Using Time Series Predictive Models for Early Detection of Gambling Addiction in Problem Gamblers

Data Gathering

In [1]:
```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

daily_agg_df = pd.read_csv('./datasets/Raw Datset II.Daily aggregates_Gray_LaPlante_PA
rg_det_df = pd.read_csv('./datasets/Raw Datset III.Responsible gambling details_Gray_L
demog_df = pd.read_csv('./datasets/Raw Datset I.Demographics_Gray_LaPlante_PAB_2012.da
```

```
C:\Users\abhiv\AppData\Local\Temp\ipykernel_14916\3515817822.py:5: DtypeWarning: Colu
mns (3,4,5) have mixed types. Specify dtype option on import or set low_memory=False.
  daily_agg_df = pd.read_csv('./datasets/Raw Datset II.Daily aggregates_Gray_LaPlante
_PAB_2012.dat', delimiter='\t')
```

In [2]:
```
rg_det_df
```

Out[2]:

	UserID	RGsumevents	RGFirst_Date	RGLast_date	Event_type_first	Interventiontype_first
0	2169867	1	11/19/2009	11/19/2009	9	18
1	7035862	1	11/15/2009	11/15/2009	9	18
2	5911218	1	11/8/2009	11/8/2009	9	18
3	5872708	1	11/3/2009	11/3/2009	9	18
4	5746942	2	10/18/2009	11/3/2009	9	18
...	...	...	...	...	...	...
2063	2451840	5		11/3/2009	12	
2064	9140426	1	8/19/2009	8/19/2009	10	
2065	2590026	2	8/5/2009	8/6/2009	10	
2066	1023918	1	4/9/2009	4/9/2009	6	
2067	6691324	1	12/19/2008	12/19/2008	1	

2068 rows × 6 columns

In [3]:
```
demog_df
```

Out[3]:

	USERID	RG_case	CountryName	LanguageName	Gender	YearofBirth	Registration_date	First_
0	2975944	1	Germany	German	M	1970	6/2/2006	
1	9822065	1	Germany.COM	German	F	1963	11/21/2009	
2	9622454	1	France.COM	French	F	1981	10/19/2009	
3	9619356	1	Italy.IT	Italian	F	1975	10/18/2009	
4	9593498	1	Germany.COM	German	F	1990	10/14/2009	
...	...	...	...	...	...	...	...	
4129	107292	0	Austria	German	M	1975	7/9/2000	
4130	92140	0	Austria	German	M	1973	6/25/2000	
4131	80281	0	Austria	German	M	1970	6/13/2000	
4132	74438	0	Austria	German	M	1975	6/9/2000	
4133	36822	0	Austria	German	M	1970	3/20/2000	

4134 rows × 8 columns

In [4]:
```
daily_agg_df
```

Out[4]:

	UserID	Date	ProductType	Turnover	Hold	NumberofBets
0	31965	5/8/2000	1	15.3388	15.3388	1
1	31965	5/10/2000	1	34.1594	34.1594	5
2	31965	5/18/2000	1	24.5419	24.5419	4
3	31965	5/22/2000	1	2.5309	2.5309	1
4	31965	5/23/2000	1	15.3387	15.3387	2
...	...	...	...	...	...	...
981777	9200696	10/12/2010	25			95
981778	7912483	9/23/2010	25			60
981779	9200696	10/11/2010	25			2
981780	9200696	10/14/2010	25			2
981781	9200696	10/24/2010	25			2

981782 rows × 6 columns

Data Transformation and Cleaning

In [5]:
```
import pandas as pd

# Define a standard date for filling empty and invalid cells
standard_date = pd.to_datetime('01/01/1900', format='%d/%m/%Y', errors='coerce')
```

```
# Fill empty and invalid cells with the standard date
daily_agg_df['Date'] = pd.to_datetime(daily_agg_df['Date'], errors='coerce').fillna(st
demog_df['Registration_date'] = pd.to_datetime(demog_df['Registration_date'], errors='
demog_df['First_Deposit_Date'] = pd.to_datetime(demog_df['First_Deposit_Date'], errors
rg_det_df['RGFirst_Date'] = pd.to_datetime(rg_det_df['RGFirst_Date'], errors='coerce')
rg_det_df['RGLast_date'] = pd.to_datetime(rg_det_df['RGLast_date'], errors='coerce').f

# Create new datetime columns
daily_agg_df['Aggregate_Date'] = pd.to_datetime(daily_agg_df['Date'])

daily_agg_df.drop('Date', axis=1, inplace=True)

demog_df['Registration_date'] = pd.to_datetime(demog_df['Registration_date'])
demog_df['First_Deposit_Date'] = pd.to_datetime(demog_df['First_Deposit_Date'])
rg_det_df['RGFirst_Date'] = pd.to_datetime(rg_det_df['RGFirst_Date'])
rg_det_df['RGLast_date'] = pd.to_datetime(rg_det_df['RGLast_date'])

# Rename the 'old_column_name' to 'new_column_name'
daily_agg_df = daily_agg_df.rename(columns={'UserID': 'UserID'})
demog_df = demog_df.rename(columns={'USERID': 'UserID'})
rg_det_df = rg_det_df.rename(columns={'UserID': 'UserID'})
```

In [6]:
```
daily_agg_df
product_type_frequencies = daily_agg_df['ProductType'].value_counts()
print(product_type_frequencies)
```

```
1     399410
2     331828
10    127223
8      37749
15     25646
4      20749
6      13558
3       7539
14      7310
19      6122
7       1741
23      1215
5        559
17       506
20       321
22       158
9         67
21        38
24        35
25         7
16         1
Name: ProductType, dtype: int64
```

In [7]:
```
daily_agg_df_t=daily_agg_df.tail(10)
```

In [8]:
```
daily_agg_df_t
```

Out[8]:

	UserID	ProductType	Turnover	Hold	NumberofBets	Aggregate_Date
981772	4608302	24			6	2010-09-17
981773	1285995	24			2	2010-07-27
981774	4608302	24			3	2010-09-16
981775	7912483	25			393	2010-09-24
981776	7912483	25			228	2010-09-22
981777	9200696	25			95	2010-10-12
981778	7912483	25			60	2010-09-23
981779	9200696	25			2	2010-10-11
981780	9200696	25			2	2010-10-14
981781	9200696	25			2	2010-10-24

In [9]:
```
demog_df
```

Out[9]:

	UserID	RG_case	CountryName	LanguageName	Gender	YearofBirth	Registration_date	First_
0	2975944	1	Germany	German	M	1970	2006-06-02	
1	9822065	1	Germany.COM	German	F	1963	2009-11-21	
2	9622454	1	France.COM	French	F	1981	2009-10-19	
3	9619356	1	Italy.IT	Italian	F	1975	2009-10-18	
4	9593498	1	Germany.COM	German	F	1990	2009-10-14	
...	...	...	...	...	...	...	...	
4129	107292	0	Austria	German	M	1975	2000-07-09	
4130	92140	0	Austria	German	M	1973	2000-06-25	
4131	80281	0	Austria	German	M	1970	2000-06-13	
4132	74438	0	Austria	German	M	1975	2000-06-09	
4133	36822	0	Austria	German	M	1970	2000-03-20	

4134 rows × 8 columns

In [10]:
```
rg_det_df
```

Out[10]:

	UserID	RGsumevents	RGFirst_Date	RGLast_date	Event_type_first	Interventiontype_first
0	2169867	1	2009-11-19	2009-11-19	9	18
1	7035862	1	2009-11-15	2009-11-15	9	18
2	5911218	1	2009-11-08	2009-11-08	9	18
3	5872708	1	2009-11-03	2009-11-03	9	18
4	5746942	2	2009-10-18	2009-11-03	9	18
...	...	...	...	...	...	...
2063	2451840	5	1900-01-01	2009-11-03	12	
2064	9140426	1	2009-08-19	2009-08-19	10	
2065	2590026	2	2009-08-05	2009-08-06	10	
2066	1023918	1	2009-04-09	2009-04-09	6	
2067	6691324	1	2008-12-19	2008-12-19	1	

2068 rows × 6 columns

In [11]:
```
from sklearn.preprocessing import LabelEncoder

# Initialize label encoders
label_encoder_country = LabelEncoder()
label_encoder_language = LabelEncoder()
label_encoder_gender = LabelEncoder()

# Fit and transform the categorical columns
demog_df['CountryName'] = label_encoder_country.fit_transform(demog_df['CountryName'])
demog_df['LanguageName'] = label_encoder_language.fit_transform(demog_df['LanguageName
demog_df['Gender'] = label_encoder_gender.fit_transform(demog_df['Gender'])
```

In [12]:
```
demog_df
```

Out[12]:

	UserID	RG_case	CountryName	LanguageName	Gender	YearofBirth	Registration_date	First_
0	2975944	1	18	8	1	1970	2006-06-02	
1	9822065	1	19	8	0	1963	2009-11-21	
2	9622454	1	17	7	0	1981	2009-10-19	
3	9619356	1	25	11	0	1975	2009-10-18	
4	9593498	1	19	8	0	1990	2009-10-14	
...	...	...	...	...	...	...	...	
4129	107292	0	4	8	1	1975	2000-07-09	
4130	92140	0	4	8	1	1973	2000-06-25	
4131	80281	0	4	8	1	1970	2000-06-13	
4132	74438	0	4	8	1	1975	2000-06-09	
4133	36822	0	4	8	1	1970	2000-03-20	

4134 rows × 8 columns

Merging of Datasets

In [13]:
```
merged_df = daily_agg_df.merge(demog_df, on='UserID', how='outer')
merged_df = merged_df.merge(rg_det_df, on='UserID', how='outer')
```

In [14]:
```
merged_df['RG_case'].value_counts()
```

Out[14]:
```
1    811570
0    170233
Name: RG_case, dtype: int64
```

In [15]:
```
merged_df
```

Out[15]:

	UserID	ProductType	Turnover	Hold	NumberofBets	Aggregate_Date	RG_case	CountryI
0	31965	1.0	15.3388	15.3388	1	2000-05-08	1	
1	31965	1.0	34.1594	34.1594	5	2000-05-10	1	
2	31965	1.0	24.5419	24.5419	4	2000-05-18	1	
3	31965	1.0	2.5309	2.5309	1	2000-05-22	1	
4	31965	1.0	15.3387	15.3387	2	2000-05-23	1	
...	...	...	...	...	...	...	...	...
981798	1190813	NaN	NaN	NaN	NaN	NaT	0	
981799	1622440	NaN	NaN	NaN	NaN	NaT	0	
981800	1108530	NaN	NaN	NaN	NaN	NaT	0	
981801	683142	NaN	NaN	NaN	NaN	NaT	0	
981802	113041	NaN	NaN	NaN	NaN	NaT	0	

981803 rows × 18 columns

In [16]:
```
merged_df_tsc=merged_df.drop(columns=['RGLast_date','RGFirst_Date','Registration_date'
merged_df_tsc=merged_df_tsc #.tail(13000)
filtered_df = merged_df_tsc[merged_df_tsc['ProductType'] == 2]
filtered_df
# Sort the DataFrame by 'user_id' and 'date'
filtered_df = filtered_df.sort_values(by=['UserID', 'Aggregate_Date'])
```

In [17]:
```
filtered_df['RG_case'].value_counts()
```

Out[17]:
```
1    294375
0     37453
Name: RG_case, dtype: int64
```

In [18]:
```
subset_columns = ['Aggregate_Date', 'UserID']

# Identify and drop duplicate rows based on the specified subset of columns
filtered_df = filtered_df.drop_duplicates(subset=subset_columns, keep='first')

# Display the DataFrame after dropping duplicates
print("DataFrame after dropping duplicates:")
print(filtered_df)
```

```
DataFrame after dropping duplicates:
         UserID  ProductType Turnover   Hold NumberofBets Aggregate_Date  \
402       31965          2.0     20.0   20.0            1     2002-11-12
1002      31965          2.0    73.18  18.96            6     2002-11-14
403       31965          2.0     10.0   10.0            1     2002-11-15
1217      31965          2.0   163.28   19.0            9     2002-11-16
1331      31965          2.0   162.34  -97.5           13     2002-11-17
...         ...          ...      ...    ...          ...            ...
973451  9822065          2.0     10.0   10.0            1     2010-02-09
973463  9822065          2.0     15.0  -3.35            1     2010-03-02
973452  9822065          2.0      1.0    1.0            1     2010-04-21
973456  9822065          2.0      1.0   -2.8            1     2010-06-10
973516  9859152          2.0     13.0   13.0            5     2009-11-27

        RG_case  CountryName  LanguageName  Gender YearofBirth  RGsumevents  \
402           1           19             8       1        1971          1.0
1002          1           19             8       1        1971          1.0
403           1           19             8       1        1971          1.0
1217          1           19             8       1        1971          1.0
1331          1           19             8       1        1971          1.0
...         ...          ...           ...     ...         ...          ...
973451        1           19             8       0        1963          1.0
973463        1           19             8       0        1963          1.0
973452        1           19             8       0        1963          1.0
973456        1           19             8       0        1963          1.0
973516        0           19             8       1        1982          NaN

        Event_type_first Interventiontype_first
402                  2.0                     8
1002                 2.0                     8
403                  2.0                     8
1217                 2.0                     8
1331                 2.0                     8
...                  ...                   ...
973451               4.0                    13
973463               4.0                    13
973452               4.0                    13
973456               4.0                    13
973516               NaN                   NaN

[322898 rows x 14 columns]
```

In [19]:

```python
filtered_df['RG_case'].value_counts()

null_mask = filtered_df.isna()

# Use sum() to count the null values in each column
null_count = null_mask.sum()

# Display columns with null values and their respective counts
print("Columns with null values and their counts:")
print(null_count[null_count > 0])
filtered_df

filtered_df = filtered_df.fillna(0)
```

