

# Prueba técnica Data Science

Ignacio David Vázquez Pérez

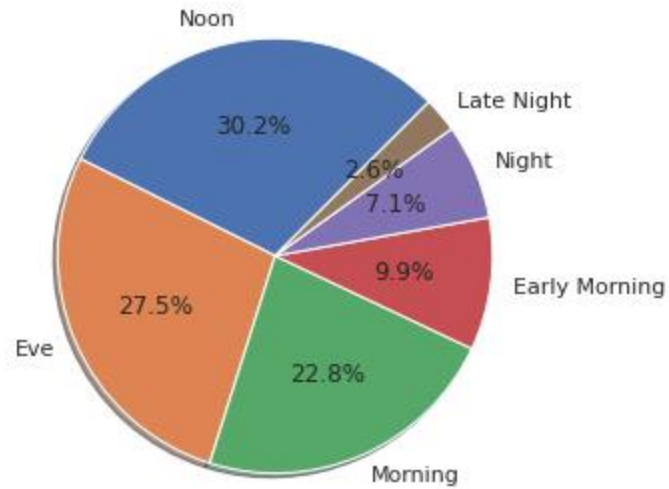
# Especificaciones técnicas requeridas:

- Incluir documentación del flujo de trabajo: A través de un documento en pdf explicar procedimiento correspondiente con cada tarea y subtarea. Es decir, agregar las visualizaciones resultantes del exploratorio de los datos, justificar la selección del o los modelos así como la interpretación de los mismos y demás puntos que consideres importantes para tu evaluación.
- Mantener en el código sólo lo necesario para la funcionalidad a realizar, quitar código comentado, así como librerías y código innecesario.
- Crear un repositorio en github, subir código y compartírnos tu url

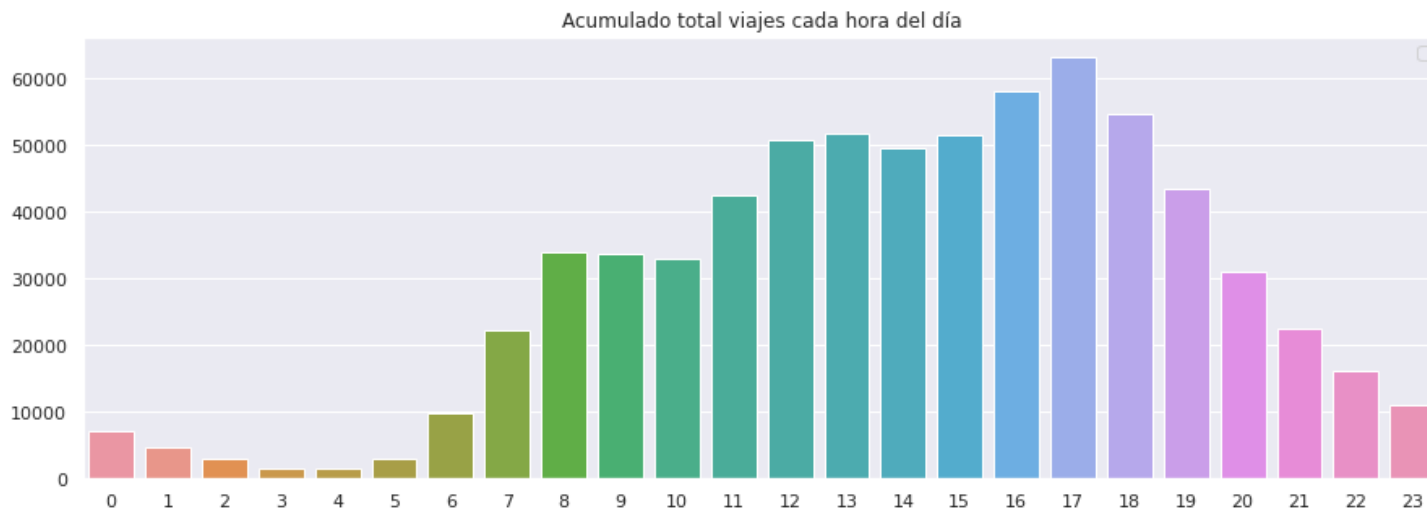
# EDA

- Primero se realizó un análisis de todas las columnas del dataset `train_set`, donde se realizó la operación de valores y el número de elementos de este, para poder familiarizarse con los atributos.
- Las fechas se cambiaron a un formato único de tipo fecha para poder realizar operaciones y graficas.
- Se creo una nueva columna para categorizar el tiempo del dia de la hora de cada viaje, se clasifico como: Early, Morning, Morning, Noon, Eve, Night, Late Night.

# Patrones de tiempo

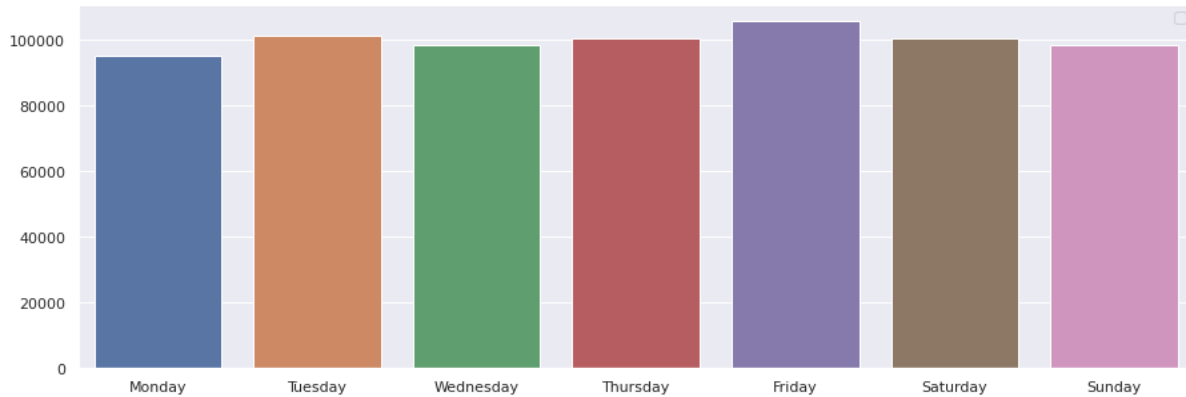


La suma total de usuarios tiene el patrón del consumo del servicio de la mañana, medio día y la tarde. Casi nadie usa el servicio en la madrugada, la noche y muy noche.



# Patrones de tiempo

Acumulado total de lecturas cada día de la semana

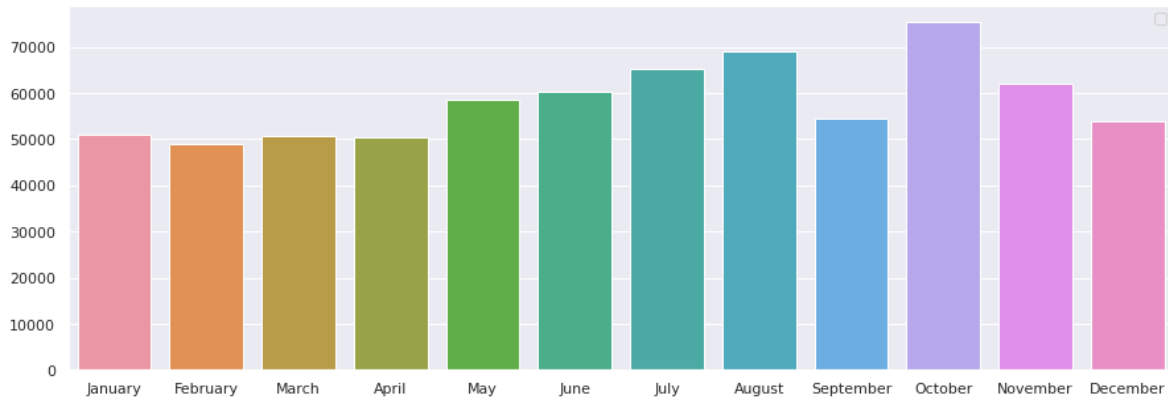


El patrón de consumo no cambia mucho entre los días de la semana, solamente el viernes es donde hay más personas utilizando el servicio.

