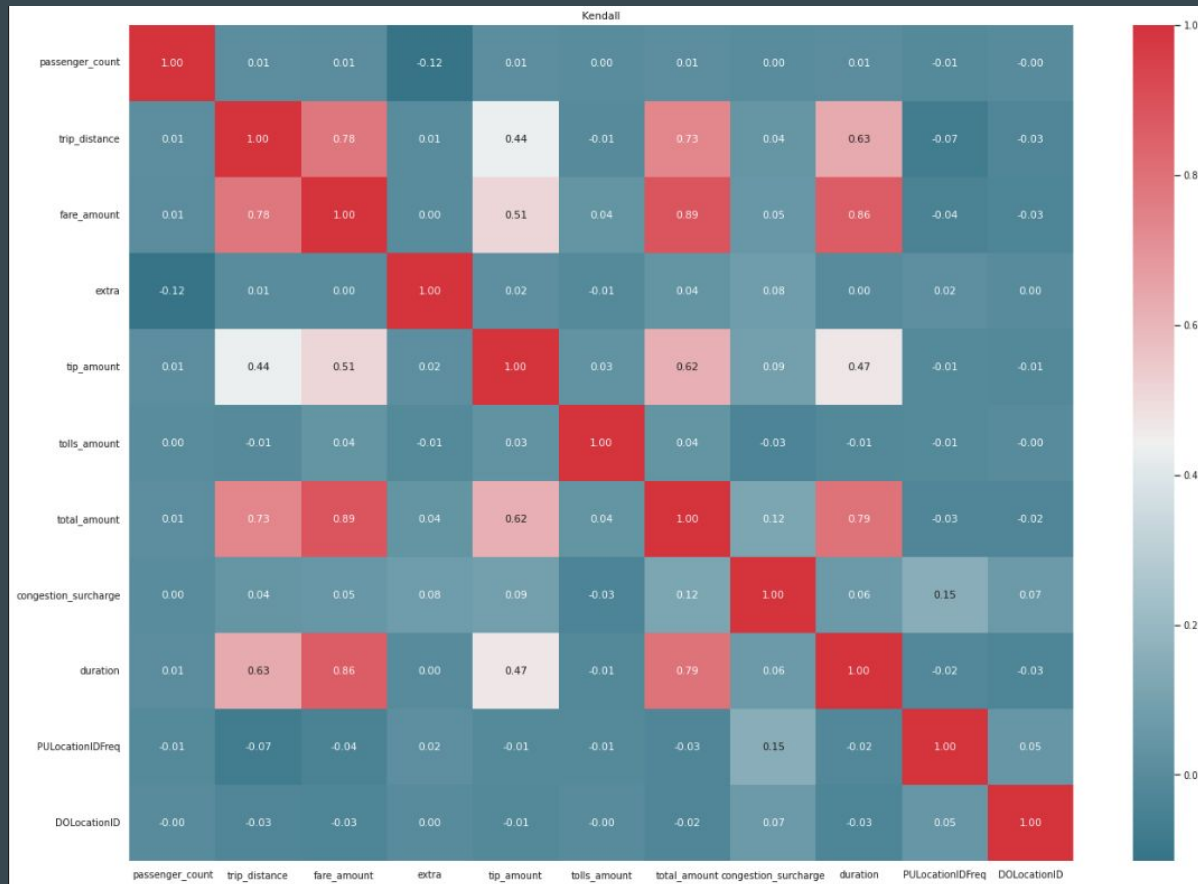
Análisis Gráfico - Fin Viaje en zona 237 - 236 - 161 vs Tip Amount



Selección de Features - Coef. Kendall

Test de correlación de Kendall:

- Test no paramétrico (no asume ninguna distribución.)
- H0: Las variables son independientes.
- H1: Las variables no son independientes.