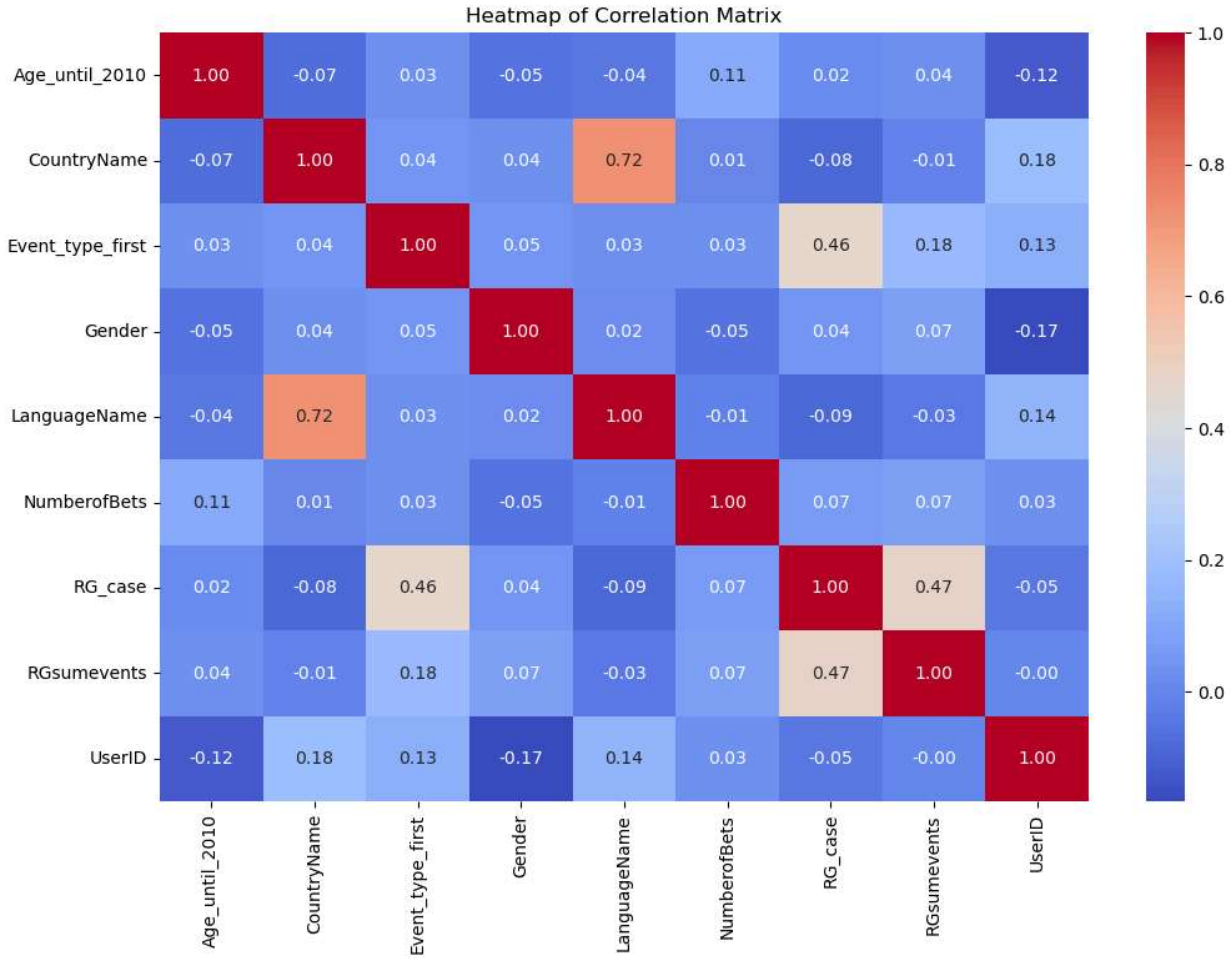
```
Columns with null values and their counts:
RGsumevents               36498
Event_type_first          36498
Interventiontype_first    36498
dtype: int64
```

In [20]:
```
filtered_df
# Convert birth year to age until 2010
current_year = 2010
filtered_df['Age_until_2010'] = filtered_df['YearofBirth'].apply(lambda birth_year: cu
```

Feature Selection

In [21]:
```
columns_for_heatmap = filtered_df.columns.difference(['ProductType'])

# Create a subset DataFrame with selected columns
heatmap_data = filtered_df[columns_for_heatmap]

# Create a heatmap using seaborn
plt.figure(figsize=(12, 8))
sns.heatmap(heatmap_data.corr(), annot=True, cmap='coolwarm', fmt=".2f")
plt.title('Heatmap of Correlation Matrix')
plt.show()
```

```
C:\Users\abhiv\AppData\Local\Temp\ipykernel_14916\2072689473.py:8: FutureWarning: The
default value of numeric_only in DataFrame.corr is deprecated. In a future version, i
t will default to False. Select only valid columns or specify the value of numeric_on
ly to silence this warning.
  sns.heatmap(heatmap_data.corr(), annot=True, cmap='coolwarm', fmt=".2f")
```



In [22]:
```
filtered_df
```

Out[22]:

	UserID	ProductType	Turnover	Hold	NumberofBets	Aggregate_Date	RG_case	CountryNa
402	31965	2.0	20.0	20.0	1	2002-11-12	1	
1002	31965	2.0	73.18	18.96	6	2002-11-14	1	
403	31965	2.0	10.0	10.0	1	2002-11-15	1	
1217	31965	2.0	163.28	19.0	9	2002-11-16	1	
1331	31965	2.0	162.34	-97.5	13	2002-11-17	1	
...	...	...	...	...	...	...	...	...
973451	9822065	2.0	10.0	10.0	1	2010-02-09	1	
973463	9822065	2.0	15.0	-3.35	1	2010-03-02	1	
973452	9822065	2.0	1.0	1.0	1	2010-04-21	1	
973456	9822065	2.0	1.0	-2.8	1	2010-06-10	1	
973516	9859152	2.0	13.0	13.0	5	2009-11-27	0	

322898 rows × 15 columns

Model Fitting

K - Means Clustering

In [23]:
```
import numpy as np
import pandas as pd
from sklearn.cluster import KMeans
from sklearn.decomposition import PCA
import matplotlib.pyplot as plt


# Pivot the DataFrame to create a 3D array with entries as rows, features as columns,
user_data_3d = filtered_df.pivot(index='UserID', columns='Aggregate_Date', values=['Tu

# Fill missing values with zeros (if any)
user_data_3d = user_data_3d.fillna(0)

# Convert empty strings (' ') to float values of zero (0)
user_data_3d = user_data_3d.replace(' ', 0).astype(float)
user_data_3d = user_data_3d.astype(float)

# Convert the pivot table to a NumPy array
user_data_array = user_data_3d.to_numpy()

#k fold value
k = 3

# Perform K-means clustering
model = KMeans(n_clusters=k, random_state=0)
y_pred = model.fit_predict(user_data_array)

# Apply PCA to reduce dimensionality to 2D
pca = PCA(n_components=2)
user_data_pca = pca.fit_transform(user_data_array)

# Visualize the clustered entries using PCA components
plt.figure(figsize=(8, 6))
scatter = plt.scatter(user_data_pca[:, 0], user_data_pca[:, 1], c=y_pred, cmap='viridi
plt.xlabel('PCA Component 1')
plt.ylabel('PCA Component 2')




cluster_dict={}

# Create a dictionary to store user IDs and cluster number
cluster_dict = {i: {'users': [], 'cluster_num': i} for i in range(k)}

# Loop through the cluster labels and append user IDs to the corresponding cluster
for user_id, cluster_label in zip(user_data_3d.index, y_pred):
    cluster_dict[cluster_label]['users'].append(user_id)

# Print the user IDs and cluster number in each cluster
for cluster_label, info in cluster_dict.items():
```

```
    print(f'Cluster {cluster_label} (Cluster {info["cluster_num"]}):\nUsers: {info["us


# Create a legend for the cluster numbers
legend_labels = ['Moderate Problem Gamblers','Early Players','Problem Gamblers']
legend = plt.legend(handles=scatter.legend_elements()[0], title='Cluster', labels=lege

plt.gca().add_artist(legend)

plt.show()
```

```
C:\Users\abhiv\AppData\Local\anaconda3\Lib\site-packages\sklearn\cluster\_kmeans.py:8
70: FutureWarning: The default value of `n_init` will change from 10 to 'auto' in 1.
4. Set the value of `n_init` explicitly to suppress the warning
  warnings.warn(
```

```
Cluster 0 (Cluster 0):
Users: [868583, 1175809, 1411743, 1457496, 1486136, 1662632, 1679490, 1776178, 179084
8, 1921204, 2070894, 2150296, 2155065, 2589710, 2704382, 2852203, 3669466, 3852889, 3
904422, 3968386, 4006708, 4371320, 4394754, 4412550, 4495603, 4532357, 5106620, 53082
71, 5488160, 5660719, 5678852, 5723033, 6158120, 6175402, 6239380, 6283338, 6709379,
6985339, 7192925]
Cluster 1 (Cluster 1):
Users: [31965, 32639, 36822, 36916, 74438, 90746, 91707, 92140, 96950, 99596, 100167,
107124, 107292, 109445, 113136, 119406, 121115, 134759, 157710, 165200, 165335, 16614
3, 178914, 187808, 191415, 212099, 217121, 221079, 233644, 275466, 306915, 332091, 34
8381, 349493, 349855, 350228, 355882, 366074, 375779, 376200, 377204, 378707, 380006,
380781, 386047, 399987, 402872, 403944, 405981, 411101, 420681, 441202, 452829, 45576
6, 464507, 465623, 467890, 468786, 469615, 470323, 471813, 484593, 485942, 486333, 48
6555, 486829, 502856, 513616, 518169, 519601, 528696, 535025, 537380, 539496, 548298,
551872, 555911, 562883, 572029, 576509, 578303, 578678, 579708, 581568, 596939, 60006
2, 601929, 606177, 606833, 607958, 610512, 610990, 612580, 614867, 615124, 617512, 61
7643, 618006, 618134, 622333, 623222, 624243, 626809, 626818, 626970, 627380, 627907,
628673, 636671, 637178, 637789, 637922, 638301, 640126, 640946, 644434, 646063, 64927
4, 651518, 651781, 652803, 654065, 654455, 656052, 659457, 660104, 662035, 668204, 66
8658, 669331, 669759, 671003, 671210, 671657, 671821, 676437, 676439, 679909, 689277,
691215, 691219, 691300, 693201, 695820, 704670, 707265, 708122, 709000, 709426, 70956
6, 712428, 713486, 714120, 715823, 716725, 717922, 719989, 721385, 722705, 724802, 72
5520, 727358, 731018, 734038, 734064, 735381, 740719, 758982, 759655, 767729, 779314,
779784, 781520, 793809, 809647, 815193, 816423, 819431, 819712, 820287, 821915, 82392
7, 829363, 832139, 836619, 838424, 839105, 841112, 842884, 845248, 847728, 852596, 85
9526, 860020, 860767, 863069, 872943, 872995, 876092, 876669, 876762, 876836, 877236,
878190, 879490, 881586, 891132, 893392, 893855, 894982, 897560, 905798, 908460, 91011
3, 910143, 912480, 912981, 921177, 921652, 934074, 939165, 939794, 940344, 943528, 94
5804, 946810, 951260, 954936, 954947, 958509, 960009, 962369, 963330, 965698, 969529,
971617, 978164, 984545, 984819, 986088, 986289, 991634, 992202, 994151, 998680, 99901
8, 1002807, 1003282, 1006572, 1006668, 1015042, 1019680, 1023918, 1029762, 1033839, 1
034626, 1038912, 1039083, 1052452, 1053169, 1054028, 1055097, 1057341, 1059479, 10659
10, 1066411, 1066905, 1068147, 1068353, 1068512, 1068666, 1069445, 1069480, 1069718,
1073054, 1073059, 1076465, 1077256, 1086185, 1086610, 1086719, 1089300, 1092942, 1097
302, 1098879, 1098995, 1099733, 1100329, 1102048, 1102347, 1104183, 1104232, 1105726,
1108323, 1108391, 1108768, 1110796, 1110869, 1112009, 1112125, 1114773, 1119668, 1119
951, 1121270, 1121292, 1123515, 1124321, 1124617, 1125602, 1128246, 1128286, 1128557,
1129130, 1131579, 1133624, 1134790, 1137771, 1139267, 1143257, 1145088, 1152572, 1153
292, 1161759, 1166481, 1168835, 1169296, 1171028, 1174435, 1177014, 1177425, 1178665,
1179202, 1180532, 1184758, 1186045, 1188402, 1188702, 1188944, 1188981, 1190225, 1190
678, 1190696, 1190818, 1190880, 1191041, 1191110, 1191496, 1191525, 1191883, 1195259,
1195708, 1198551, 1198657, 1200490, 1202145, 1205205, 1207102, 1208186, 1210080, 1211
034, 1211153, 1215396, 1216806, 1216960, 1217056, 1217347, 1217527, 1217606, 1218000,
1220034, 1221797, 1222876, 1226174, 1226493, 1226683, 1227473, 1227499, 1229500, 1230
025, 1230735, 1236874, 1247913, 1251523, 1260516, 1260523, 1261748, 1261993, 1262533,
1263607, 1265820, 1267246, 1268833, 1269295, 1269446, 1269588, 1273875, 1275992, 1276
276, 1276696, 1277104, 1277167, 1277939, 1278165, 1280420, 1280671, 1281272, 1282148,
1282506, 1284616, 1285413, 1285959, 1285995, 1286146, 1286507, 1288342, 1289020, 1289
575, 1290126, 1290225, 1290981, 1292031, 1292371, 1292648, 1295199, 1298752, 1298980,
1299294, 1299793, 1301234, 1301389, 1304180, 1305631, 1305661, 1306998, 1307000, 1307
421, 1307533, 1309623, 1311762, 1315386, 1316055, 1316554, 1317059, 1318469, 1320903,
1321102, 1323207, 1325161, 1325647, 1325682, 1329184, 1330470, 1333978, 1334455, 1336
326, 1336704, 1337767, 1339503, 1342111, 1342426, 1343436, 1344751, 1344830, 1345978,
1348134, 1348150, 1348549, 1353206, 1356035, 1357532, 1361415, 1362339, 1362354, 1362
573, 1363362, 1363440, 1364127, 1364138, 1365118, 1365304, 1367719, 1371692, 1374073,
1374185, 1375149, 1376027, 1376313, 1376503, 1377425, 1377479, 1381040, 1383267, 1387
059, 1387925, 1388917, 1390642, 1393636, 1394702, 1397690, 1398079, 1398948, 1402673,
1402736, 1403471, 1406089, 1406743, 1407020, 1407759, 1411882, 1412153, 1412349, 1415
558, 1416109, 1418087, 1420230, 1422024, 1423249, 1425148, 1426789, 1426870, 1427115,
1427840, 1428100, 1428339, 1430272, 1430651, 1431358, 1436194, 1444002, 1444024, 1444
```