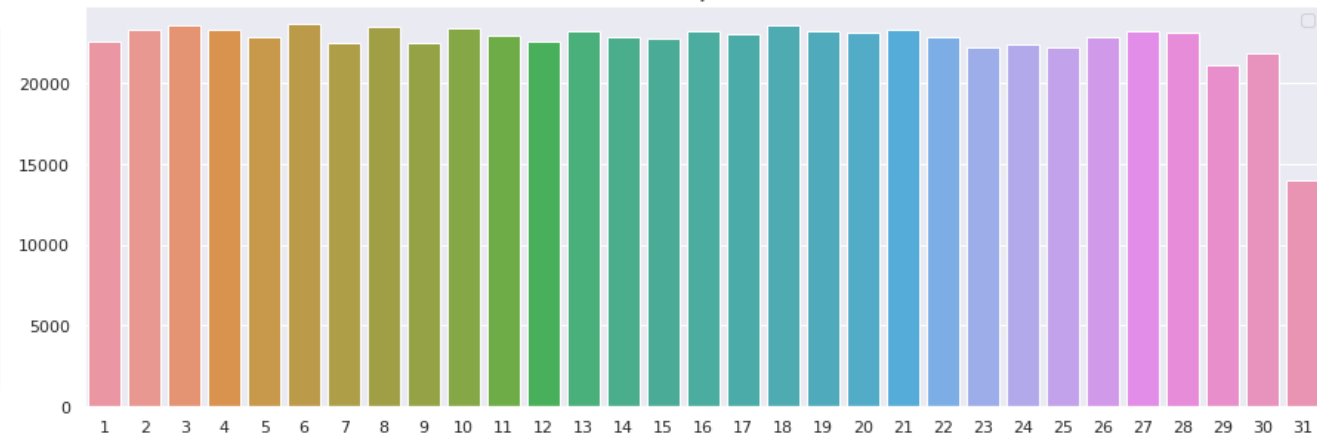
Durante los meses de mayo hasta agosto hay una tendencia de consumo al alza, se interrumpe en septiembre y luego el mes con tasa más alta es octubre donde hay más consumo en todo el año.

Durante los días del mes se mantiene igual, solamente decrece hacia finales del mes

Acumulado total de lecturas cada día de la semana

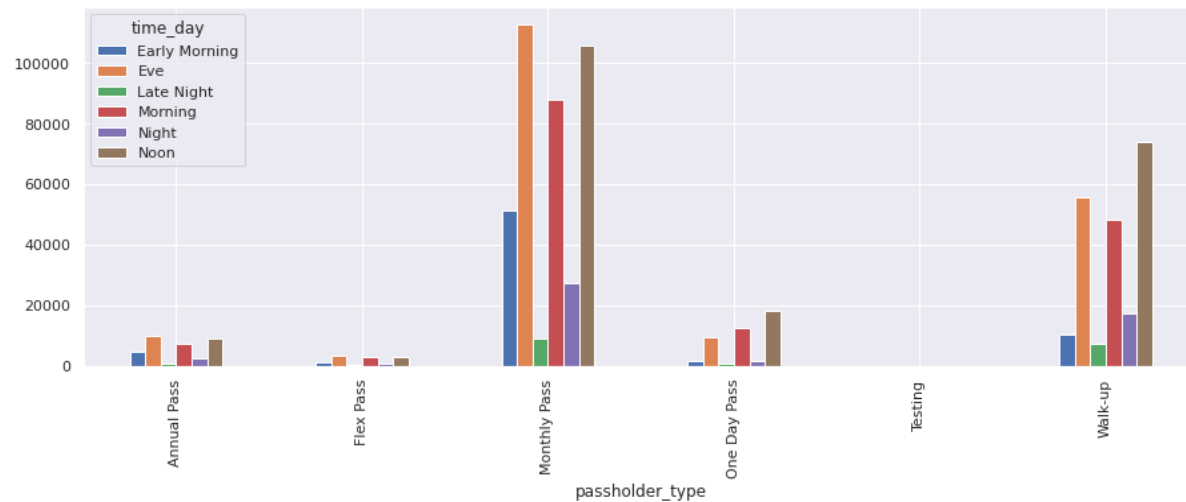
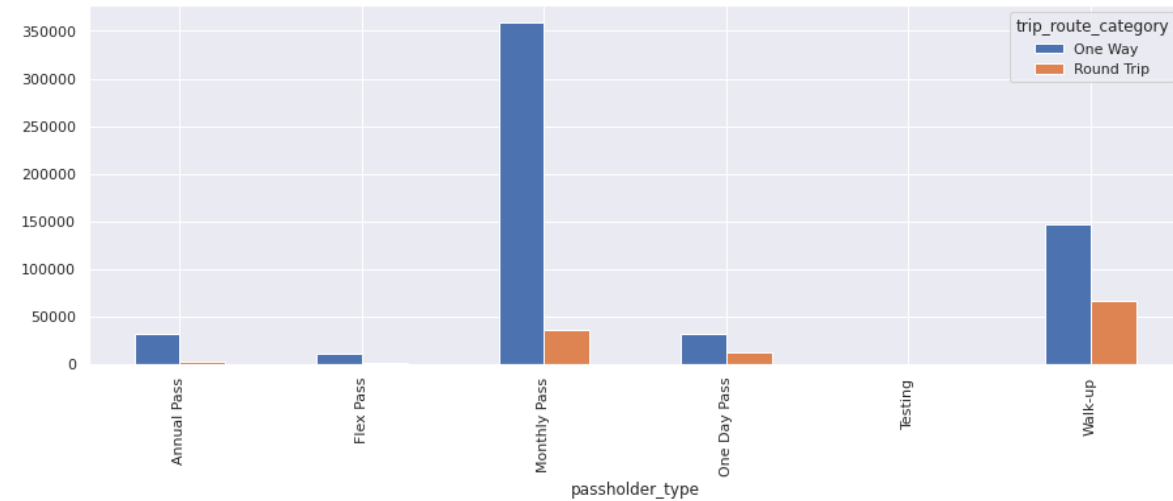