Datasets Analizados

Dataset EQ



65
categorías

Dataset Top10



10
categorías

Dataset 90%



39
categorías

Modelos Aplicados

Random Forest



Clasificador

Logistic Regression



Clasificador

Modelos Aplicados - Random Forest

```
#multi modelos en random forest
from sklearn.ensemble import RandomForestClassifier
total_trees = 100 # number of trees
max_depth = 5

def run_random_forest(x_tr, x_te, y_tr, y_te):

    rf_aux = RandomForestClassifier(n_estimators = total_trees, criterion = 'entropy', max_depth = max_depth, random_state=0)
    rf_aux.fit(x_tr, y_tr.values.reshape(-1))
    # Utilizamos el método de predicción en los datos de prueba
    y_rf_pred_aux = rf_aux.predict(x_te)
    print(classification_report(y_te,y_rf_pred_aux))
    #grafico los features segun su importancia
    plt.barh(x_tr.columns, rf_aux.feature_importances_)

print(f'Dataset EQ:')
run_random_forest(X_train_eq, X_test_eq, y_train_eq, y_test_eq)
print(f'Dataset Top10:')
run_random_forest(X_train_top10, X_test_top10, y_train_top10, y_test_top10)
print(f'Dataset 90Percent:')
run_random_forest(X_train_90percent, X_test_90percent, y_train_90percent, y_test_90percent)
```

✓ 7m34.4s

Python

Modelos Aplicados

RF - Dataset EQ

	precision	recall	f1-score	support
0	0.00	0.00	0.00	12
4	0.00	0.00	0.00	1101
12	0.00	0.00	0.00	103
13	0.11	0.23	0.15	2488
24	0.00	0.00	0.00	940
41	0.17	0.03	0.04	1575
42	0.00	0.00	0.00	881
43	0.00	0.00	0.00	3272
45	0.00	0.00	0.00	557
48	0.08	0.01	0.02	6814
50	0.00	0.00	0.00	3191
68	0.00	0.00	0.00	6432
74	0.19	0.10	0.13	1763
75	0.05	0.00	0.00	3237
79	0.08	0.23	0.12	6516
87	0.12	0.11	0.11	2203
88	0.00	0.00	0.00	935
90	0.00	0.00	0.00	4437
100	0.00	0.00	0.00	3861
107	0.00	0.00	0.00	6234
113	0.00	0.00	0.00	4021
114	0.00	0.00	0.00	2909
...				
accuracy			0.10	227587
macro avg	0.07	0.06	0.04	227587
weighted avg	0.07	0.10	0.04	227587