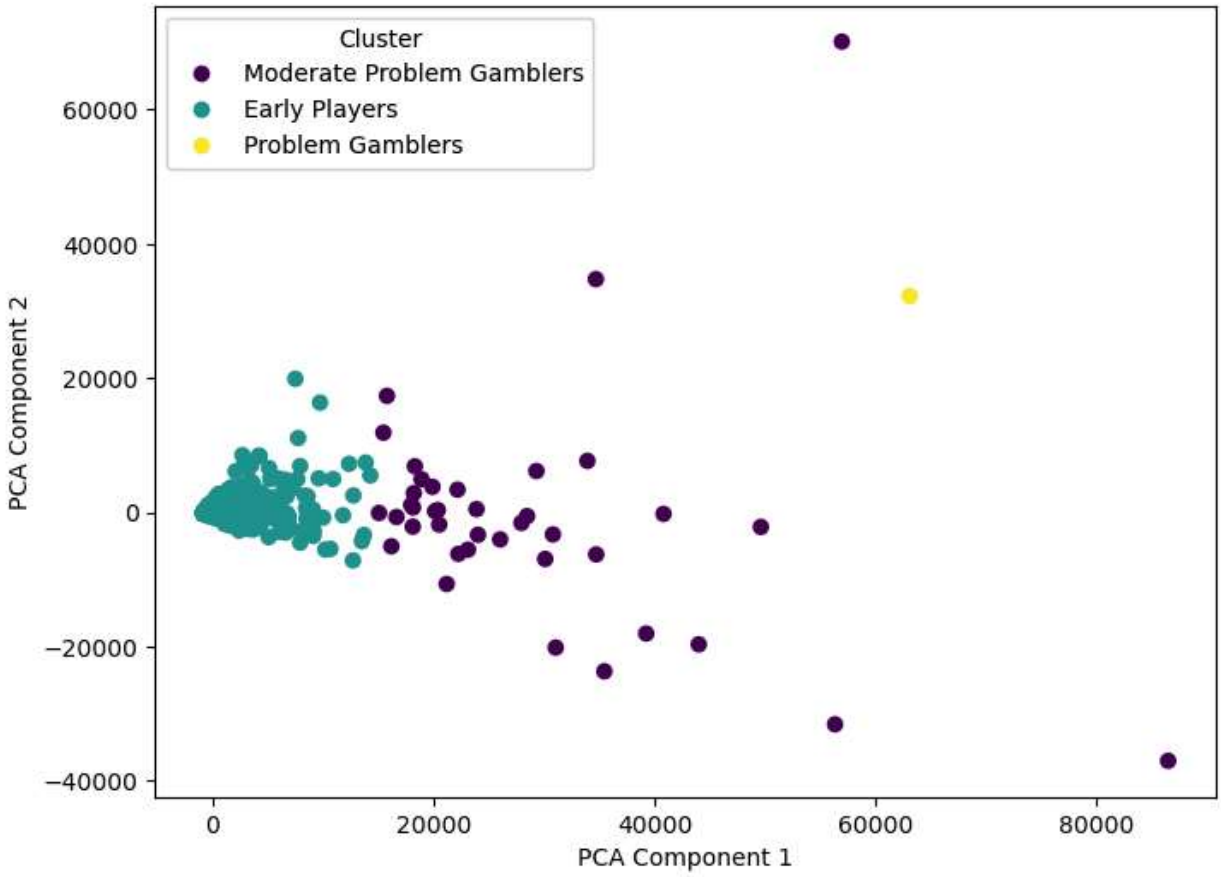
638, 1444690, 1446765, 1446979, 1447394, 1448747, 1448755, 1448966, 1448980, 1449058,
1453530, 1455429, 1455607, 1458694, 1460790, 1461123, 1466020, 1467698, 1467720, 1468
506, 1474080, 1475521, 1475836, 1476237, 1476815, 1477587, 1481182, 1482265, 1487056,
1487262, 1489176, 1494069, 1495100, 1499093, 1504089, 1504941, 1511327, 1511559, 1511
812, 1516266, 1516433, 1516508, 1519947, 1521035, 1521486, 1521811, 1522428, 1522618,
1524410, 1524893, 1527006, 1528877, 1529711, 1531097, 1533580, 1533943, 1538409, 1539
163, 1539371, 1540796, 1541252, 1541799, 1545811, 1546185, 1548350, 1548547, 1549759,
1551158, 1556539, 1556993, 1559015, 1560024, 1560193, 1561422, 1564451, 1567292, 1569
967, 1572636, 1573032, 1575816, 1578866, 1579804, 1586572, 1587474, 1589646, 1589909,
1590508, 1590556, 1592977, 1594233, 1594288, 1599117, 1600042, 1601073, 1601397, 1602
071, 1603442, 1604150, 1606117, 1609008, 1610238, 1610754, 1613141, 1613727, 1614409,
1615173, 1618935, 1624023, 1626239, 1630307, 1630590, 1645647, 1646641, 1648376, 1649
239, 1651030, 1651912, 1652164, 1659620, 1663129, 1664927, 1666227, 1669102, 1669496,
1671815, 1677908, 1680006, 1680615, 1683379, 1687563, 1689908, 1690828, 1691317, 1692
544, 1693518, 1696904, 1699984, 1700174, 1711860, 1713213, 1715092, 1717071, 1717825,
1720080, 1723411, 1725647, 1729399, 1730948, 1731044, 1735247, 1738054, 1738946, 1739
322, 1740517, 1740827, 1741128, 1741291, 1742295, 1745411, 1745784, 1747125, 1751937,
1752640, 1753291, 1755673, 1759481, 1759742, 1761005, 1763045, 1764945, 1766119, 1766
299, 1766392, 1768666, 1770262, 1775842, 1776268, 1776887, 1776969, 1777482, 1780205,
1782215, 1782457, 1785350, 1788435, 1791179, 1791277, 1791309, 1792037, 1792412, 1794
523, 1800624, 1803500, 1804280, 1804765, 1805683, 1806096, 1812446, 1815916, 1817630,
1819339, 1819450, 1820063, 1821069, 1823158, 1824860, 1825093, 1825523, 1828196, 1829
131, 1829513, 1830174, 1832172, 1834053, 1834871, 1835132, 1835689, 1837597, 1838010,
1838269, 1838486, 1839073, 1839746, 1842142, 1845693, 1846776, 1851270, 1852623, 1853
602, 1855060, 1862422, 1862652, 1864071, 1865113, 1865291, 1865799, 1866579, 1866785,
1867152, 1869162, 1869318, 1870184, 1873384, 1877206, 1878774, 1879399, 1881868, 1882
402, 1886970, 1887198, 1888342, 1892312, 1894098, 1896830, 1897984, 1898064, 1900947,
1903663, 1908533, 1916328, 1918491, 1919907, 1922517, 1922874, 1926155, 1928220, 1929
576, 1930053, 1940754, 1946527, 1951934, 1951951, 1953416, 1954512, 1960205, 1961268,
1965428, 1966235, 1967044, 1969410, 1970026, 1979513, 1980711, 1982649, 1982722, 1983
077, 1983307, 1983615, 1984245, 1986069, 1986177, 1988710, 1989952, 1991320, 1995745,
1996350, 1996750, 2015501, 2015746, 2017170, 2018397, 2024125, 2025088, 2025552, 2027
653, 2031783, 2032749, 2035245, 2036135, 2037459, 2039222, 2044508, 2048423, 2049556,
2051142, 2053042, 2059854, 2060010, 2061805, 2062223, 2063492, 2064756, 2064874, 2067
210, 2067768, 2071786, 2074932, 2077486, 2080044, 2080948, 2082726, 2083227, 2085469,
2088451, 2091973, 2097334, 2101393, 2107517, 2108076, 2109792, 2109844, 2113806, 2114
144, 2115475, 2115515, 2115661, 2115829, 2115885, 2117010, 2117792, 2117915, 2118301,
2118401, 2118768, 2118821, 2121263, 2121662, 2128299, 2128610, 2130822, 2130868, 2130
908, 2131491, 2132666, 2132724, 2137863, 2139355, 2141713, 2141867, 2143548, 2143609,
2143770, 2145324, 2146460, 2154785, 2169867, 2171273, 2173193, 2173859, 2177863, 2179
252, 2180684, 2182157, 2182953, 2183448, 2183730, 2187527, 2188068, 2190858, 2196540,
2196668, 2209271, 2215644, 2217800, 2226043, 2229447, 2229692, 2229823, 2230789, 2231
307, 2236967, 2239045, 2245377, 2248380, 2251748, 2251981, 2253237, 2254195, 2258689,
2260208, 2262246, 2263489, 2265146, 2265283, 2267078, 2267659, 2281766, 2284240, 2288
380, 2292452, 2293422, 2294237, 2295217, 2296295, 2299626, 2301362, 2302227, 2312117,
2312644, 2313963, 2314619, 2314911, 2315978, 2327108, 2335397, 2336892, 2337250, 2337
641, 2340708, 2344763, 2345086, 2346130, 2347760, 2348698, 2354456, 2355664, 2367866,
2368879, 2369090, 2370401, 2370763, 2378672, 2387521, 2387631, 2388325, 2388884, 2389
448, 2389953, 2392977, 2394146, 2394176, 2395168, 2398820, 2401635, 2402783, 2403003,
2403964, 2404466, 2405222, 2407845, 2410472, 2410487, 2415497, 2418874, 2421685, 2421
731, 2426061, 2426245, 2426472, 2437022, 2440555, 2441127, 2442024, 2444971, 2446847,
2451840, 2453026, 2454635, 2455322, 2460804, 2463390, 2479411, 2480120, 2483161, 2486
108, 2486182, 2488952, 2493621, 2494719, 2497340, 2499607, 2504528, 2509218, 2510850,
2512417, 2512616, 2512671, 2512799, 2513495, 2514092, 2514158, 2514186, 2514226, 2514
412, 2514492, 2515376, 2515716, 2515780, 2517866, 2518485, 2522297, 2530627, 2544602,
2546637, 2547501, 2553963, 2553978, 2554254, 2559605, 2560163, 2563714, 2566614, 2568
868, 2570969, 2575293, 2578415, 2578609, 2580478, 2581124, 2590026, 2595325, 2602629,
2606689, 2607401, 2609261, 2610717, 2611270, 2612177, 2613968, 2613991, 2614721, 2617
334, 2619667, 2628935, 2630099, 2631482, 2631959, 2633677, 2637226, 2637579, 2637689,
2637920, 2640595, 2644036, 2645819, 2652607, 2652627, 2653065, 2657008, 2657435, 2658

193, 2659713, 2662389, 2668544, 2669121, 2670228, 2670782, 2671082, 2672817, 2673854,
2674431, 2674913, 2675400, 2676078, 2676974, 2676986, 2678522, 2685297, 2686509, 2689
756, 2691752, 2691847, 2696359, 2697251, 2700081, 2700940, 2710254, 2713872, 2713897,
2715739, 2716478, 2718387, 2724141, 2724908, 2726040, 2739568, 2742218, 2747243, 2747
359, 2756947, 2757089, 2759033, 2760534, 2760998, 2764251, 2765830, 2769016, 2775461,
2776967, 2777944, 2780351, 2780775, 2781022, 2782230, 2783309, 2785971, 2787151, 2788
362, 2789254, 2790387, 2790410, 2791284, 2794275, 2795050, 2798232, 2798379, 2799291,
2801071, 2803911, 2804406, 2808818, 2811791, 2816866, 2818081, 2820596, 2824087, 2824
386, 2824516, 2824913, 2825283, 2825290, 2829998, 2832999, 2835565, 2836222, 2838112,
2840190, 2841492, 2846231, 2846850, 2850549, 2850575, 2858490, 2859017, 2865204, 2865
324, 2868064, 2870233, 2871123, 2873035, 2874213, 2874823, 2875900, 2878772, 2888533,
2896136, 2903523, 2905881, 2907188, 2907298, 2909924, 2912319, 2923335, 2925134, 2927
797, 2928939, 2929192, 2930403, 2936645, 2940631, 2947836, 2948495, 2950279, 2952717,
2952806, 2956360, 2959362, 2962326, 2963327, 2970714, 2973592, 2975944, 2976931, 2981
530, 2982680, 2983951, 2985514, 2990875, 2991786, 2999195, 2999207, 3001494, 3001919,
3007415, 3007767, 3020285, 3032087, 3033642, 3041605, 3043690, 3047035, 3047650, 3056
908, 3057677, 3063690, 3067154, 3068010, 3075507, 3075642, 3079149, 3081335, 3089826,
3092883, 3097696, 3101778, 3105888, 3108964, 3113595, 3113982, 3115642, 3119828, 3119
956, 3123449, 3125143, 3127806, 3137317, 3137602, 3139734, 3141820, 3142810, 3145000,
3153074, 3153898, 3154500, 3158143, 3159911, 3161583, 3166557, 3168337, 3170006, 3175
332, 3176106, 3176447, 3177629, 3186379, 3189501, 3191576, 3193571, 3195985, 3201228,
3201331, 3204628, 3205721, 3208357, 3211029, 3220319, 3222322, 3222999, 3228531, 3229
927, 3239668, 3240092, 3243610, 3262791, 3276027, 3279457, 3283965, 3284620, 3293643,
3294817, 3294931, 3297399, 3300769, 3302342, 3306270, 3310069, 3315310, 3316099, 3321
251, 3324091, 3333180, 3334747, 3335713, 3335950, 3337667, 3344255, 3344391, 3356540,
3359885, 3361967, 3363267, 3363556, 3374574, 3378409, 3389381, 3412869, 3418212, 3418
331, 3419034, 3424027, 3426731, 3430600, 3431103, 3431662, 3438352, 3439198, 3443815,
3457670, 3458145, 3461699, 3464785, 3466439, 3471973, 3472036, 3472797, 3474092, 3474
653, 3479664, 3479874, 3482060, 3482637, 3485383, 3486964, 3492065, 3492657, 3494737,
3495839, 3496350, 3496778, 3499407, 3501612, 3509547, 3510474, 3512289, 3513036, 3514
222, 3516403, 3517053, 3517409, 3522326, 3522511, 3523355, 3523419, 3524836, 3531155,
3533664, 3534225, 3534409, 3537600, 3541124, 3542327, 3543480, 3548997, 3550011, 3551
338, 3551691, 3552893, 3553000, 3555377, 3555676, 3556967, 3559230, 3560408, 3566442,
3566841, 3567291, 3567414, 3567514, 3575629, 3578811, 3578832, 3579257, 3585210, 3593
395, 3598979, 3602619, 3604982, 3614179, 3616138, 3618875, 3621498, 3622203, 3627982,
3630907, 3636161, 3640058, 3641589, 3647824, 3648924, 3649908, 3658329, 3658346, 3658
458, 3663169, 3665686, 3667286, 3673262, 3680952, 3687077, 3687392, 3688036, 3690350,
3700434, 3703708, 3703970, 3704921, 3705154, 3705362, 3719998, 3722766, 3724040, 3724
936, 3725278, 3729425, 3730470, 3732545, 3734190, 3736747, 3738033, 3742803, 3744005,
3744808, 3749821, 3756964, 3758200, 3758557, 3759236, 3759377, 3762316, 3762986, 3767
300, 3775532, 3776825, 3777776, 3780381, 3781288, 3783409, 3787335, 3787839, 3789290,
3789713, 3790516, 3795516, 3795547, 3796332, 3797443, 3803154, 3803789, 3803808, 3804
873, 3804929, 3805497, 3805977, 3810030, 3816363, 3818425, 3820671, 3830788, 3831718,
3832102, 3833071, 3833762, 3834246, 3835026, 3835056, 3835611, 3835801, 3837155, 3839
573, 3841216, 3842441, 3845115, 3845339, 3847660, 3858441, 3858830, 3859961, 3866642,
3878333, 3879168, 3880140, 3898314, 3900386, 3904166, 3905927, 3910295, 3914576, 3914
709, 3916577, 3918664, 3920385, 3928266, 3929060, 3935764, 3941874, 3942389, 3942982,
3949330, 3953802, 3957159, 3964007, 3967404, 3970169, 3970969, 3976253, 3977539, 3980
898, 3985099, 3987134, 3990787, 3990865, 3990915, 3991188, 3992020, 3992162, 4002953,
4003852, 4008596, 4010854, 4014604, 4017465, 4022654, 4023500, 4027773, 4029771, 4035
471, 4039020, 4039888, 4046258, 4051960, 4060630, 4062082, 4063603, 4066997, 4070137,
4070702, 4079142, 4083316, 4086589, 4100320, 4101116, 4105251, 4107386, 4108321, 4111
713, 4111916, 4112499, 4115065, 4124215, 4124780, 4127435, 4132830, 4136167, 4136365,
4137676, 4147552, 4155348, 4157690, 4159330, 4161293, 4162915, 4163814, 4165436, 4165
995, 4172003, 4174765, 4176210, 4182510, 4187911, 4187994, 4189245, 4191011, 4202311,
4208451, 4211279, 4212062, 4212210, 4213940, 4214808, 4218661, 4220193, 4223135, 4224
792, 4227662, 4227692, 4227877, 4233000, 4235440, 4235877, 4236352, 4239077, 4240399,
4240601, 4241294, 4251068, 4252031, 4252539, 4255332, 4255946, 4256504, 4258218, 4258
756, 4259441, 4263629, 4267114, 4267728, 4268490, 4268949, 4271790, 4272209, 4273613,
4273908, 4278913, 4283381, 4300750, 4301766, 4301894, 4304688, 4305050, 4307903, 4315