Acumulado total viajes cada día del mes

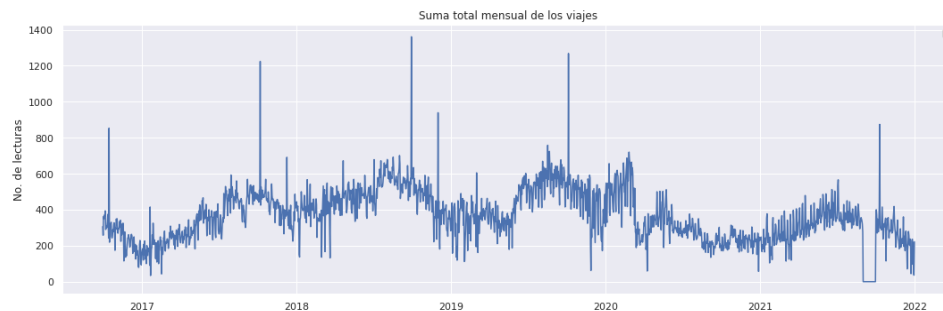
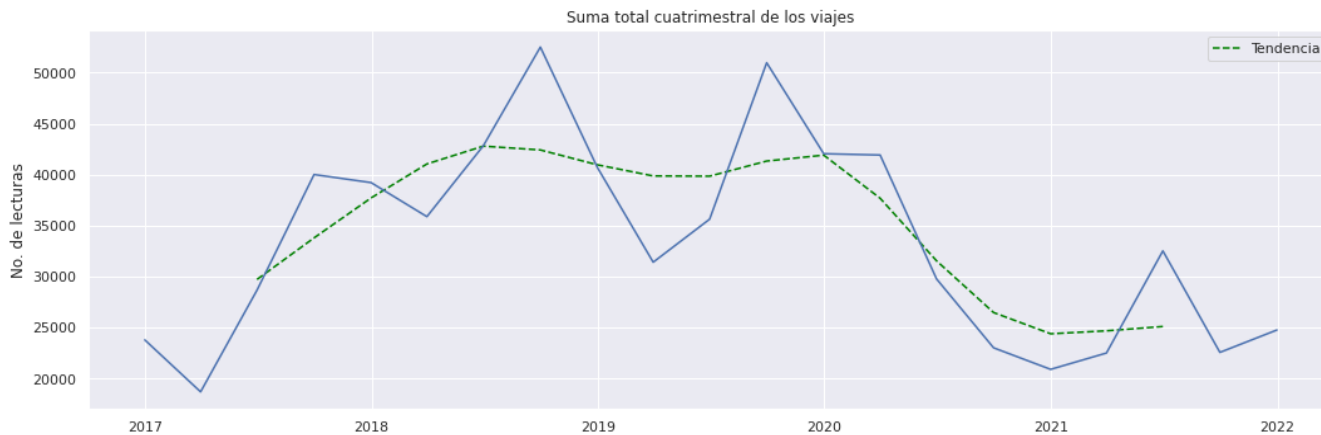
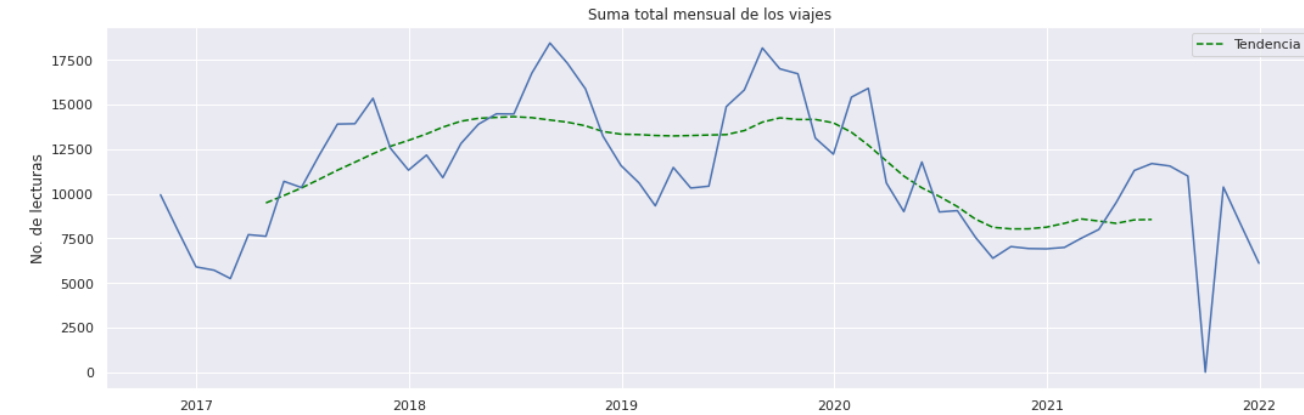


# Patrones por tipo de membresía

Se realizó un agrupamiento por tipo de membresía para saber los patrones respecto a otras variables como son el tipo de ruta y el tiempo del día donde se realizan los viajes. El grupo de membresía más activo es el del pase mensual, donde el tiempo del día donde más realizan viajes es en la tarde, mediodía y la mañana respectivamente. Todos los tipos de membresía su tipo de ruta la mayoría de las veces es de un solo viaje. El pase walk-up queda en segundo lugar de consumo, y realizan más viajes durante el mediodía, la tarde y después la mañana.



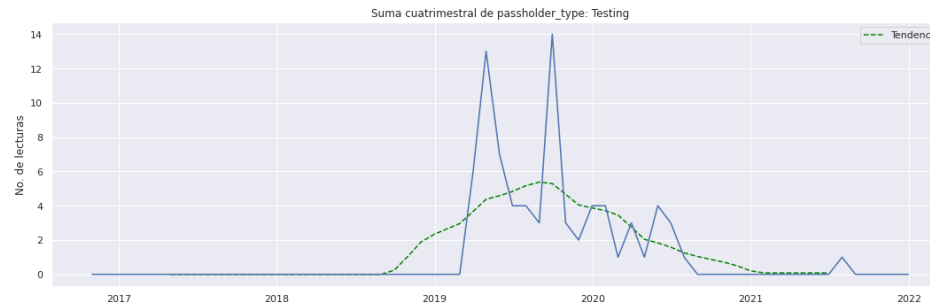
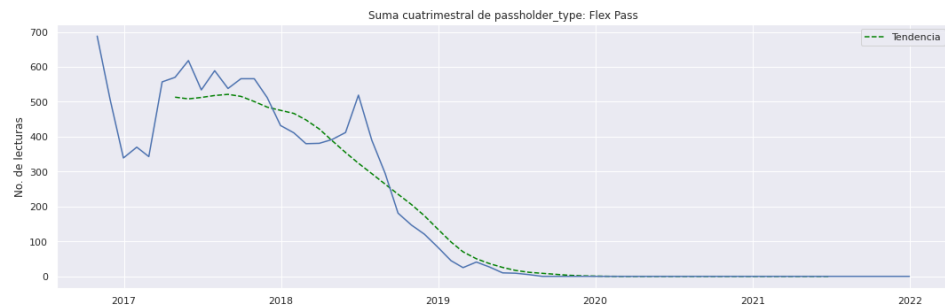
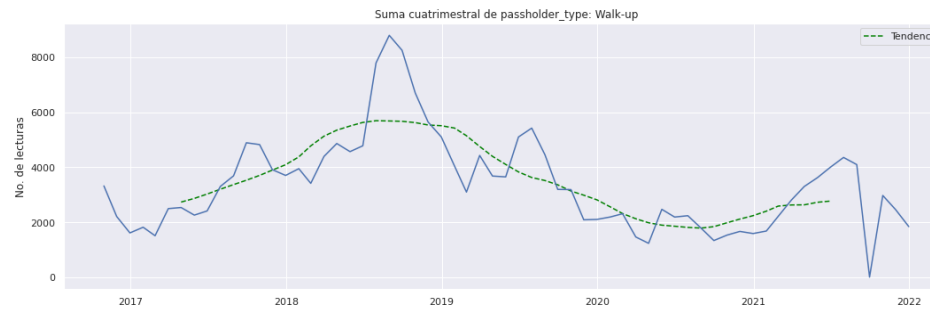
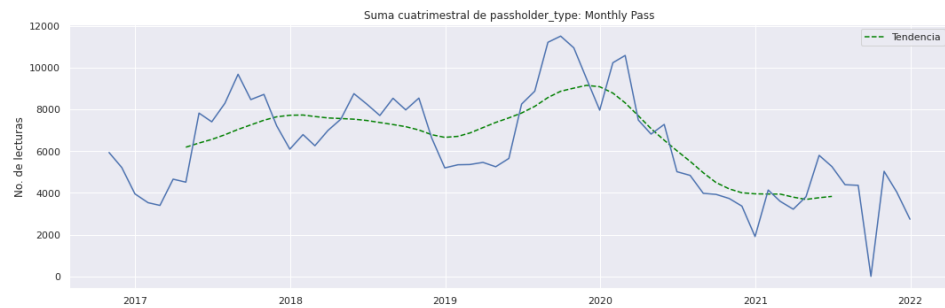
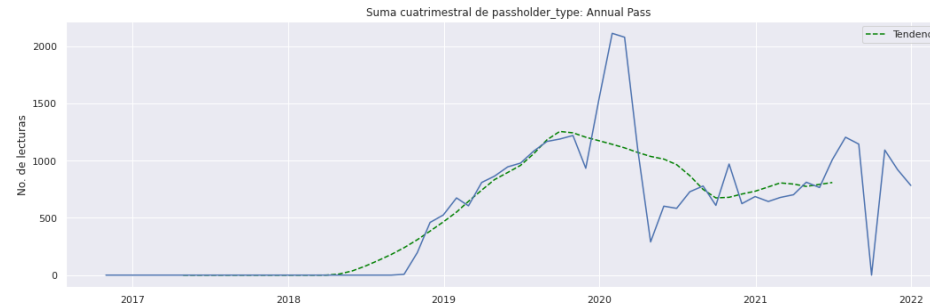
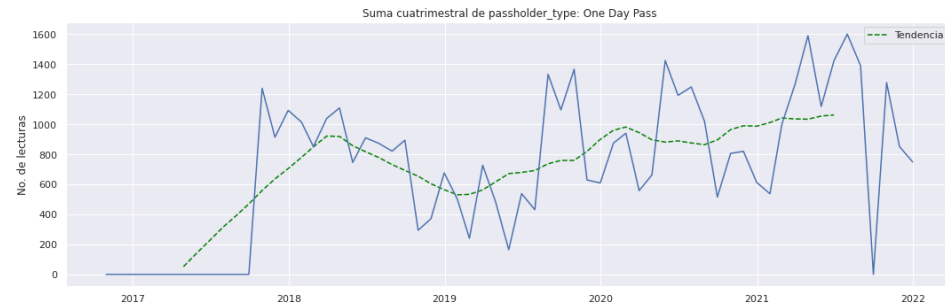
# Series de tiempo