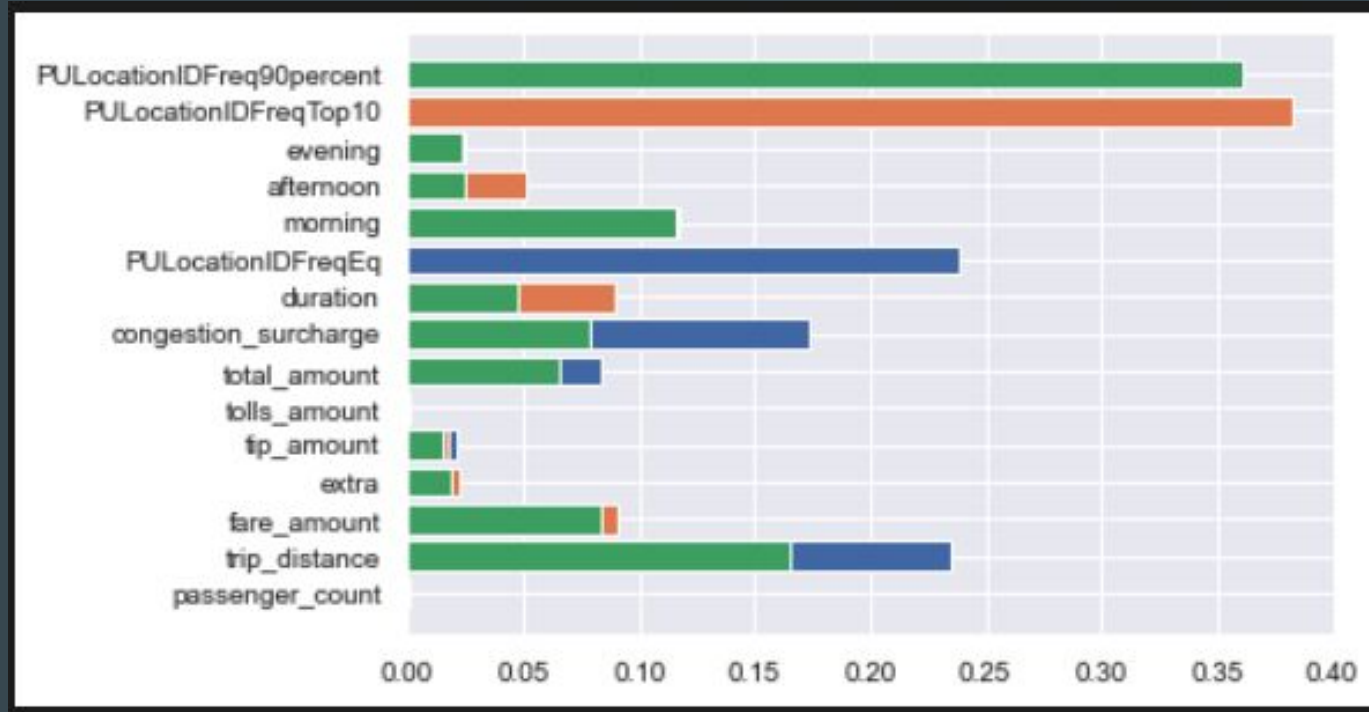
RF - Dataset Top10

	precision	recall	f1-score	support
141	0.00	0.00	0.00	7716
142	0.30	0.02	0.04	8022
161	0.19	0.54	0.29	10569
162	0.00	0.00	0.00	8246
170	0.22	0.31	0.26	8931
230	0.32	0.00	0.00	6940
234	0.36	0.01	0.02	7968
236	0.23	0.68	0.35	13304
237	0.23	0.14	0.17	11606
239	0.40	0.12	0.19	7892
accuracy			0.22	91194
macro avg	0.23	0.18	0.13	91194
weighted avg	0.23	0.22	0.15	91194

RF - Dataset 90%

	precision	recall	f1-score	support
13	0.20	0.24	0.22	2447
43	0.00	0.00	0.00	3274
48	0.15	0.03	0.05	6839
50	0.00	0.00	0.00	3234
68	0.00	0.00	0.00	6408
75	0.34	0.12	0.18	3216
79	0.09	0.25	0.13	6512
90	0.00	0.00	0.00	4445
100	0.00	0.00	0.00	3832
107	0.00	0.00	0.00	6268
113	0.00	0.00	0.00	4015
114	0.00	0.00	0.00	2906
137	0.00	0.00	0.00	3943
140	0.00	0.00	0.00	5676
141	0.00	0.00	0.00	7716
142	0.22	0.02	0.04	8023
143	0.00	0.00	0.00	4307
148	0.00	0.00	0.00	2709
151	0.42	0.10	0.16	2448
158	0.00	0.00	0.00	2885
161	0.09	0.55	0.16	10569
162	0.20	0.00	0.01	8246
163	0.00	0.00	0.00	6388
...				
accuracy			0.10	227432
macro avg	0.07	0.06	0.04	227432
weighted avg	0.07	0.10	0.05	227432

Pesos de las variables de RF



Modelos Aplicados - Logistic Regression

```
def run_lr(x_tr, x_te, y_tr, y_te):  
    scaler = StandardScaler()  
    X_train_sc = scaler.fit_transform(x_tr) # Estandarizamos los datos  
    X_test_sc = scaler.transform(x_te)  
  
    lr = LogisticRegression(random_state = 1, max_iter=300, n_jobs=-1)  
    lr.fit(X_train_sc, y_tr)  
    y_lr_pred = lr.predict(X_test_sc)  
    print(classification_report(y_te, y_lr_pred))  
  
print(f'Dataset EQ:')  
run_lr(X_train_eq, X_test_eq, y_train_eq, y_test_eq)  
print(f'Dataset Top10:')  
run_lr(X_train_top10, X_test_top10, y_train_top10, y_test_top10)  
print(f'Dataset 90Percent:')  
run_lr(X_train_90percent, X_test_90percent, y_train_90percent, y_test_90percent)
```