528, 4316134, 4318071, 4320785, 4321851, 4324456, 4324910, 4326263, 4326454, 4332470,
4338065, 4339987, 4342173, 4343190, 4346578, 4348335, 4359190, 4359774, 4361017, 4361
395, 4364542, 4364693, 4365898, 4367159, 4368709, 4381619, 4385825, 4387346, 4388436,
4391886, 4398918, 4405909, 4407380, 4411724, 4412962, 4413987, 4423638, 4425784, 4427
998, 4432017, 4432630, 4437506, 4439493, 4439804, 4444055, 4444971, 4445665, 4449772,
4455380, 4456261, 4461152, 4461204, 4475391, 4477271, 4482362, 4485120, 4485847, 4486
087, 4487329, 4488267, 4491415, 4498578, 4501072, 4503235, 4503325, 4504325, 4504778,
4514270, 4520504, 4521895, 4523711, 4526190, 4527767, 4530023, 4531943, 4534916, 4541
490, 4543142, 4544282, 4549204, 4550348, 4556197, 4558728, 4564194, 4564933, 4571319,
4572043, 4573657, 4578985, 4579060, 4579343, 4579575, 4580838, 4585066, 4585711, 4587
239, 4588003, 4589603, 4593718, 4594622, 4595030, 4597571, 4599005, 4599145, 4600997,
4602219, 4607711, 4607869, 4608302, 4609592, 4610412, 4620291, 4632593, 4636089, 4639
641, 4642329, 4645875, 4647003, 4648529, 4653627, 4657171, 4659410, 4667915, 4675453,
4682533, 4683871, 4685809, 4686195, 4690079, 4692310, 4693383, 4693834, 4694966, 4695
190, 4697358, 4699296, 4700436, 4701220, 4701658, 4703930, 4706721, 4707465, 4708962,
4714797, 4717280, 4717806, 4718618, 4723797, 4723927, 4724247, 4724793, 4725377, 4725
440, 4726623, 4728720, 4733608, 4734610, 4738703, 4741334, 4745022, 4750760, 4754574,
4754903, 4764719, 4773672, 4775118, 4776390, 4778231, 4779119, 4780579, 4781184, 4782
932, 4785430, 4788017, 4789988, 4790949, 4792984, 4793332, 4793506, 4795594, 4796476,
4800972, 4803105, 4803731, 4805410, 4806066, 4808150, 4808882, 4810358, 4811008, 4812
384, 4812422, 4812626, 4813899, 4814986, 4816031, 4817579, 4821096, 4821963, 4837595,
4857484, 4858250, 4860191, 4863248, 4863608, 4863645, 4863920, 4864900, 4865685, 4866
582, 4867029, 4867936, 4872118, 4874828, 4882145, 4882865, 4886424, 4888078, 4888406,
4890437, 4893144, 4894302, 4896650, 4897013, 4901809, 4903969, 4907036, 4909756, 4909
878, 4911082, 4911519, 4912516, 4916704, 4917155, 4917464, 4920698, 4921336, 4921440,
4921552, 4924378, 4932146, 4936637, 4938386, 4939096, 4940917, 4944708, 4946392, 4963
002, 4970372, 4970800, 4972981, 4973642, 4978317, 4980745, 4986792, 4990500, 4994458,
4994531, 4995723, 4996838, 5005168, 5013004, 5020362, 5021938, 5022200, 5028144, 5032
434, 5034333, 5038360, 5041127, 5042290, 5042963, 5043396, 5044150, 5046922, 5049659,
5061924, 5070173, 5070706, 5070927, 5071236, 5071663, 5084881, 5086781, 5097506, 5100
350, 5101193, 5104588, 5105138, 5105551, 5106110, 5108327, 5112671, 5121779, 5121862,
5128030, 5133765, 5135681, 5141940, 5143155, 5147109, 5149047, 5151648, 5159352, 5160
540, 5160926, 5163709, 5164325, 5167895, 5171821, 5175043, 5175623, 5178004, 5178879,
5179708, 5183520, 5193178, 5194256, 5195578, 5196907, 5198289, 5199657, 5201159, 5204
669, 5205130, 5205182, 5206109, 5207909, 5210215, 5212669, 5215521, 5218179, 5221552,
5223699, 5223765, 5225846, 5226144, 5229055, 5233772, 5233791, 5240271, 5243531, 5248
725, 5250607, 5250620, 5250975, 5266116, 5275259, 5277765, 5278809, 5281639, 5282505,
5292112, 5293064, 5294697, 5296024, 5296865, 5299312, 5302724, 5303756, 5306717, 5308
420, 5312173, 5312401, 5312633, 5313336, 5313473, 5313713, 5317415, 5318433, 5320445,
5321798, 5326898, 5327108, 5327208, 5332448, 5343080, 5344620, 5345825, 5348916, 5349
782, 5351172, 5355263, 5372244, 5373487, 5374667, 5376596, 5376766, 5377519, 5385439,
5386219, 5389318, 5392477, 5392759, 5395841, 5396402, 5400125, 5403022, 5406999, 5410
743, 5410988, 5411459, 5411906, 5417704, 5419513, 5425402, 5426534, 5427456, 5428519,
5429528, 5434093, 5436914, 5438659, 5440247, 5444281, 5451167, 5457785, 5458401, 5461
975, 5469148, 5469767, 5470090, 5471987, 5472147, 5473118, 5474101, 5474804, 5478413,
5478522, 5479140, 5488730, 5490769, 5490778, 5494873, 5494987, 5499555, 5508645, 5508
867, 5515613, 5516276, 5519759, 5522420, 5524866, 5526456, 5531889, 5536600, 5545891,
5546056, 5546682, 5549292, 5549778, 5549911, 5554333, 5555972, 5557975, 5560128, 5566
333, 5569750, 5570612, 5571695, 5589113, 5591502, 5592984, 5594467, 5595566, 5596445,
5598776, 5599403, 5601601, 5601680, 5604041, 5604384, 5605210, 5610752, 5613794, 5617
435, 5617436, 5617934, 5618421, 5622189, 5630001, 5631328, 5638913, 5644752, 5653968,
5659426, 5659487, 5659562, 5661477, 5661598, 5669669, 5670707, 5673449, 5673729, 5678
031, 5679214, 5680468, 5684034, 5691406, 5691546, 5696556, 5698470, 5699878, 5709219,
5716084, 5716801, 5718960, 5727776, 5731752, 5740712, 5746142, 5746942, 5747107, 5748
429, 5754834, 5756589, 5771220, 5778851, 5779301, 5780298, 5780761, 5781728, 5785051,
5793012, 5798136, 5799772, 5800871, 5803392, 5806905, 5807457, 5809689, 5810533, 5812
038, 5821490, 5824971, 5825306, 5826709, 5826900, 5826958, 5829863, 5831301, 5833653,
5833825, 5833888, 5841593, 5844572, 5846754, 5848734, 5851917, 5852286, 5853346, 5854
655, 5857632, 5860935, 5863650, 5872708, 5875832, 5894326, 5895179, 5908748, 5911218,
5917254, 5919839, 5921918, 5923200, 5924857, 5925043, 5941013, 5945245, 5947607, 5950

764, 5957978, 5959991, 5961595, 5961786, 5962099, 5962652, 5967776, 5969492, 5970238,
5973926, 5975827, 5977470, 5978823, 5986101, 5987741, 5988895, 5991366, 6002495, 6005
561, 6007749, 6016604, 6021643, 6024007, 6024614, 6024784, 6024940, 6026635, 6029485,
6033632, 6041782, 6044359, 6046997, 6050005, 6050681, 6052290, 6055498, 6057066, 6058
704, 6061003, 6074204, 6077958, 6086787, 6091409, 6098935, 6101120, 6101255, 6104222,
6114396, 6114863, 6116403, 6119321, 6122343, 6123259, 6126460, 6126648, 6132014, 6144
218, 6146719, 6148174, 6159581, 6176615, 6178182, 6178845, 6179226, 6180590, 6186382,
6187999, 6190165, 6198425, 6199330, 6202339, 6203853, 6205827, 6206370, 6207740, 6209
782, 6210682, 6216479, 6217359, 6219557, 6223749, 6226264, 6226733, 6234418, 6236744,
6237129, 6237906, 6237966, 6240983, 6241129, 6243517, 6243951, 6244477, 6246077, 6248
524, 6253459, 6253627, 6257596, 6262149, 6263708, 6264757, 6265534, 6271391, 6275284,
6282620, 6286166, 6288559, 6296284, 6310412, 6316166, 6318133, 6322307, 6324238, 6325
014, 6327172, 6329080, 6329514, 6332215, 6332678, 6333251, 6340903, 6348485, 6353364,
6353958, 6360996, 6363863, 6368653, 6381527, 6385247, 6385334, 6388195, 6392668, 6394
513, 6395232, 6395575, 6396582, 6397749, 6407477, 6407540, 6408147, 6408433, 6411509,
6412008, 6424006, 6427788, 6428756, 6430963, 6432178, 6434840, 6437115, 6450551, 6452
858, 6455448, 6455454, 6461754, 6462869, 6463723, 6465687, 6466181, 6469369, 6472076,
6473581, 6474229, 6475436, 6485628, 6491509, 6492967, 6497417, 6504842, 6505651, 6507
881, 6508152, 6521874, 6522721, 6528968, 6532996, 6535157, 6538146, 6541481, 6543716,
6545016, 6549218, 6549874, 6550161, 6551514, 6553645, 6560734, 6562237, 6562294, 6577
298, 6584342, 6584848, 6585349, 6586468, 6587044, 6587241, 6600062, 6600359, 6603527,
6606784, 6612813, 6614616, 6619645, 6625290, 6630496, 6635219, 6641451, 6648351, 6649
792, 6656467, 6660207, 6677158, 6687752, 6690894, 6691324, 6696204, 6698316, 6703227,
6703614, 6719025, 6720318, 6721658, 6727436, 6732031, 6739693, 6740804, 6741503, 6745
615, 6745993, 6753396, 6753866, 6761813, 6765868, 6767703, 6770567, 6772264, 6774193,
6778625, 6779845, 6781537, 6788013, 6788959, 6790656, 6791129, 6793650, 6795757, 6796
228, 6797147, 6798552, 6801388, 6804278, 6804999, 6805097, 6809437, 6810468, 6813779,
6814570, 6818258, 6818614, 6826846, 6832274, 6853605, 6859667, 6860191, 6862150, 6865
325, 6865482, 6866932, 6867192, 6872908, 6873751, 6875004, 6880488, 6884788, 6888806,
6899641, 6901733, 6902992, 6904157, 6907191, 6913085, 6913105, 6913897, 6923889, 6936
355, 6947883, 6951493, 6954494, 6958689, 6959078, 6959629, 6965828, 6972991, 6979052,
6980813, 6991527, 6994608, 7013176, 7015624, 7019325, 7025066, 7034773, 7035862, 7037
523, 7038917, 7044145, 7047913, 7050418, 7054816, 7055616, 7057514, 7057540, 7062960,
7066755, 7067122, 7069475, 7074951, 7076627, 7079265, 7081909, 7084014, 7084127, 7085
147, 7087750, 7088542, 7088945, 7092024, 7101438, 7104968, 7105758, 7106100, 7111851,
7114388, 7114723, 7115398, 7139108, 7139334, 7144283, 7153205, 7162719, 7162869, 7165
714, 7167104, 7168329, 7171562, 7175719, 7178510, 7181265, 7181534, 7187529, 7190095,
7193066, 7195367, 7196221, 7198452, 7199337, 7204225, 7207010, 7207688, 7213503, 7217
305, 7219743, 7223699, 7235026, 7235291, 7236344, 7236602, 7236832, 7237486, 7253727,
7253856, 7257578, 7260125, 7260144, 7264701, 7267113, 7269055, 7270272, 7272359, 7272
696, 7276733, 7282163, 7282685, 7287456, 7288683, 7289279, 7308442, 7309814, 7311006,
7313441, 7315725, 7317741, 7318774, 7331786, 7332618, 7332704, 7333976, 7335300, 7339
107, 7339267, 7340853, 7343299, 7349424, 7355155, 7361310, 7378348, 7379457, 7389429,
7390814, 7394420, 7395910, 7403363, 7404711, 7411413, 7411986, 7414920, 7418711, 7421
159, 7424323, 7424788, 7425466, 7429868, 7431218, 7444582, 7447994, 7451398, 7456924,
7459013, 7462955, 7463375, 7463621, 7472343, 7476569, 7477106, 7478267, 7481215, 7497
739, 7498302, 7498966, 7505656, 7505796, 7512216, 7515133, 7517683, 7517776, 7528655,
7529960, 7535456, 7545239, 7548307, 7556974, 7561004, 7563563, 7563659, 7563763, 7569
174, 7574499, 7575110, 7575369, 7579093, 7598795, 7599928, 7600109, 7600817, 7604494,
7606420, 7608444, 7609074, 7610555, 7611346, 7611469, 7612328, 7613563, 7615466, 7633
734, 7635171, 7636650, 7637533, 7638520, 7638892, 7657199, 7660121, 7666587, 7670914,
7674946, 7694297, 7701111, 7701927, 7702403, 7702844, 7703164, 7705233, 7710580, 7713
140, 7723000, 7725159, 7728951, 7729448, 7732106, 7737396, 7738804, 7741240, 7744142,
7752021, 7765332, 7767287, 7772353, 7779301, 7780980, 7782629, 7783488, 7783965, 7792
953, 7795911, 7796899, 7798419, 7802954, 7811781, 7812140, 7817050, 7817217, 7820328,
7821179, 7821776, 7825254, 7828491, 7829652, 7830521, 7831028, 7848271, 7857150, 7858
504, 7861364, 7862495, 7866773, 7867174, 7868666, 7869102, 7873575, 7875229, 7875912,
7876206, 7877402, 7890027, 7894479, 7904815, 7911769, 7911794, 7915512, 7920763, 7921
334, 7926998, 7930077, 7938374, 7938885, 7938906, 7939242, 7940016, 7946086, 7948343,
7950431, 7951137, 7953847, 7954803, 7956936, 7967147, 7973162, 7976490, 7976927, 7978