Esta grafica de serie de tiempo muestra la suma total mensual y cuatrimestral de todos los viajes del servicio. La línea verde es la tendencia. En ambas gráficas se muestra una tendencia a la baja respecto a años anteriores, hasta la fecha esta tendencia se ha mantenido estable pero no muestra un crecimiento significativo.

La ultima grafica muestra la suma diaria de todos los viajes del servicio, se muestra una baja importante antes del 2022 que no corresponde al patrón de consumo de los años atrás.

# Series de tiempo



Se graficaron las series de tiempo por cada tipo de membresía. Los 3 tipos de membresía que tienen una tendencia a la alza esta el pase de un día, el pase anual se manera poco significativa, y el walk-up. Tanto el pase mensual y el flex pass tienen tendencia a la baja. Se grafico el testing para entender su comportamiento.



# Agrupaciones

|               |               | time_day_Count |
|---------------|---------------|----------------|
| time_day      | start_station |                |
| Eve           | 3005          | 9805           |
| Noon          | 3030          | 9359           |
| Early Morning | 3014          | 9041           |
| Noon          | 4214          | 7553           |
|               | 3005          | 6745           |
| Morning       | 3005          | 5942           |
| Noon          | 4210          | 5562           |
| Eve           | 3035          | 4768           |
|               | 3064          | 4759           |
|               | 4214          | 4731           |
| Noon          | 3035          | 4683           |
| Eve           | 3030          | 4585           |
|               | 3031          | 4520           |
| Noon          | 4215          | 4261           |
|               | 3031          | 4254           |

|       |               |           | count | mean       |
|-------|---------------|-----------|-------|------------|
| 46648 | Noon          | 3030_3014 | 4487  | 7.224426   |
| 410   | Early Morning | 3014_3030 | 4281  | 7.431675   |
| 51582 | Noon          | 4214_4214 | 3377  | 76.597868  |
| 51429 | Noon          | 4210_4210 | 2336  | 81.072774  |
| 13451 | Eve           | 4214_4214 | 2116  | 71.866730  |
| 30770 | Morning       | 4214_4214 | 1921  | 79.900573  |
| 51583 | Noon          | 4214_4215 | 1905  | 52.469816  |
| 30635 | Morning       | 4210_4210 | 1715  | 93.072886  |
| 13306 | Eve           | 4210_4210 | 1503  | 77.616766  |
| 51631 | Noon          | 4215_4215 | 1358  | 100.886598 |
| 51434 | Noon          | 4210_4215 | 1347  | 62.490720  |
| 13452 | Eve           | 4214_4215 | 1296  | 50.665895  |
| 54063 | Noon          | 4345_4345 | 1181  | 41.881456  |
| 30771 | Morning       | 4214_4215 | 1155  | 51.219913  |
| 8750  | Eve           | 3030_3014 | 1069  | 8.701590   |

Se realizó la agrupación por la estación más recurrente por el tiempo del día, tanto estación individual como la combinación de la estación donde se comienza y termina. Se recomienda tomar en consideración ampliar el servicio en estas estaciones para favorecer al usuario con más estaciones y bicicletas. También se comprueba la correlación con el uso del servicio con el tiempo del día. La combinación de estaciones es la misma y se valida que la gente utiliza la misma ruta, pero a diferentes horarios del día.

# Agrupaciones

|                 |                     |               |               | time_day_Count |
|-----------------|---------------------|---------------|---------------|----------------|
| passholder_type | trip_route_category | time_day      | start_station |                |
| Monthly Pass    | One Way             | Early Morning | 3014          | 7978           |
|                 |                     | Noon          | 3030          | 7863           |
|                 |                     | Eve           | 3005          | 6719           |
|                 |                     | Noon          | 3005          | 4204           |
|                 |                     | Morning       | 3005          | 3708           |
|                 |                     | Eve           | 3030          | 3535           |
|                 |                     | Early Morning | 3042          | 3514           |
|                 |                     | Eve           | 3064          | 3396           |
| Walk-up         | One Way             | Noon          | 3035          | 3197           |
|                 |                     |               | 4214          | 3120           |
| Monthly Pass    | One Way             | Eve           | 3031          | 3073           |
|                 |                     | Noon          | 3035          | 2907           |
| Walk-up         | Round Trip          | Noon          | 4214          | 2832           |
| Monthly Pass    | One Way             | Morning       | 3031          | 2798           |
|                 |                     | Noon          | 3031          | 2748           |

Se realizó la agrupación con las diferentes variables categóricas y se contó cuantas veces se repite. Se comprueba que, dependiendo del tipo de membresía, el usuario prefiere un tipo de ruta en específico, así como el tiempo del día y la estación. El pase mensual es el más común que realice viajes de manera más concurrente, y el pase de un solo día se realiza en su mayoría al medio día en la misma estación.

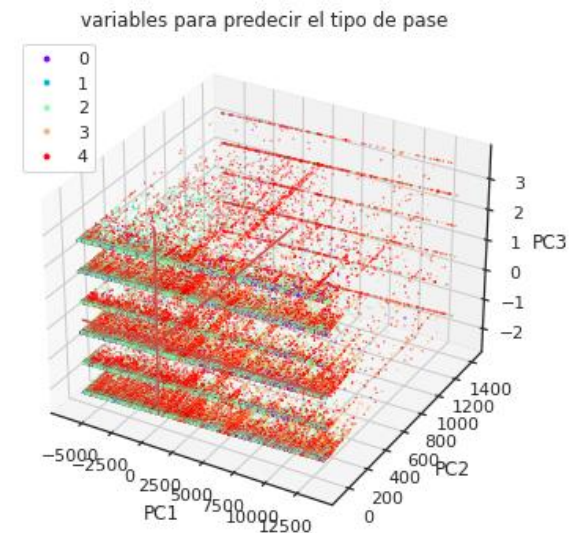
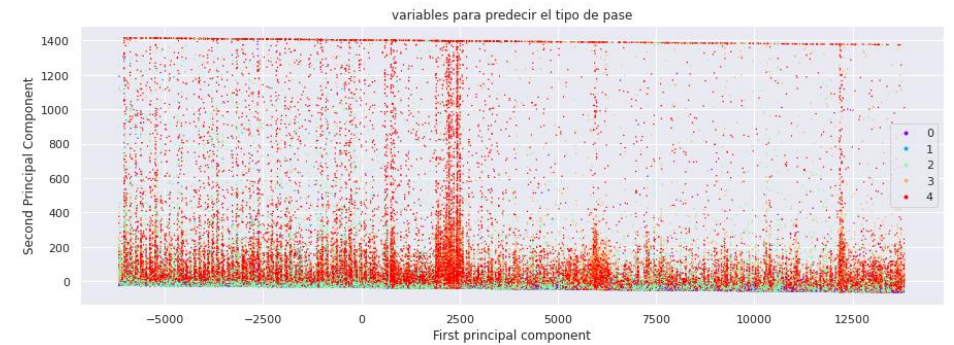
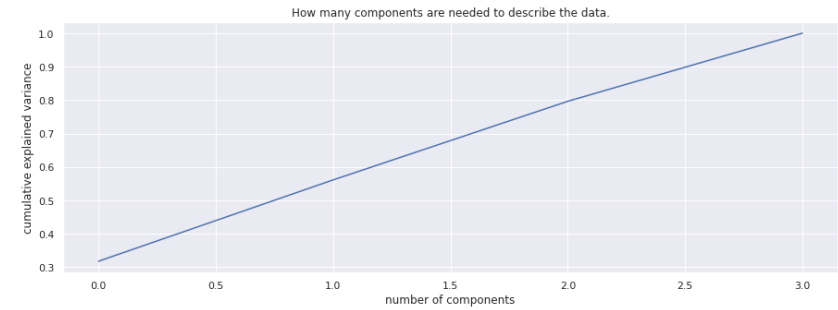
# Agrupaciones