Modelos Aplicados

LR - Dataset EQ

	precision	recall	f1-score	support
0	0.00	0.00	0.00	12
4	0.00	0.00	0.00	1103
12	0.00	0.00	0.00	108
13	0.08	0.11	0.09	2447
24	0.00	0.00	0.00	900
41	0.11	0.05	0.07	1561
42	0.10	0.02	0.03	870
43	0.00	0.00	0.00	3274
45	0.00	0.00	0.00	570
48	0.04	0.00	0.00	6839
50	0.00	0.00	0.00	3234
68	0.00	0.00	0.00	6408
74	0.09	0.01	0.02	1781
75	0.14	0.02	0.03	3216
79	0.06	0.35	0.10	6513
87	0.03	0.03	0.03	2255
88	0.22	0.01	0.02	967
90	0.00	0.00	0.00	4445
100	0.03	0.00	0.00	3832
107	0.00	0.00	0.00	6268
113	0.00	0.00	0.00	4015
114	0.00	0.00	0.00	2906
116	0.00	0.00	0.00	576
...				
macro avg	0.17	0.15	0.10	91194
weighted avg	0.17	0.18	0.12	91194

LR - Dataset Top10

	precision	recall	f1-score	support
141	0.18	0.02	0.03	7716
142	0.20	0.00	0.00	8022
161	0.18	0.42	0.25	10569
162	0.21	0.00	0.00	8246
170	0.17	0.25	0.20	8931
230	0.16	0.00	0.01	6940
234	0.04	0.00	0.00	7968
236	0.18	0.58	0.28	13304
237	0.19	0.15	0.17	11606
239	0.18	0.04	0.06	7892
accuracy			0.18	91194
macro avg	0.17	0.15	0.10	91194
weighted avg	0.17	0.18	0.12	91194

accuracy = 0.07

LR - Dataset 90%

13	0.14	0.13	0.14	2447
43	0.00	0.00	0.00	3274
48	0.04	0.00	0.00	6839
50	0.00	0.00	0.00	3234
68	0.00	0.00	0.00	6408
75	0.30	0.09	0.14	3216
79	0.07	0.35	0.11	6512
90	0.00	0.00	0.00	4445
100	0.00	0.00	0.00	3832
107	0.17	0.00	0.00	6268
113	0.00	0.00	0.00	4015
114	0.00	0.00	0.00	2906
137	0.00	0.00	0.00	3943
140	0.00	0.00	0.00	5676
141	0.05	0.00	0.00	7716
142	0.00	0.00	0.00	8023
143	0.00	0.00	0.00	4307
148	0.06	0.00	0.01	2709
151	0.34	0.12	0.18	2448
158	0.00	0.00	0.00	2885
161	0.08	0.40	0.13	10569
162	0.00	0.00	0.00	8246
163	0.00	0.00	0.00	6388
...				
accuracy			0.08	227432
macro avg	0.05	0.05	0.03	227432
weighted avg	0.05	0.08	0.04	227432

Conclusiones

Modelo	Dataset Eq	Dataset Top10	Dataset 90%
Random Forest	10%	22%	10%
Logistic Regression	8%	18%	8%

PyCaret

	Model	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC	TT (Sec)
rf	Random Forest Classifier	0.1666	0.8195	0.1470	0.1600	0.1624	0.1439	0.1439	317.7660
dt	Decision Tree Classifier	0.1391	0.5587	0.1274	0.1394	0.1392	0.1169	0.1169	5.8300
ridge	Ridge Classifier	0.0690	0.0000	0.0301	0.0244	0.0250	0.0279	0.0321	0.8780
nb	Naive Bayes	0.0608	0.6045	0.0593	0.0230	0.0233	0.0251	0.0284	2.0740
ada	Ada Boost Classifier	0.0554	0.5536	0.0383	0.0164	0.0159	0.0113	0.0170	44.9280
svm	SVM - Linear Kernel	0.0265	0.0000	0.0175	0.0125	0.0047	0.0024	0.0036	53.6530
qda	Quadratic Discriminant Analysis	0.0004	0.0000	0.0152	0.0000	0.0000	0.0000	0.0000	2.4550

¡Muchas gracias!