198, 7984895, 7996728, 7999306, 8000046, 8000341, 8002134, 8002364, 8012174, 8015920, 8022974, 8025018, 8026715, 8028906, 8037545, 8049760, 8050741, 8059124, 8059163, 8062578, 8068361, 8069918, 8073792, 8081239, 8083856, 8083991, 8088487, 8099434, 8104398, 8110575, 8118671, 8120498, 8128820, 8130721, 8131705, 8132486, 8134142, 8138081, 8139470, 8139534, 8139899, 8142628, 8145391, 8148005, 8150850, 8151566, 8156080, 8156715, 8162240, 8164928, 8175063, 8182233, 8190293, 8203281, 8203597, 8223981, 8224971, 8226558, 8227916, 8228768, 8234572, 8235410, 8258393, 8263131, 8274346, 8276165, 8281635, 8305302, 8347914, 8353493, 8376796, 8377409, 8378566, 8381254, 8382370, 8400251, 8400458, 8413551, 8431573, 8432380, 8447124, 8485173, 8486219, 8494970, 8495659, 8498146, 8525546, 8534251, 8536222, 8549120, 8555135, 8564180, 8568677, 8574037, 8577660, 8579956, 8580374, 8582643, 8584170, 8585004, 8597522, 8597691, 8598394, 8600820, 8644086, 8646657, 8650996, 8656794, 8662139, 8663789, 8675384, 8675589, 8678580, 8678849, 8689302, 8692574, 8705083, 8705144, 8707354, 8711631, 8713595, 8722594, 8725279, 8727619, 8733735, 8742157, 8744503, 8745162, 8745352, 8745628, 8745888, 8747247, 8749288, 8750122, 8751962, 8766412, 8769784, 8770288, 8770731, 8772754, 8781245, 8802916, 8804356, 8815127, 8824256, 8827960, 8850404, 8853618, 8856883, 8861859, 8867428, 8869608, 8876590, 8891460, 8891684, 8903787, 8911179, 8913492, 8916255, 8918121, 8921900, 8922102, 8933678, 8935727, 8936443, 8936569, 8957722, 8960628, 8965012, 8977004, 8979850, 8986807, 8986951, 8988808, 9013172, 9022565, 9024633, 9029366, 9051516, 9056272, 9064811, 9065100, 9068963, 9076916, 9079242, 9084457, 9090597, 9095099, 9099550, 9100329, 9117350, 9120241, 9137566, 9138407, 9138531, 9140876, 9142202, 9143296, 9143507, 9147789, 9153784, 9155990, 9174350, 9174665, 9174968, 9181030, 9184118, 9188113, 9195444, 9198097, 9198271, 9200366, 9200696, 9206426, 9209255, 9225017, 9231308, 9231906, 9248197, 9261793, 9285607, 9288483, 9294590, 9300212, 9303388, 9305300, 9330463, 9330988, 9332547, 9341990, 9342005, 9344425, 9353702, 9353953, 9366429, 9377459, 9377577, 9384716, 9388566, 9392213, 9393349, 9393666, 9394114, 9394133, 9394138, 9400483, 9405319, 9407149, 9411611, 9413701, 9415459, 9417518, 9425061, 9440523, 9442290, 9442366, 9444688, 9445556, 9445577, 9445578, 9450603, 9454181, 9460104, 9464003, 9470262, 9476328, 9484775, 9517845, 9531463, 9531528, 9538107, 9541648, 9559120, 9562214, 9591321, 9591540, 9593498, 9595767, 9596951, 9603002, 9616886, 9618154, 9619356, 9622454, 9625061, 9637706, 9637937, 9639674, 9642660, 9642721, 9648676, 9657006, 9668315, 9670967, 9673702, 9678195, 9678591, 9679378, 9679637, 9696651, 9723694, 9738587, 9744092, 9753172, 9758033, 9786180, 9806890, 9807483, 9820476, 9822065, 9859152]
Cluster 2 (Cluster 2):
Users: [4754125]

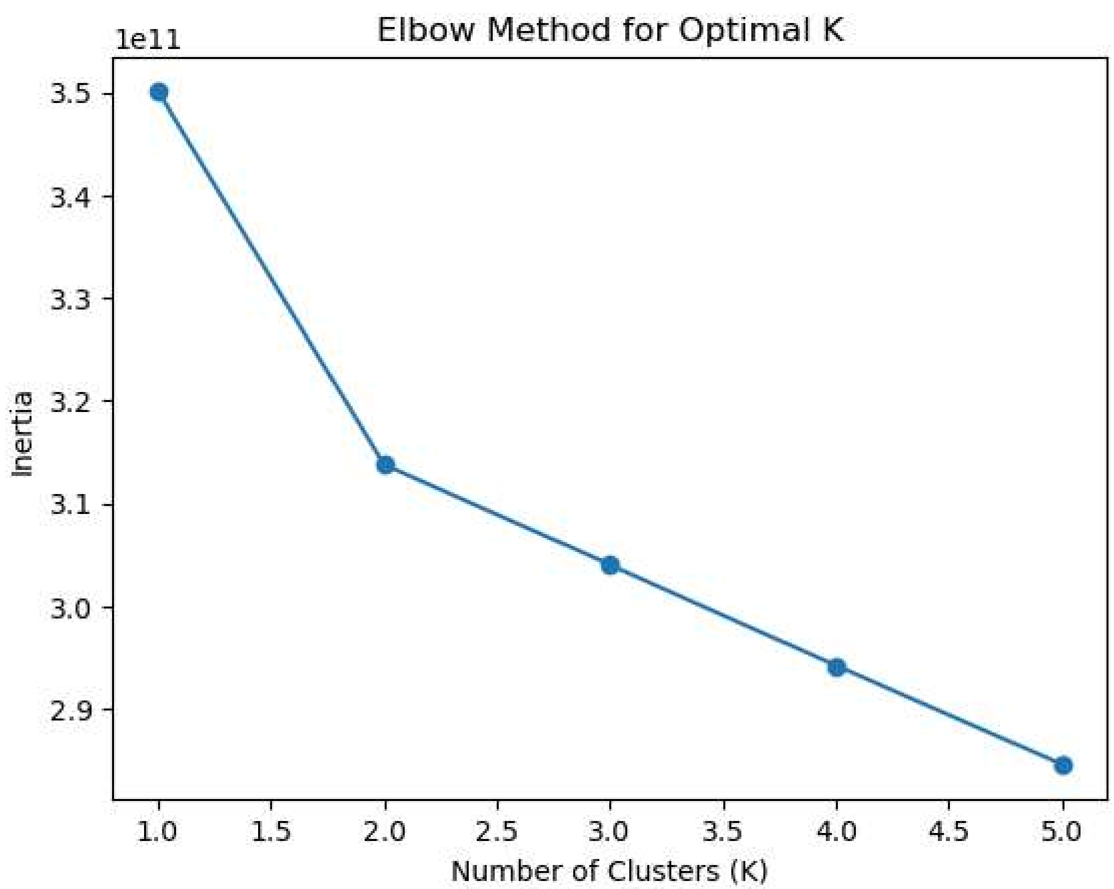


K means Evaluation

In [24]:
```
inertias = []
for k_value in range(1, 6):
    model = KMeans(n_clusters=k_value, random_state=0)
    model.fit(user_data_array)
    inertias.append(model.inertia_)

plt.plot(range(1, 6), inertias, marker='o')
plt.xlabel('Number of Clusters (K)')
plt.ylabel('Inertia')
plt.title('Elbow Method for Optimal K')
```

```
C:\Users\abhiv\AppData\Local\anaconda3\Lib\site-packages\sklearn\cluster\_kmeans.py:8
70: FutureWarning: The default value of `n_init` will change from 10 to 'auto' in 1.
4. Set the value of `n_init` explicitly to suppress the warning
  warnings.warn(
C:\Users\abhiv\AppData\Local\anaconda3\Lib\site-packages\sklearn\cluster\_kmeans.py:8
70: FutureWarning: The default value of `n_init` will change from 10 to 'auto' in 1.
4. Set the value of `n_init` explicitly to suppress the warning
  warnings.warn(
C:\Users\abhiv\AppData\Local\anaconda3\Lib\site-packages\sklearn\cluster\_kmeans.py:8
70: FutureWarning: The default value of `n_init` will change from 10 to 'auto' in 1.
4. Set the value of `n_init` explicitly to suppress the warning
  warnings.warn(
C:\Users\abhiv\AppData\Local\anaconda3\Lib\site-packages\sklearn\cluster\_kmeans.py:8
70: FutureWarning: The default value of `n_init` will change from 10 to 'auto' in 1.
4. Set the value of `n_init` explicitly to suppress the warning
  warnings.warn(
C:\Users\abhiv\AppData\Local\anaconda3\Lib\site-packages\sklearn\cluster\_kmeans.py:8
70: FutureWarning: The default value of `n_init` will change from 10 to 'auto' in 1.
4. Set the value of `n_init` explicitly to suppress the warning
  warnings.warn(
```

Out[24]: Text(0.5, 1.0, 'Elbow Method for Optimal K')



In [25]:
```python
from sklearn.metrics import silhouette_score

silhouette_avg = silhouette_score(user_data_array, y_pred)
print(f"Silhouette Score: {silhouette_avg}")
```

```
Silhouette Score: 0.9098500527957909
```

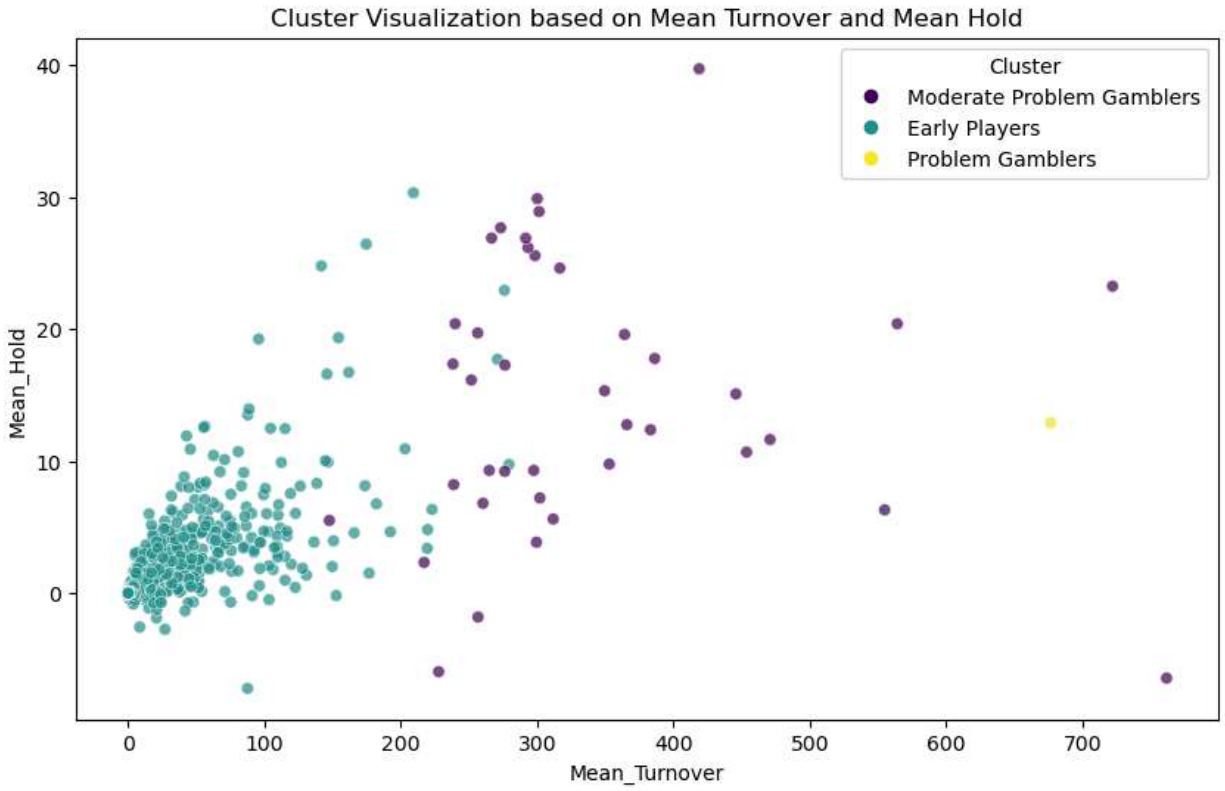
K means Labeling Analysis

In [26]:
```
user_data_3d_turnover = user_data_3d['Turnover']

# Create a new DataFrame to store the aggregated features
agguser_allfeature = pd.DataFrame()

#['Turnover', 'Hold', 'NumberofBets', 'YearofBirth', 'Interventiontype_first']

# Calculate the mean of each row and add it as a new column
agguser_allfeature['Mean_Turnover'] = user_data_3d_turnover.mean(axis=1)
agguser_allfeature['Mean_Hold'] = user_data_3d['Hold'].mean(axis=1)
agguser_allfeature['Mean_NumberofBets'] = user_data_3d['NumberofBets'].mean(axis=1)
agguser_allfeature['Mean_YearofBirth'] = user_data_3d['Age_until_2010'].mean(axis=1)
agguser_allfeature['Mean_Interventiontype_first'] = user_data_3d['Interventiontype_fir
agguser_allfeature['Cluster'] = y_pred


# Set the index of agguser_allfeaturue to match user_data_3d_turnover
agguser_allfeature.index = user_data_3d_turnover.index.tolist()

# Display the updated DataFrame
print(agguser_allfeature)
```

```
         Mean_Turnover  Mean_Hold  Mean_NumberofBets  Mean_YearofBirth  \
31965        29.325776   4.097601           3.243651         12.808167
32639         0.001836   0.001836           0.000343          0.014070
36822         0.008236  -0.008476           0.002059          0.027454
36916         0.027111   0.027111           0.001030          0.028140
74438         0.383665   0.150206           0.008236          0.180165
...                ...        ...                ...               ...
9806890       0.028240  -0.000159           0.002745          0.009952
9807483       0.102132  -0.015731           0.003775          0.009266
9820476       0.030298  -0.001102           0.001030          0.008579
9822065       0.107430   0.016380           0.058339          0.354839
9859152       0.004461   0.004461           0.001716          0.009609

         Mean_Interventiontype_first  Cluster
31965                       2.627316        1
32639                       0.000000        1
36822                       0.000000        1
36916                       0.004118        1
74438                       0.000000        1
...                              ...      ...
9806890                     0.000000        1
9807483                     0.000343        1
9820476                     0.004461        1
9822065                     0.098147        1
9859152                     0.000000        1

[3161 rows x 6 columns]
```