|        |              |            |               |           | count | mean       |
|--------|--------------|------------|---------------|-----------|-------|------------|
| 50001  | Monthly Pass | One Way    | Noon          | 3030_3014 | 4359  | 6.849736   |
| 17192  | Monthly Pass | One Way    | Early Morning | 3014_3030 | 4235  | 7.337662   |
| 108323 | Walk-up      | Round Trip | Noon          | 4214_4214 | 2832  | 76.681497  |
| 108319 | Walk-up      | Round Trip | Noon          | 4210_4210 | 1891  | 78.528821  |
| 107170 | Walk-up      | Round Trip | Eve           | 4214_4214 | 1826  | 72.207010  |
| 107716 | Walk-up      | Round Trip | Morning       | 4214_4214 | 1585  | 79.434069  |
| 102884 | Walk-up      | One Way    | Noon          | 4214_4215 | 1513  | 53.352941  |
| 107712 | Walk-up      | Round Trip | Morning       | 4210_4210 | 1371  | 91.079504  |
| 107166 | Walk-up      | Round Trip | Eve           | 4210_4210 | 1242  | 74.282609  |
| 102765 | Walk-up      | One Way    | Noon          | 4210_4215 | 1144  | 63.715909  |
| 81864  | Walk-up      | One Way    | Eve           | 4214_4215 | 1068  | 53.606742  |
| 23303  | Monthly Pass | One Way    | Eve           | 3030_3014 | 998   | 8.546092   |
| 108380 | Walk-up      | Round Trip | Noon          | 4345_4345 | 973   | 40.684481  |
| 18280  | Monthly Pass | One Way    | Early Morning | 3042_3030 | 886   | 5.820542   |
| 108324 | Walk-up      | Round Trip | Noon          | 4215_4215 | 885   | 100.714124 |

En esta agrupación se realizó con la combinación de las dos estaciones y el promedio del recorrido. El pase de un mes tiene una ruta definida. EL pase de un solo día se realiza con rutas más largas de tiempo en promedio. A su vez se comprueba que las variables categóricas así como la duración del viaje tiene una correlación con el tipo de pase. Se va a considerar esta tabla para realizar el entrenamiento del modelo.

# Modelo analítico

- Se juntaron las dos columnas de estaciones en una sola para usarse como una variable categórica de predicción
- Las variables para entrenar el modelo son las siguientes: 'duration', 'trip\_route\_category', 'station\_route', 'time\_day', 'passholder\_type'
- Se convirtieron a variables categóricas excepto la columna de duración
- Se eliminaron las filas que tienen el tipo de pase "Testing" ya que no aporta a los datos
- Se realizó un Principal Component Analysis para visualizar la separabilidad de los datos en 2 y 3 dimensiones, así como el número de componentes que se necesitan para describir el tipo de pase.
- Debido a la no separabilidad de los datos, se consideraron algoritmos de clasificación que no sean lineales

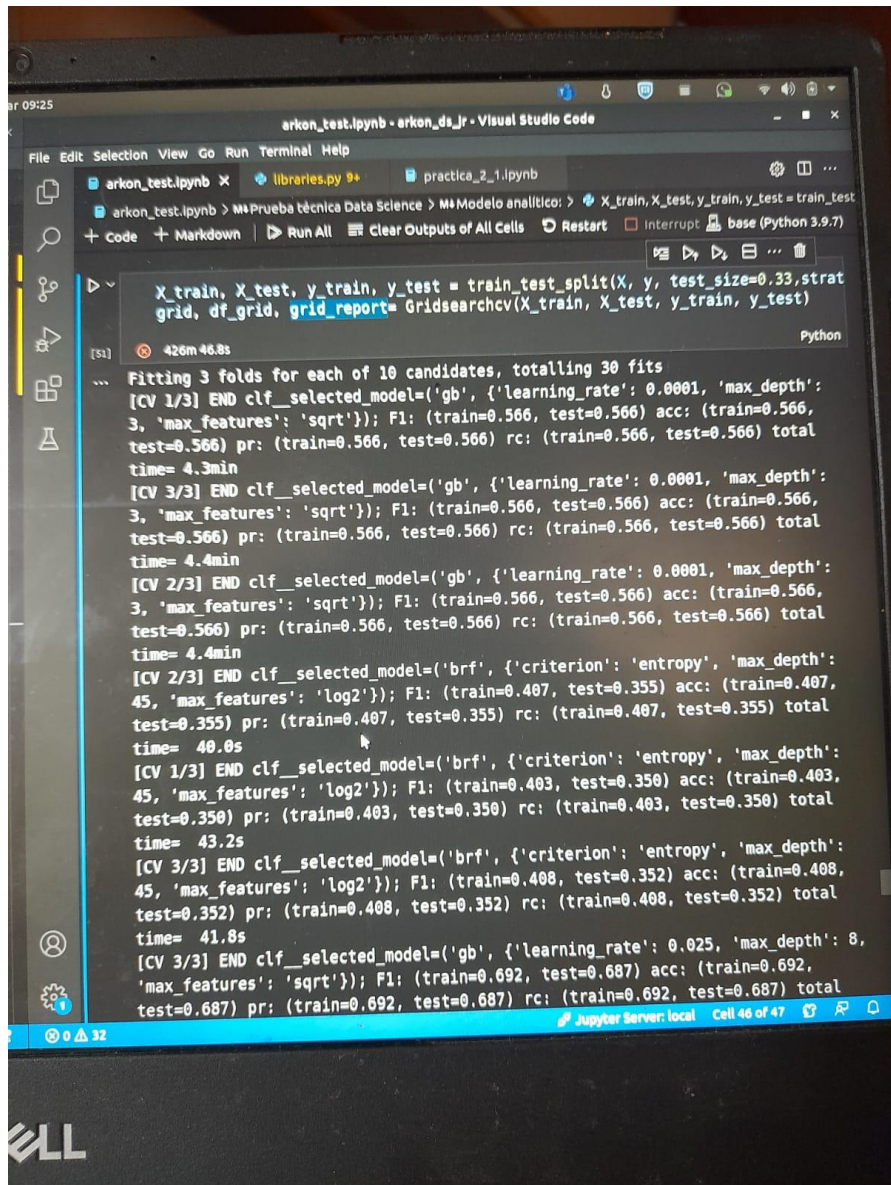


# Modelo analitico

- Se realizó una búsqueda aleatoria con validación cruzada personalizada utilizando diferentes modelos de clasificación. AL principio se descartaron los demás modelos que no tenían una accuracy más de 60% y se fueron descartando los modelos sencillos.
- Se realizaron numerosas pruebas con los algoritmos de Support Vector Machine, Balanced Random Forest (es un modelo modificado para datos imbalanceados como es el caso), y Gradient Boosting.
- Estos modelos se probaron combinaciones aleatorias de sus hiperparametros, y cada modelo con su configuración se entrenó 3 veces para evitar el sobre entrenamiento. Al final se registraron las métricas de cada modelo, así como el tiempo de entrenamiento N veces (al final probe con 30, 15, y 10 iteraciones, pero el tiempo fue demasiado y decidí terminarlo).
- Debido al volumen del dataset, al realizar numerosas veces el entrenamiento, mi laptop llegó al tope de recursos (tanto de memoria RAM como de procesamiento), y por las impresiones de pantalla, se eligió al mejor modelo y sus hiperparametros aunque no haya terminado el proceso completo.

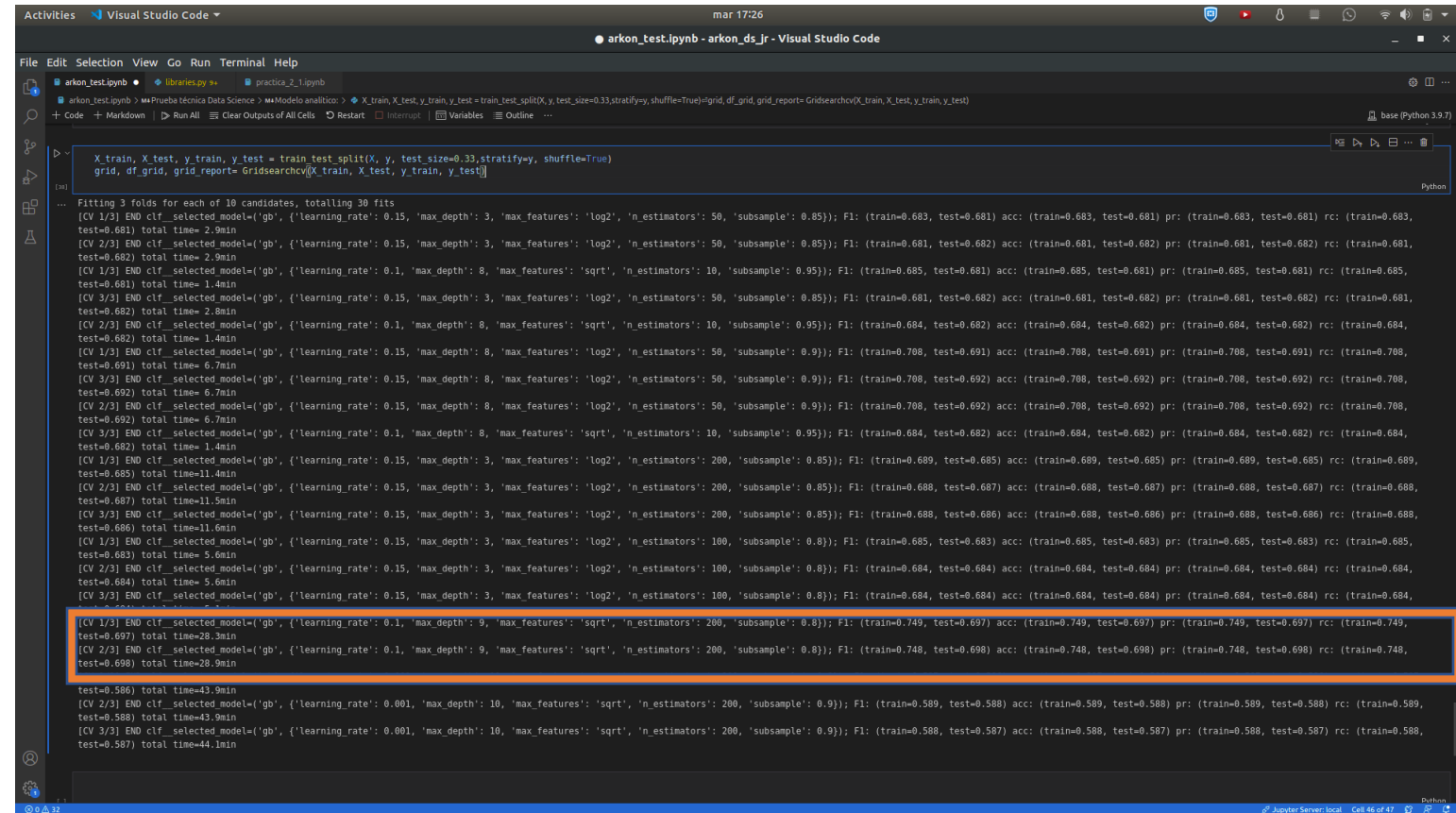


El modelo con mejores métricas es computacionalmente más demandante pero la diferencia era mucha comparado con los demás y se eligió el modelo gradient boost con esos mismos hiperparametros.



```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.33, stratify=y, shuffle=True, random_state=42)
grid = GridSearchCV(X_train, X_test, y_train, y_test)

Fitting 3 folds for each of 10 candidates, totalling 30 fits
[CV 1/3] END clf_selected_model='gb', {'learning_rate': 0.001, 'max_depth': 3, 'max_features': 'sqrt'}; F1: (train=0.566, test=0.566) acc: (train=0.566, test=0.566) pr: (train=0.566, test=0.566) rc: (train=0.566, test=0.566) total time= 4.3min
[CV 3/3] END clf_selected_model='gb', {'learning_rate': 0.001, 'max_depth': 3, 'max_features': 'sqrt'}; F1: (train=0.566, test=0.566) acc: (train=0.566, test=0.566) pr: (train=0.566, test=0.566) rc: (train=0.566, test=0.566) total time= 4.4min
[CV 2/3] END clf_selected_model='gb', {'learning_rate': 0.001, 'max_depth': 3, 'max_features': 'sqrt'}; F1: (train=0.566, test=0.566) acc: (train=0.566, test=0.566) pr: (train=0.566, test=0.566) rc: (train=0.566, test=0.566) total time= 4.4min
[CV 2/3] END clf_selected_model='brf', {'criterion': 'entropy', 'max_depth': 45, 'max_features': 'log2'}; F1: (train=0.407, test=0.355) acc: (train=0.407, test=0.355) pr: (train=0.407, test=0.355) rc: (train=0.407, test=0.355) total time= 40.0s
[CV 1/3] END clf_selected_model='brf', {'criterion': 'entropy', 'max_depth': 45, 'max_features': 'log2'}; F1: (train=0.403, test=0.350) acc: (train=0.403, test=0.350) pr: (train=0.403, test=0.350) rc: (train=0.403, test=0.350) total time= 43.2s
[CV 3/3] END clf_selected_model='brf', {'criterion': 'entropy', 'max_depth': 45, 'max_features': 'log2'}; F1: (train=0.408, test=0.352) acc: (train=0.408, test=0.352) pr: (train=0.408, test=0.352) rc: (train=0.408, test=0.352) total time= 41.8s
[CV 3/3] END clf_selected_model='gb', {'learning_rate': 0.025, 'max_depth': 8, 'max_features': 'sqrt'}; F1: (train=0.692, test=0.687) acc: (train=0.692, test=0.687) pr: (train=0.692, test=0.687) rc: (train=0.692, test=0.687) total time= 44.1min
```



```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.33, stratify=y, shuffle=True, random_state=42)
grid = GridSearchCV(X_train, X_test, y_train, y_test)

Fitting 3 folds for each of 10 candidates, totalling 30 fits
[CV 1/3] END clf_selected_model='gb', {'learning_rate': 0.15, 'max_depth': 3, 'max_features': 'log2', 'n_estimators': 50, 'subsample': 0.85}; F1: (train=0.683, test=0.681) acc: (train=0.683, test=0.681) pr: (train=0.683, test=0.681) rc: (train=0.683, test=0.681) total time= 2.9min
[CV 2/3] END clf_selected_model='gb', {'learning_rate': 0.15, 'max_depth': 3, 'max_features': 'log2', 'n_estimators': 50, 'subsample': 0.85}; F1: (train=0.681, test=0.682) acc: (train=0.681, test=0.682) pr: (train=0.681, test=0.682) rc: (train=0.681, test=0.682) total time= 2.9min
[CV 3/3] END clf_selected_model='gb', {'learning_rate': 0.1, 'max_depth': 8, 'max_features': 'sqrt', 'n_estimators': 10, 'subsample': 0.95}; F1: (train=0.685, test=0.681) acc: (train=0.685, test=0.681) pr: (train=0.685, test=0.681) rc: (train=0.685, test=0.681) total time= 1.4min
[CV 2/3] END clf_selected_model='gb', {'learning_rate': 0.15, 'max_depth': 3, 'max_features': 'log2', 'n_estimators': 50, 'subsample': 0.85}; F1: (train=0.681, test=0.682) acc: (train=0.681, test=0.682) pr: (train=0.681, test=0.682) rc: (train=0.681, test=0.682) total time= 2.8min
[CV 2/3] END clf_selected_model='gb', {'learning_rate': 0.1, 'max_depth': 8, 'max_features': 'sqrt', 'n_estimators': 10, 'subsample': 0.95}; F1: (train=0.684, test=0.682) acc: (train=0.684, test=0.682) pr: (train=0.684, test=0.682) rc: (train=0.684, test=0.682) total time= 1.4min
[CV 1/3] END clf_selected_model='gb', {'learning_rate': 0.15, 'max_depth': 8, 'max_features': 'log2', 'n_estimators': 50, 'subsample': 0.9}; F1: (train=0.708, test=0.691) acc: (train=0.708, test=0.691) pr: (train=0.708, test=0.691) rc: (train=0.708, test=0.691) total time= 6.7min