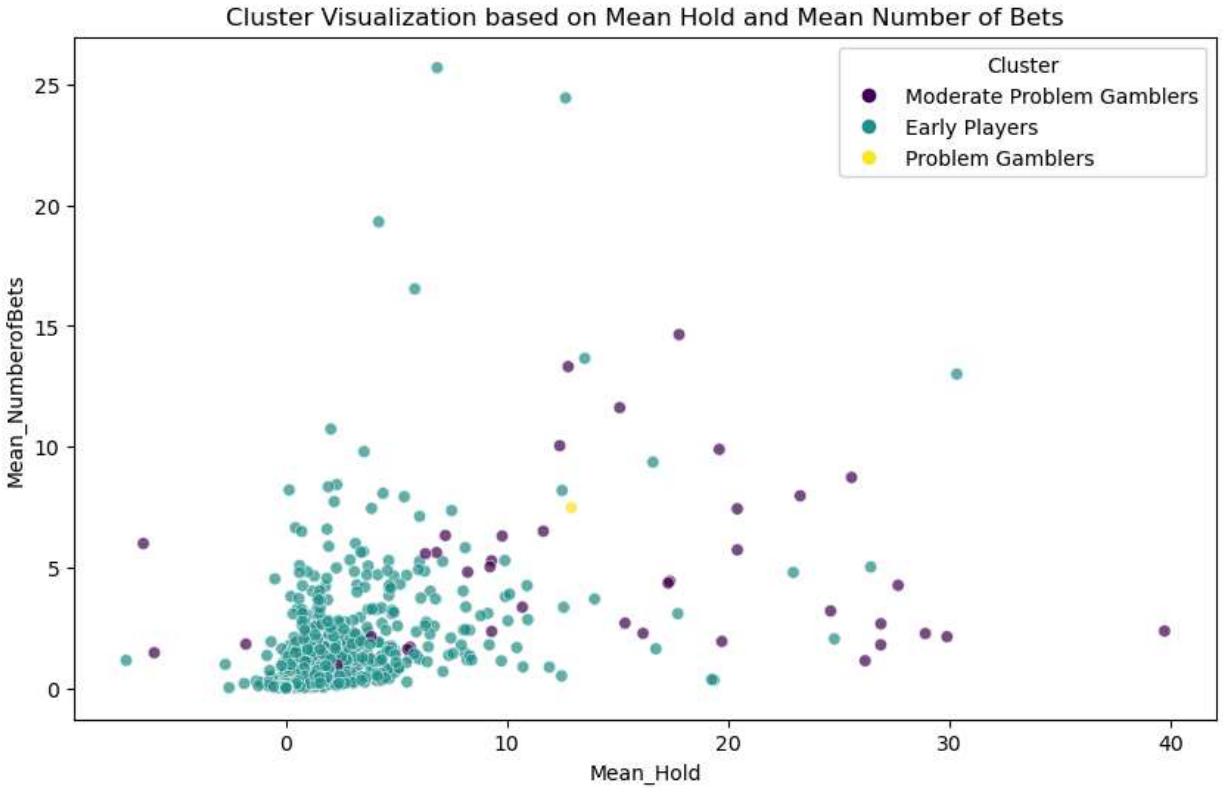
In [27]:
```
from pandas.plotting import parallel_coordinates
import seaborn as sns

# Cast the index to integers
agguser_allfeature.index = agguser_allfeature.index.astype(int)

# Create a scatter plot
plt.figure(figsize=(10, 6))
sns.scatterplot(data=agguser_allfeature, x='Mean_Turnover', y='Mean_Hold', hue='Cluste
```

```
plt.xlabel('Mean_Turnover')
plt.ylabel('Mean_Hold')
plt.title('Cluster Visualization based on Mean Turnover and Mean Hold')
#plt.legend(title='Cluster', loc='upper right')
legend_labels = ['Moderate Problem Gamblers','Early Players','Problem Gamblers']# [f'C
legend = plt.legend(handles=scatter.legend_elements()[0], title='Cluster', labels=lege


plt.show()
```



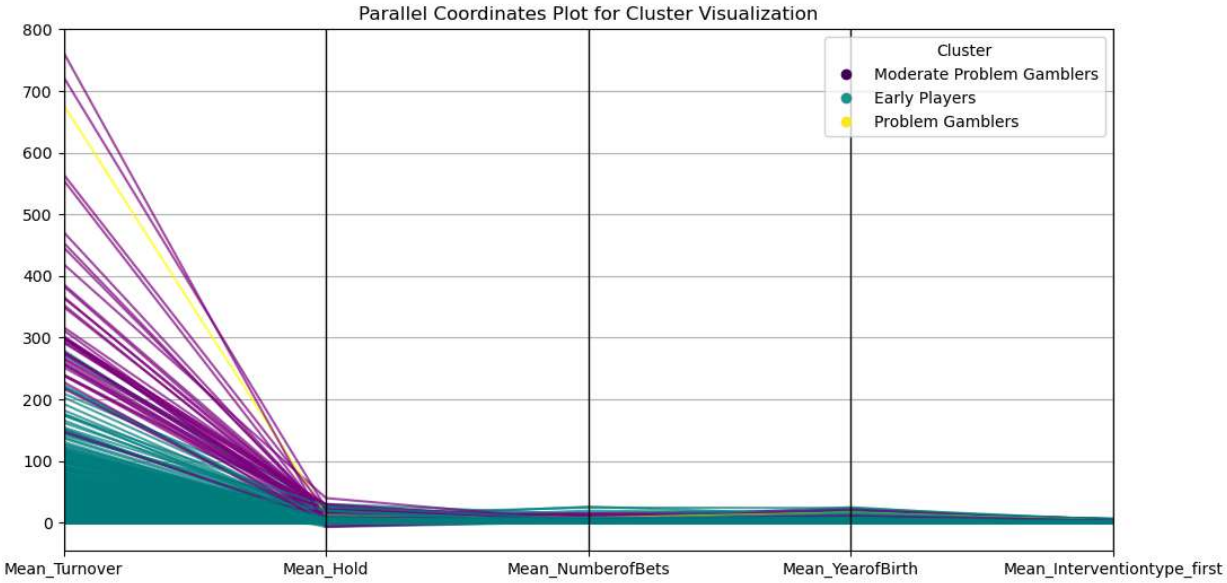
In [28]:

```
# Create a scatter plot
plt.figure(figsize=(10, 6))
sns.scatterplot(data=agguser_allfeature, x='Mean_Hold', y='Mean_NumberofBets', hue='Cl
plt.xlabel('Mean_Hold')
plt.ylabel('Mean_NumberofBets')
plt.title('Cluster Visualization based on Mean Hold and Mean Number of Bets')
#plt.legend(title='Cluster', loc='upper right')
legend_labels = ['Moderate Problem Gamblers','Early Players','Problem Gamblers']# [f'C
legend = plt.legend(handles=scatter.legend_elements()[0], title='Cluster', labels=lege

plt.show()
```



In [29]:
```
from pandas.plotting import parallel_coordinates

# Define custom colors for each cluster
custom_colors = [ 'teal','purple', 'yellow']

# Create a parallel coordinates plot with custom colors
plt.figure(figsize=(12, 6))
parallel_coordinates(agguser_allfeature, 'Cluster', colormap='viridis', alpha=0.7, col
plt.title('Parallel Coordinates Plot for Cluster Visualization')

# Add a custom legend
legend_labels = ['Moderate Problem Gamblers','Early Players','Problem Gamblers']# [f'C
legend = plt.legend(handles=scatter.legend_elements()[0], title='Cluster', labels=lege

plt.show()
```

```
C:\Users\abhiv\AppData\Local\Temp\ipykernel_14916\1398724544.py:8: UserWarning: 'colo
r' and 'colormap' cannot be used simultaneously. Using 'color'
  parallel_coordinates(agguser_allfeature, 'Cluster', colormap='viridis', alpha=0.7,
color=custom_colors)
C:\Users\abhiv\AppData\Local\anaconda3\Lib\site-packages\IPython\core\pylabtools.py:1
52: UserWarning: Creating legend with loc="best" can be slow with large amounts of da
ta.
  fig.canvas.print_figure(bytes_io, **kw)
```



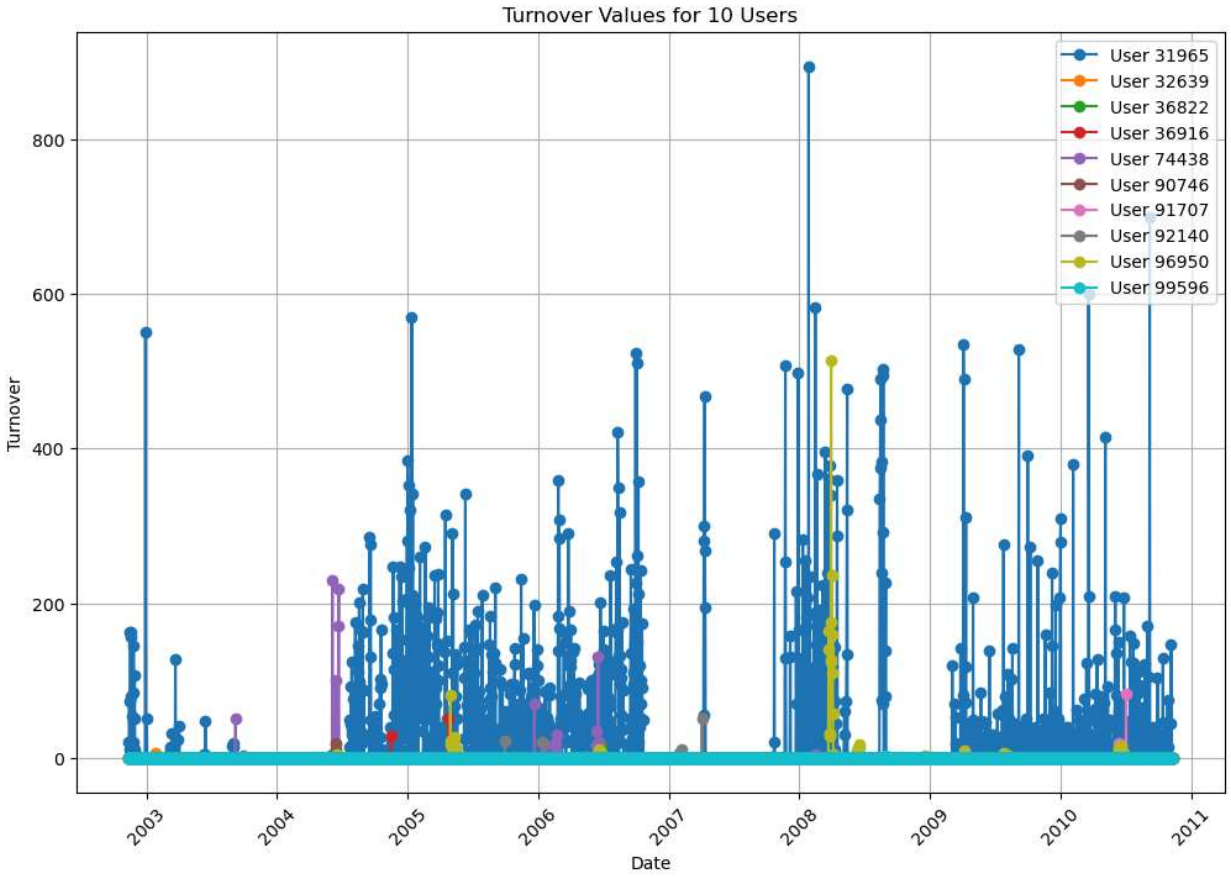
In [30]:
```
### Interactive Slider to understand Clustering using K - means
```

In [31]:
```
import numpy as np
import pandas as pd
from sklearn.cluster import KMeans
import matplotlib.pyplot as plt
from ipywidgets import interact

from ipywidgets import interact, DatePicker



# Create a function to plot the clustering results for a specific time frame
def plot_clusters(time_frame):
    print(time_frame)
    start_date = '2002-11-12' #).date()
        # Extract the year from the time_frame and construct a new start_date
    start_year = str(int(time_frame))
    start_date = pd.to_datetime(start_year + '-11-12')

    plt.figure(figsize=(8, 6))
    plt.scatter(user_data_3d['NumberofBets'][start_date], user_data_3d['Hold'][start_d
   # plt.scatter(user_data_array[:, 0], user_data_array[:, 2], c=y_pred, cmap='viridis
    plt.xlabel('NumberofBets')
    plt.ylabel('Hold')
    plt.title(f'K-means Clustering: NumberofBets vs Hold (Time Frame {time_frame})')


    # filtered_data_array = user_data_array[time_frame_condition]
    # filtered_y_pred = y_pred[time_frame_condition]

    # Scatter plot the filtered data
    # plt.scatter(filtered_data_array[:, 0], filtered_data_array[:, 2], c=filtered_y_p

    plt.show()


start_date='2002-11-12'
end_date='2010-11-10'
```

```
# Create the interaction using the date pickers
interact(plot_clusters, time_frame=(2002, 2009))
```

Out[31]:
```
interactive(children=(IntSlider(value=2005, description='time_frame', max=2009, min=2
002), Output()), _dom_cla…
<function __main__.plot_clusters(time_frame)>
```

In [32]:
```
import matplotlib.pyplot as plt

# List of 10 user IDs to plot
user_ids_to_plot = [31965, 32639, 36822, 36916, 74438, 90746, 91707, 92140, 96950, 995

# Create a subplot for the graph
plt.figure(figsize=(12, 8))
plt.title('Turnover Values for 10 Users')
plt.xlabel('Date')
plt.ylabel('Turnover')
plt.grid(True)

for user_id in user_ids_to_plot:



    user_data_for_user = user_data_3d.loc[user_id, 'Turnover']

    # Extract the turnover values
    turnover_values = user_data_for_user.to_numpy()

    # Extract the corresponding date indices
    dates = user_data_for_user.index.get_level_values('Aggregate_Date')

    # Plot the turnover values against dates
    plt.plot(dates, turnover_values, marker='o', linestyle='-', label=f'User {user_id}

# Add a legend to differentiate the users
plt.legend(loc='upper right')

# Rotate the x-axis labels for better visibility
plt.xticks(rotation=45)

# Show the plot
plt.show()
```



In [33]:
```
user_data_3d
```

Out[33]:

	Turnover	...										
Aggregate_Date	**2002-11-12**	**2002-11-13**	**2002-11-14**	**2002-11-15**	**2002-11-16**	**2002-11-17**	**2002-11-18**	**2002-11-19**	**2002-11-20**	**2002-11-22**	**...**	**2010-11-01**
UserID												
31965	20.0	0.0	73.18	10.0	163.28	162.34	156.0	3.74	80.0	0.0	...	0.0
32639	0.0	0.0	0.00	0.0	0.00	0.00	0.0	0.00	0.0	0.0	...	0.0
36822	0.0	0.0	0.00	0.0	0.00	0.00	0.0	0.00	0.0	0.0	...	0.0
36916	0.0	0.0	0.00	0.0	0.00	0.00	0.0	0.00	0.0	0.0	...	0.0
74438	0.0	0.0	0.00	0.0	0.00	0.00	0.0	0.00	0.0	0.0	...	0.0
...	...	...	...	...	...	...	...	...	...	...	...	...
9806890	0.0	0.0	0.00	0.0	0.00	0.00	0.0	0.00	0.0	0.0	...	0.0
9807483	0.0	0.0	0.00	0.0	0.00	0.00	0.0	0.00	0.0	0.0	...	0.0
9820476	0.0	0.0	0.00	0.0	0.00	0.00	0.0	0.00	0.0	0.0	...	0.0
9822065	0.0	0.0	0.00	0.0	0.00	0.00	0.0	0.00	0.0	0.0	...	0.0
9859152	0.0	0.0	0.00	0.0	0.00	0.00	0.0	0.00	0.0	0.0	...	0.0

3161 rows × 23312 columns

In [34]:
```
# features to consider : turnover, hold , number of bets for time series prediction
```

In [35]:
```
moderate_pg_players= [868583, 1175809, 1411743, 1457496, 1486136, 1662632, 1679490, 17

count = len(moderate_pg_players)
print("Count of moderate addicted players:", count)
```

```
Count of moderate addicted players: 40
```

In [36]:
```
bv=user_data_3d['Hold'][user_data_3d.index == 868583]
bv.value_counts
#user_data_3d[user_data_3d['UserID']==868583]
```

Out[36]:
```
<bound method DataFrame.value_counts of Aggregate_Date  2002-11-12  2002-11-13  2002-
11-14  2002-11-15  2002-11-16  \
UserID
868583                 0.0         0.0         0.0         0.0         0.0

Aggregate_Date  2002-11-17  2002-11-18  2002-11-19  2002-11-20  2002-11-22  \
UserID
868583                 0.0         0.0         0.0         0.0         0.0

Aggregate_Date  ...  2010-11-01  2010-11-02  2010-11-03  2010-11-04  \
UserID          ...
868583          ...         0.0         0.0         0.0         0.0

Aggregate_Date  2010-11-05  2010-11-06  2010-11-07  2010-11-08  2010-11-09  \
UserID
868583                 0.0         0.0         0.0         0.0         0.0

Aggregate_Date  2010-11-10
UserID
868583                 0.0

[1 rows x 2914 columns]>
```

Filtering of data based on hypothesis testing for Stationarity

In [37]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from statsmodels.tsa.stattools import adfuller

# Create a dictionary to store user IDs, their corresponding 'is_stationary' values, a
is_stationary_data = {'UserID': moderate_pg_players, 'is_stationary': [], 'hypothesis_

# Iterate through the list of user IDs
for user_id in moderate_pg_players:
    # Extract the user's turnover data
    user_turnover = user_data_3d['Turnover'][user_data_3d.index == user_id].values.rav

    # Perform the ADF test for stationarity
    result = adfuller(user_turnover)

    # Check if the time series is stationary based on the p-value
    if result[1] <= 0.05:
```

```
        is_stationary = 1  # Stationary
        hypothesis_test_result = 'Reject Null Hypothesis'  # Stationarity is significa
    else:
        is_stationary = 0  # Not stationary
        hypothesis_test_result = 'Fail to Reject Null Hypothesis'  # Stationarity is r

    # Append the 'is_stationary' and 'hypothesis_test_result' values to the list
    is_stationary_data['is_stationary'].append(is_stationary)
    is_stationary_data['hypothesis_test_result'].append(hypothesis_test_result)

# Create a new DataFrame from the dictionary
is_stationary_df = pd.DataFrame(is_stationary_data)

# Print the new DataFrame
print(is_stationary_df)
```

```
     UserID  is_stationary          hypothesis_test_result
0    868583              1          Reject Null Hypothesis
1   1175809              1          Reject Null Hypothesis
2   1411743              1          Reject Null Hypothesis
3   1457496              1          Reject Null Hypothesis
4   1486136              1          Reject Null Hypothesis
5   1662632              1          Reject Null Hypothesis
6   1679490              1          Reject Null Hypothesis
7   1776178              1          Reject Null Hypothesis
8   1790848              1          Reject Null Hypothesis
9   1921204              1          Reject Null Hypothesis
10  2070894              1          Reject Null Hypothesis
11  2150296              1          Reject Null Hypothesis
12  2155065              1          Reject Null Hypothesis
13  2589710              1          Reject Null Hypothesis
14  2704382              1          Reject Null Hypothesis
15  2852203              1          Reject Null Hypothesis
16  3669466              1          Reject Null Hypothesis
17  3852889              1          Reject Null Hypothesis
18  3904422              1          Reject Null Hypothesis
19  3968386              1          Reject Null Hypothesis
20  4006708              1          Reject Null Hypothesis
21  4371320              1          Reject Null Hypothesis
22  4394754              1          Reject Null Hypothesis
23  4412550              1          Reject Null Hypothesis
24  4495603              0  Fail to Reject Null Hypothesis
25  4532357              1          Reject Null Hypothesis
26  5106620              1          Reject Null Hypothesis
27  5308271              1          Reject Null Hypothesis
28  5488160              1          Reject Null Hypothesis
29  5660719              1          Reject Null Hypothesis
30  5678852              1          Reject Null Hypothesis
31  5723033              1          Reject Null Hypothesis
32  6158120              1          Reject Null Hypothesis
33  6175402              1          Reject Null Hypothesis
34  6239380              0  Fail to Reject Null Hypothesis
35  6283338              1          Reject Null Hypothesis
36  6709379              1          Reject Null Hypothesis
37  6985339              1          Reject Null Hypothesis
38  7192925              1          Reject Null Hypothesis
39  4754125              1          Reject Null Hypothesis
```

In [38]:
```
# Stationray for Hold feature
```

In [39]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from statsmodels.tsa.stattools import adfuller

# Create a dictionary to store user IDs and their corresponding 'is_stationary' values
is_stationary_data = {'UserID': moderate_pg_players, 'is_stationary_hold': []}

# Iterate through the list of user IDs
for user_id in moderate_pg_players:
    # Extract the user's turnover data
    user_turnover = user_data_3d['Hold'][user_data_3d.index == user_id].values.ravel()

    # Perform the ADF user_data_3d_timeseries_stationarity_check
    result = adfuller(user_turnover)

    # Check if the time series is stationary based on the p-value
    if result[1] <= 0.05:
        is_stationary = 1  # Stationary
    else:
        is_stationary = 0  # Not stationary

    # Append the 'is_stationary' value to the list
    is_stationary_data['is_stationary_hold'].append(is_stationary)

# Create a new DataFrame from the dictionary
is_stationary_df['is_stationary_hold'] = is_stationary_data['is_stationary_hold']

# Print the new DataFrame
#print(is_stationary_df)
```

In [40]:
```
#NumberofBets
```

In [41]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from statsmodels.tsa.stattools import adfuller

# Create a dictionary to store user IDs and their corresponding 'is_stationary' values
is_stationary_data = {'UserID': moderate_pg_players, 'is_stationary_NumberofBets': []}

# Iterate through the list of user IDs
for user_id in moderate_pg_players:
    # Extract the user's turnover data
    user_turnover = user_data_3d['NumberofBets'][user_data_3d.index == user_id].values

    # Perform the ADF user_data_3d_timeseries_stationarity_check
    result = adfuller(user_turnover)

    # Check if the time series is stationary based on the p-value
    if result[1] <= 0.05:
        is_stationary = 1  # Stationary
    else:
        is_stationary = 0  # Not stationary

    # Append the 'is_stationary' value to the list
    is_stationary_data['is_stationary_NumberofBets'].append(is_stationary)
```

```
# Create a new DataFrame from the dictionary
is_stationary_df['is_stationary_NumberofBets'] = is_stationary_data['is_stationary_Num

# Print the new DataFrame
#print(is_stationary_df)
```

In [42]:
```
#YearofBirth
```

In [43]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from statsmodels.tsa.stattools import adfuller

# Create a dictionary to store user IDs and their corresponding 'is_stationary' values
is_stationary_data = {'UserID': moderate_pg_players, 'is_stationary_Age_until_2010': [

# Iterate through the list of user IDs
for user_id in moderate_pg_players:
    # Extract the user's turnover data
    user_turnover = user_data_3d['Age_until_2010'][user_data_3d.index == user_id].valu

    # Perform the ADF user_data_3d_timeseries_stationarity_check
    result = adfuller(user_turnover)

    # Check if the time series is stationary based on the p-value
    if result[1] <= 0.05:
        is_stationary = 1  # Stationary
    else:
        is_stationary = 0  # Not stationary

    # Append the 'is_stationary' value to the list
    is_stationary_data['is_stationary_Age_until_2010'].append(is_stationary)

# Create a new DataFrame from the dictionary
is_stationary_df['is_stationary_Age_until_2010'] = is_stationary_data['is_stationary_A

# Print the new DataFrame
#print(is_stationary_df)
```

In [44]:
```
#Interventiontype_first
```

In [45]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from statsmodels.tsa.stattools import adfuller

# Create a dictionary to store user IDs and their corresponding 'is_stationary' values
is_stationary_data = {'UserID': moderate_pg_players, 'is_stationary_Interventiontype_f

# Iterate through the list of user IDs
for user_id in moderate_pg_players:
    # Extract the user's turnover data
    user_turnover = user_data_3d['Interventiontype_first'][user_data_3d.index == user_

    # Perform the ADF user_data_3d_timeseries_stationarity_check
    result = adfuller(user_turnover)

    # Check if the time series is stationary based on the p-value
```

```
    if result[1] <= 0.05:
        is_stationary = 1  # Stationary
    else:
        is_stationary = 0  # Not stationary

    # Append the 'is_stationary' value to the list
    is_stationary_data['is_stationary_Interventiontype_first'].append(is_stationary)

# Create a new DataFrame from the dictionary
is_stationary_df['is_stationary_Interventiontype_first'] = is_stationary_data['is_stat

# Print the new DataFrame
#print(is_stationary_df)
```

```
C:\Users\abhiv\AppData\Local\anaconda3\Lib\site-packages\statsmodels\regression\linea
r_model.py:940: RuntimeWarning: divide by zero encountered in log
  llf = -nobs2*np.log(2*np.pi) - nobs2*np.log(ssr / nobs) - nobs2
C:\Users\abhiv\AppData\Local\anaconda3\Lib\site-packages\statsmodels\regression\linea
r_model.py:940: RuntimeWarning: divide by zero encountered in log
  llf = -nobs2*np.log(2*np.pi) - nobs2*np.log(ssr / nobs) - nobs2
```

In [46]:
```
is_stationary_df
```

Out[46]:

	UserID	is_stationary	hypothesis_test_result	is_stationary_hold	is_stationary_NumberofBets	is_sta
0	868583	1	Reject Null Hypothesis	1	1	
1	1175809	1	Reject Null Hypothesis	1	1	
2	1411743	1	Reject Null Hypothesis	1	1	
3	1457496	1	Reject Null Hypothesis	1	1	
4	1486136	1	Reject Null Hypothesis	1	1	
5	1662632	1	Reject Null Hypothesis	1	1	
6	1679490	1	Reject Null Hypothesis	1	1	
7	1776178	1	Reject Null Hypothesis	1	1	
8	1790848	1	Reject Null Hypothesis	1	1	
9	1921204	1	Reject Null Hypothesis	1	1	
10	2070894	1	Reject Null Hypothesis	1	1	
11	2150296	1	Reject Null Hypothesis	1	1	
12	2155065	1	Reject Null Hypothesis	1	1	
13	2589710	1	Reject Null Hypothesis	1	1	
14	2704382	1	Reject Null Hypothesis	1	1	
15	2852203	1	Reject Null Hypothesis	1	1	
16	3669466	1	Reject Null Hypothesis	1	1	
17	3852889	1	Reject Null Hypothesis	1	0	
18	3904422	1	Reject Null Hypothesis	1	1	
19	3968386	1	Reject Null Hypothesis	1	1	
20	4006708	1	Reject Null Hypothesis	1	1	
21	4371320	1	Reject Null Hypothesis	1	1	
22	4394754	1	Reject Null Hypothesis	1	1	
23	4412550	1	Reject Null Hypothesis	1	1	
24	4495603	0	Fail to Reject Null Hypothesis	1	1	
25	4532357	1	Reject Null Hypothesis	1	1	
26	5106620	1	Reject Null Hypothesis	1	1	
27	5308271	1	Reject Null Hypothesis	1	1	
28	5488160	1	Reject Null Hypothesis	1	1	
29	5660719	1	Reject Null Hypothesis	1	1	
30	5678852	1	Reject Null Hypothesis	1	1	
31	5723033	1	Reject Null Hypothesis	1	1	
32	6158120	1	Reject Null Hypothesis	1	1	

	UserID	is_stationary	hypothesis_test_result	is_stationary_hold	is_stationary_NumberofBets	is_sta
33	6175402	1	Reject Null Hypothesis	1	1	
34	6239380	0	Fail to Reject Null Hypothesis	1	1	
35	6283338	1	Reject Null Hypothesis	1	1	
36	6709379	1	Reject Null Hypothesis	1	1	
37	6985339	1	Reject Null Hypothesis	1	1	
38	7192925	1	Reject Null Hypothesis	1	1	
39	4754125	1	Reject Null Hypothesis	1	1	

In [47]:
```
arimausers = is_stationary_df[(is_stationary_df['is_stationary'] == 1) & (is_stationar
user_data_3d['Turnover']
x_single_user_turnover=user_data_3d['Turnover'][user_data_3d.index == arimausers['User
```

In [48]:
```
arimausers.reset_index()#[x_single_user_turnover['UserID']=='800']
```

Out[48]:

	index	UserID	is_stationary	hypothesis_test_result	is_stationary_hold	is_stationary_NumberofBet
0	0	868583	1	Reject Null Hypothesis	1	
1	1	1175809	1	Reject Null Hypothesis	1	
2	2	1411743	1	Reject Null Hypothesis	1	
3	3	1457496	1	Reject Null Hypothesis	1	
4	4	1486136	1	Reject Null Hypothesis	1	
5	5	1662632	1	Reject Null Hypothesis	1	
6	6	1679490	1	Reject Null Hypothesis	1	
7	8	1790848	1	Reject Null Hypothesis	1	
8	9	1921204	1	Reject Null Hypothesis	1	
9	10	2070894	1	Reject Null Hypothesis	1	
10	11	2150296	1	Reject Null Hypothesis	1	
11	12	2155065	1	Reject Null Hypothesis	1	
12	13	2589710	1	Reject Null Hypothesis	1	
13	14	2704382	1	Reject Null Hypothesis	1	
14	15	2852203	1	Reject Null Hypothesis	1	
15	16	3669466	1	Reject Null Hypothesis	1	
16	19	3968386	1	Reject Null Hypothesis	1	
17	20	4006708	1	Reject Null Hypothesis	1	
18	22	4394754	1	Reject Null Hypothesis	1	
19	25	4532357	1	Reject Null Hypothesis	1	
20	26	5106620	1	Reject Null Hypothesis	1	
21	28	5488160	1	Reject Null Hypothesis	1	
22	29	5660719	1	Reject Null Hypothesis	1	
23	30	5678852	1	Reject Null Hypothesis	1	
24	31	5723033	1	Reject Null Hypothesis	1	
25	33	6175402	1	Reject Null Hypothesis	1	
26	35	6283338	1	Reject Null Hypothesis	1	
27	36	6709379	1	Reject Null Hypothesis	1	
28	38	7192925	1	Reject Null Hypothesis	1	
29	39	4754125	1	Reject Null Hypothesis	1	

In [49]:
```
print(*arimausers['UserID'])
```