[CV 3/3] END clf_selected_model='gb', {'learning_rate': 0.15, 'max_depth': 8, 'max_features': 'log2', 'n_estimators': 50, 'subsample': 0.9}; F1: (train=0.708, test=0.692) acc: (train=0.708, test=0.692) pr: (train=0.708, test=0.692) rc: (train=0.708, test=0.692) total time= 6.7min
[CV 2/3] END clf_selected_model='gb', {'learning_rate': 0.15, 'max_depth': 8, 'max_features': 'log2', 'n_estimators': 50, 'subsample': 0.9}; F1: (train=0.708, test=0.692) acc: (train=0.708, test=0.692) pr: (train=0.708, test=0.692) rc: (train=0.708, test=0.692) total time= 6.7min
[CV 3/3] END clf_selected_model='gb', {'learning_rate': 0.1, 'max_depth': 8, 'max_features': 'sqrt', 'n_estimators': 10, 'subsample': 0.95}; F1: (train=0.684, test=0.682) acc: (train=0.684, test=0.682) pr: (train=0.684, test=0.682) rc: (train=0.684, test=0.682) total time= 1.4min
[CV 1/3] END clf_selected_model='gb', {'learning_rate': 0.15, 'max_depth': 3, 'max_features': 'log2', 'n_estimators': 200, 'subsample': 0.85}; F1: (train=0.689, test=0.685) acc: (train=0.689, test=0.685) pr: (train=0.689, test=0.685) rc: (train=0.689, test=0.685) total time= 11.4min
[CV 2/3] END clf_selected_model='gb', {'learning_rate': 0.15, 'max_depth': 3, 'max_features': 'log2', 'n_estimators': 200, 'subsample': 0.85}; F1: (train=0.688, test=0.687) acc: (train=0.688, test=0.687) pr: (train=0.688, test=0.687) rc: (train=0.688, test=0.687) total time= 11.5min
[CV 3/3] END clf_selected_model='gb', {'learning_rate': 0.15, 'max_depth': 3, 'max_features': 'log2', 'n_estimators': 200, 'subsample': 0.85}; F1: (train=0.688, test=0.686) acc: (train=0.688, test=0.686) pr: (train=0.688, test=0.686) rc: (train=0.688, test=0.686) total time= 11.6min
[CV 1/3] END clf_selected_model='gb', {'learning_rate': 0.15, 'max_depth': 3, 'max_features': 'log2', 'n_estimators': 100, 'subsample': 0.8}; F1: (train=0.685, test=0.683) acc: (train=0.685, test=0.683) pr: (train=0.685, test=0.683) rc: (train=0.685, test=0.683) total time= 5.6min
[CV 2/3] END clf_selected_model='gb', {'learning_rate': 0.15, 'max_depth': 3, 'max_features': 'log2', 'n_estimators': 100, 'subsample': 0.8}; F1: (train=0.684, test=0.684) acc: (train=0.684, test=0.684) pr: (train=0.684, test=0.684) rc: (train=0.684, test=0.684) total time= 5.6min
[CV 3/3] END clf_selected_model='gb', {'learning_rate': 0.15, 'max_depth': 3, 'max_features': 'log2', 'n_estimators': 100, 'subsample': 0.8}; F1: (train=0.684, test=0.684) acc: (train=0.684, test=0.684) pr: (train=0.684, test=0.684) rc: (train=0.684, test=0.684) total time= 5.6min
[CV 1/3] END clf_selected_model='gb', {'learning_rate': 0.1, 'max_depth': 9, 'max_features': 'sqrt', 'n_estimators': 200, 'subsample': 0.8}; F1: (train=0.749, test=0.697) acc: (train=0.749, test=0.697) pr: (train=0.749, test=0.697) rc: (train=0.749, test=0.697) total time= 28.3min
[CV 2/3] END clf_selected_model='gb', {'learning_rate': 0.1, 'max_depth': 9, 'max_features': 'sqrt', 'n_estimators': 200, 'subsample': 0.8}; F1: (train=0.748, test=0.698) acc: (train=0.748, test=0.698) pr: (train=0.748, test=0.698) rc: (train=0.748, test=0.698) total time= 28.9min
[CV 3/3] END clf_selected_model='gb', {'learning_rate': 0.001, 'max_depth': 10, 'max_features': 'sqrt', 'n_estimators': 200, 'subsample': 0.9}; F1: (train=0.589, test=0.588) acc: (train=0.589, test=0.588) pr: (train=0.589, test=0.588) rc: (train=0.589, test=0.588) total time= 43.9min
[CV 3/3] END clf_selected_model='gb', {'learning_rate': 0.001, 'max_depth': 10, 'max_features': 'sqrt', 'n_estimators': 200, 'subsample': 0.9}; F1: (train=0.588, test=0.587) acc: (train=0.588, test=0.587) pr: (train=0.588, test=0.587) rc: (train=0.588, test=0.587) total time= 44.1min
```

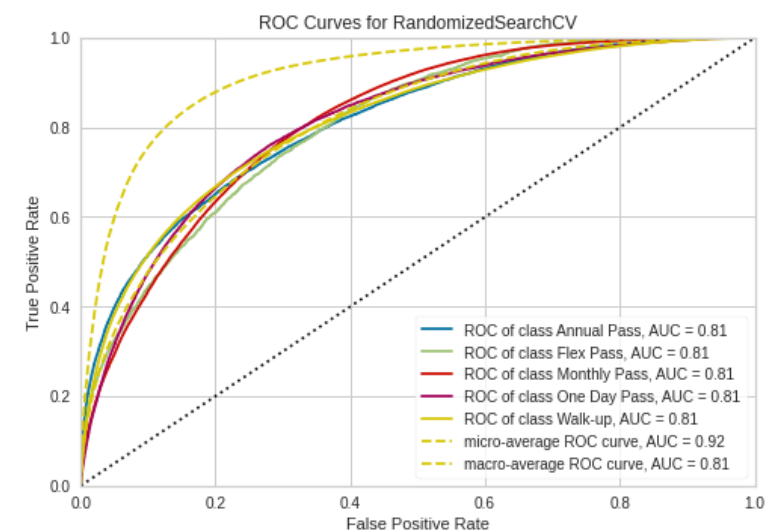
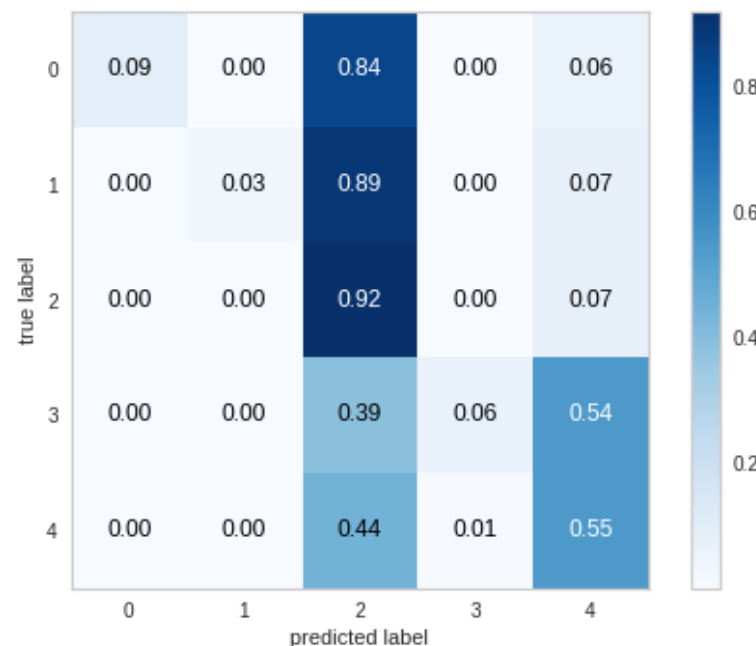


Con cada iteración modifique el número de iteraciones de validación cruzada, el número de hiperparametros, pero siempre llegaba al tope mi laptop de recursos.

- Influye mucho el número de ejemplos de cada etiqueta al momento de realizar una predicción multivariable. Mientras menos ejemplos, menor va a ser la métrica de evaluación
- La predicción más alta fue el pase mensual, seguido con el anual pass y el walk-up pass (aunque walkup tenga más datos). Se confirma que hay un patrón más marcado los usuarios del anual pass aunque no tenga muchos datos.
- En la matriz de confusión, en la categoría de pase mensual se confunde con otro tipo de membresías. Se considera que en las demás no hay un patrón marcado con las variables.
- La curva ROC promedio micro es **la suma de la tasa de verdaderos positivos dividida por la suma de la tasa de falsos positivos, y** se muestra que tienen un buen estado de falsos/positivos, por lo cual este modelo y los datos se puede mejorar para realizar predicciones más altas. No hay ninguna categoría que se encuentre debajo del 50%.

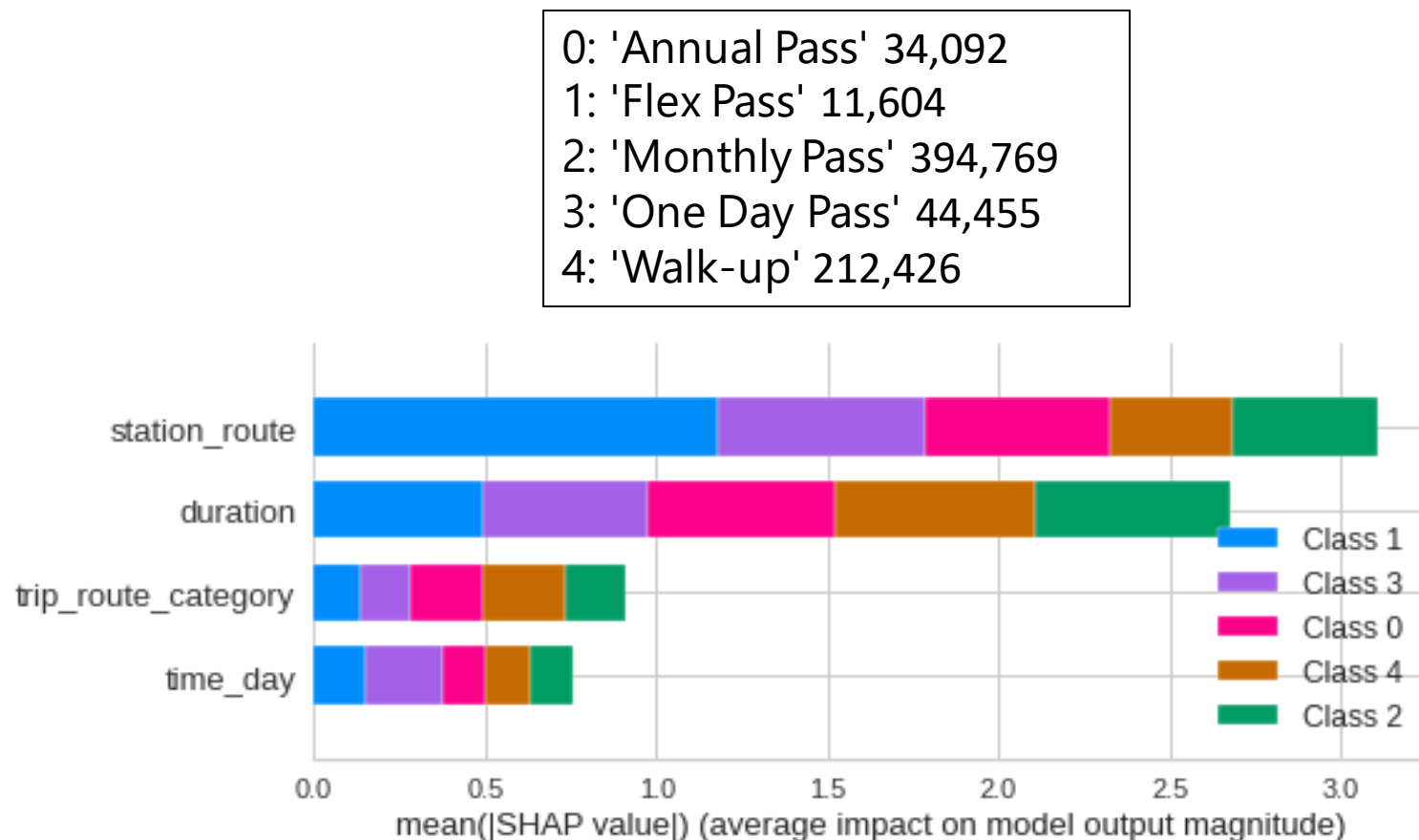
0: 'Annual Pass' 34,092  
 1: 'Flex Pass' 11,604  
 2: 'Monthly Pass' 394,769  
 3: 'One Day Pass' 44,455  
 4: 'Walk-up' 212,426

|              | precision | recall   | f1-score | support       |
|--------------|-----------|----------|----------|---------------|
| 0            | 0.673983  | 0.092792 | 0.163125 | 11251.000000  |
| 1            | 0.299517  | 0.032384 | 0.058449 | 3829.000000   |
| 2            | 0.707686  | 0.918894 | 0.799578 | 130274.000000 |
| 3            | 0.438458  | 0.064349 | 0.112227 | 14670.000000  |
| 4            | 0.672289  | 0.545256 | 0.602146 | 70101.000000  |
| accuracy     | 0.695461  | 0.695461 | 0.695461 | 0.695461      |
| macro avg    | 0.558387  | 0.330735 | 0.347105 | 230125.000000 |
| weighted avg | 0.671302  | 0.695461 | 0.652170 | 230125.000000 |





- Se confirma la hipótesis que dependiendo del tipo de membresía, las variables tienen una diferente proporción de influencia, lo cual se vio influido en el rendimiento de las métricas del modelo
- Se realizó una operación para explicar el peso de las variables después que el modelo se entrenó con la librería SHAP, hay que considerar que se utilizó una pequeña proporción de los datos debido a que este algoritmo es muy demandante computacionalmente



# Conclusiones

- Claramente hay un patrón marcado en el uso del servicio dependiendo del tiempo en el día, que a su vez influye las estaciones que se utilizan por el tipo de membresía
- Se puede considerar la tendencia al alza de usuarios en estaciones específicas para proporcionar un servicio que se adecue a la demanda
- Al tener un desbalance de las etiquetas de predicción, se propone generar datos de manera artificial para probar la eficacia del entrenamiento
- Se comprobó el dependiendo del tipo de usuario, van a tener rutas, horarios, duración del viaje, y el tipo de ruta definido
- Si se pretende vender más membresías, se puede segmentar el tipo de usuario que tiene pase de un día a pase mensual, y los que tienen el pase mensual invitarlos a pagar el pase anual, considerando su patrón de consumo del servicio.
- Entrenar el modelo con más hilos e hiperparametros para poder mejorar las métricas