```
868583 1175809 1411743 1457496 1486136 1662632 1679490 1790848 1921204 2070894 215029
6 2155065 2589710 2704382 2852203 3669466 3968386 4006708 4394754 4532357 5106620 548
8160 5660719 5678852 5723033 6175402 6283338 6709379 7192925 4754125
```

Model Fitting

ARIMA/SARIMA model fitting

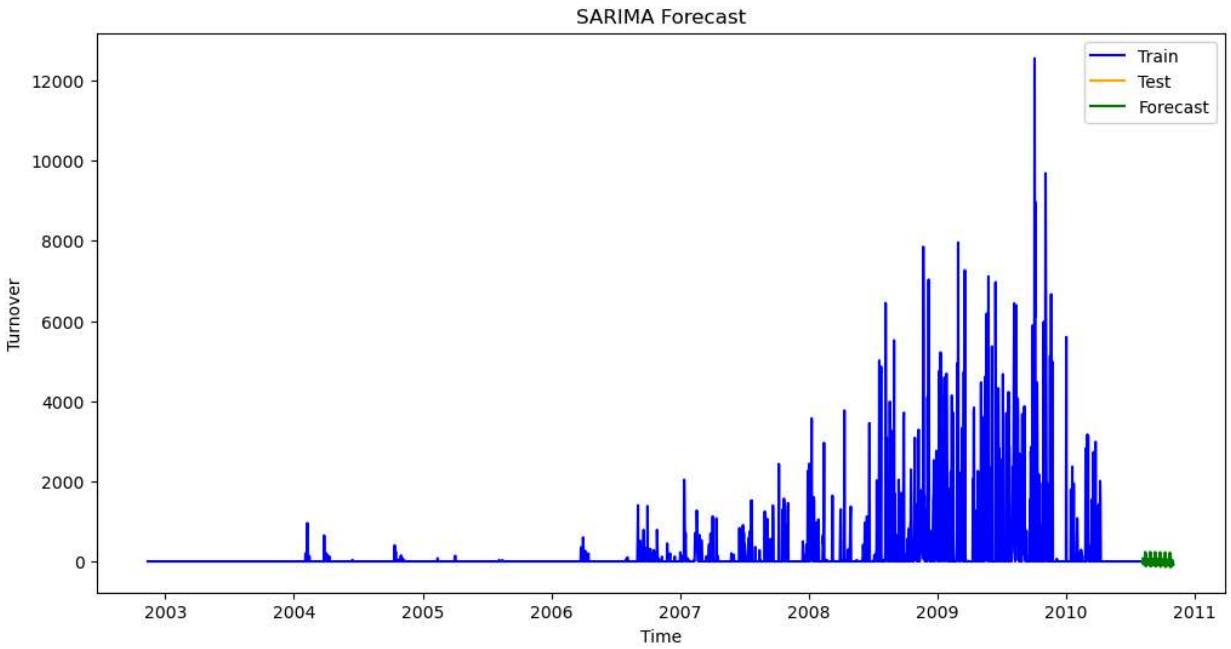
In [50]:

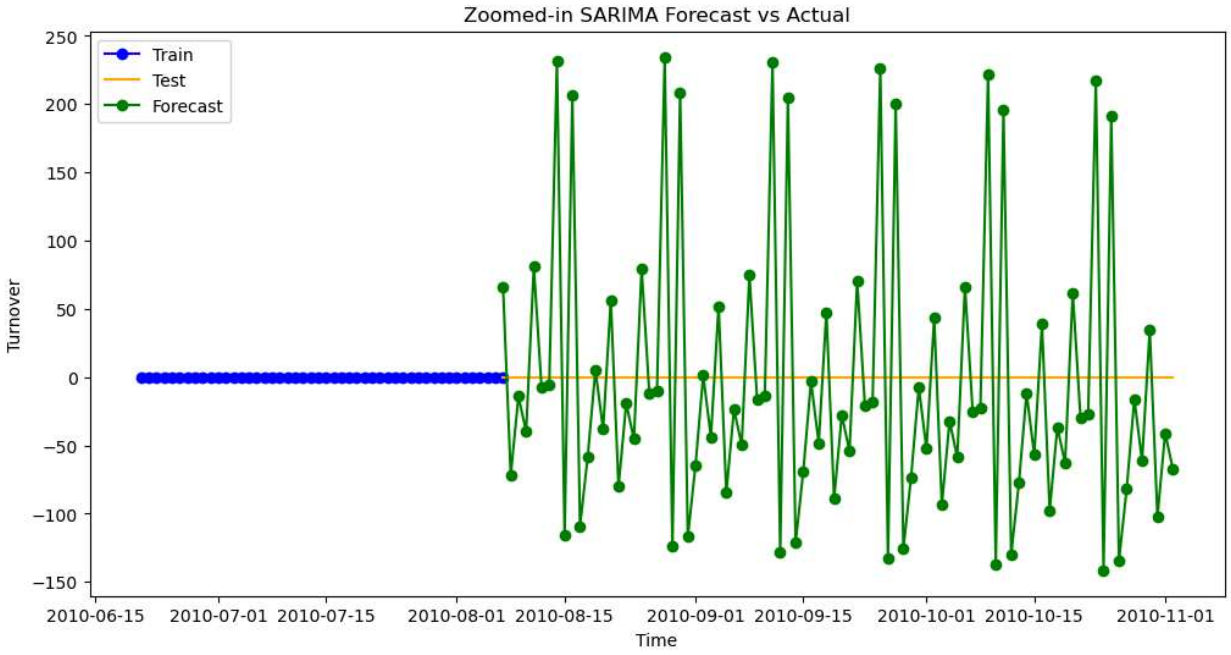
```
from statsmodels.tsa.statespace.sarimax import SARIMAX
from sklearn.preprocessing import MinMaxScaler
import matplotlib.pyplot as plt
import pandas as pd
from sklearn.metrics import mean_squared_error
from statsmodels.tsa.arima.model import ARIMA



# Loop through each user in arimausers
for user_id in arimausers['UserID'].head(10).tolist() + [4754125]:
    x_single_user_turnover = user_data_3d['Turnover'][user_data_3d.index == user_id]
    x_single_user_turnover_ravel = x_single_user_turnover.values.ravel()



    frequency = 'D'  # Daily frequency

    start_date='2002-11-12'
    end_date='2010-11-10'



    # Create a time index for the data
    time_index = pd.date_range(start=start_date, periods=len(x_single_user_turnover_ra


    # Scale the data
    scaler = MinMaxScaler(feature_range=(0, 1))
    x_single_user_turnover_ravel_scaled = scaler.fit_transform(x_single_user_turnover_

    # Split the data into training and test sets
    split_ratio = 0.97  # 97% for training, 3% for testing
    split_index = int(len(x_single_user_turnover_ravel) * split_ratio)

    train_data = x_single_user_turnover_ravel_scaled[:split_index]
    test_data = x_single_user_turnover_ravel_scaled[split_index:]

    # Create a time index for the data
    time_index_train = pd.date_range(start=start_date, periods=len(train_data), freq=f

    # Create a time index for the data
    time_index_test = pd.date_range(start=time_index_train[-1], periods=len(test_data)

    #ARIMA

    # Define the ARIMA model
    model_arima = ARIMA(train_data, order=(1, 1, 1))
```

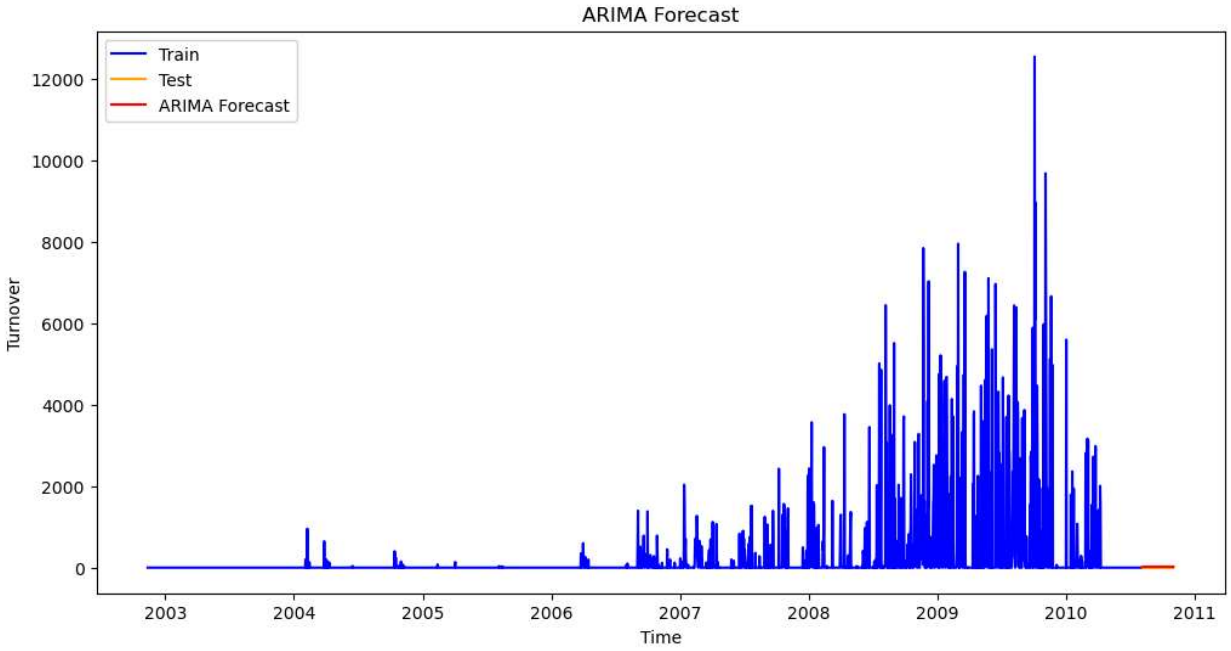
```
# Fit the ARIMA model to the training data
FITmodel_arima = model_arima.fit()

# Forecast the test series using ARIMA
FITmodel_arima_forecast = FITmodel_arima.predict(start=split_index, end=len(x_sing

# Inverse scale the ARIMA forecasted values
FITmodel_arima_forecast = scaler.inverse_transform(FITmodel_arima_forecast.reshape

#SARIMA

# Define the SARIMA model with seasonal difference and order
model_sarima_monthly = SARIMAX(train_data, order=(1, 1, 1), seasonal_order=(1, 1,

# Fit the model to the training data
FITmodel_sarima_monthly = model_sarima_monthly.fit()

# Forecast the test series
FITmodel_sarima_monthly_forecast = FITmodel_sarima_monthly.forecast(steps=len(test

# Inverse scale the forecasted values
FITmodel_sarima_monthly_forecast = scaler.inverse_transform(FITmodel_sarima_monthl

# Inverse scale the training data
train_data_inverse = scaler.inverse_transform(train_data.reshape(-1, 1)).reshape(-

# Inverse scale the test data
test_data_inverse = scaler.inverse_transform(test_data.reshape(-1, 1)).reshape(-1)

# Plot the forecast
plt.figure(figsize=(12, 6))
plt.plot(time_index_train, train_data_inverse, label='Train', color='blue')
forecast_dates = pd.date_range(start=time_index_train[-1], periods=len(test_data),
plt.plot(time_index_test, test_data_inverse, label='Test', color='orange')
plt.plot(forecast_dates, FITmodel_sarima_monthly_forecast, label='Forecast', color
plt.xlabel('Time')
plt.ylabel('Turnover')
plt.legend()
plt.title('SARIMA Forecast')
plt.show()


plt.figure(figsize=(12, 6))
plt.plot(time_index_train[-48:], train_data_inverse[-48:], label='Train', color='b
plt.plot(time_index_test, test_data_inverse, label='Test', color='orange')
plt.plot(forecast_dates, FITmodel_sarima_monthly_forecast, label='Forecast', color
plt.xlabel('Time')
plt.ylabel('Turnover')
plt.legend()
plt.title('Zoomed-in SARIMA Forecast vs Actual')
plt.show()



 # Plot the ARIMA forecast
plt.figure(figsize=(12, 6))
plt.plot(time_index_train, train_data_inverse, label='Train', color='blue')
plt.plot(time_index_test, test_data_inverse, label='Test', color='orange')
plt.plot(forecast_dates, FITmodel_arima_forecast, label='ARIMA Forecast', color='r
```

```
plt.xlabel('Time')
plt.ylabel('Turnover')
plt.legend()
plt.title('ARIMA Forecast')
plt.show()


# Calculate RMSE for ARIMA model
rmse_arima = mean_squared_error(test_data_inverse, FITmodel_arima_forecast[:len(te

# Calculate MAE for ARIMA model
mae_arima = np.mean(np.abs(test_data_inverse - FITmodel_arima_forecast[:len(test_d
# Display results for ARIMA model
print(f"ARIMA Model Results (User {user_id}):")
print(f"RMSE (ARIMA): {rmse_arima}")
print(f"MAE (ARIMA): {mae_arima}")


print("=" * 30)  # Separating results for different users




#SARIMA

# Calculate predictions for training and testing data
train_predictions = FITmodel_sarima_monthly.predict(start=0, end=split_index - 1)
test_predictions = FITmodel_sarima_monthly_forecast

# Trim the predicted values to match the length of the actual values
test_predictions_inverse_trimmed = test_predictions[:len(test_data_inverse)]

# Calculate RMSE for training and testing
#rmse_train = mean_squared_error(train_data_inverse, train_predictions, squared=Fa
rmse_test = mean_squared_error(test_data_inverse, test_predictions_inverse_trimmed
# Calculate MAE for SARIMA model
mae_sarima = np.mean(np.abs(test_data_inverse - test_predictions_inverse_trimmed))

# Display results for each user
print(f"User {user_id}:")
print(f"SARIMAX RMSE : {rmse_test}")
print(f"SARIMAX MAE : {mae_sarima}")
print("=" * 30)  # Separating results for different users


# Compute differences between consecutive predicted values
differences = np.diff(test_predictions.flatten())

# Set a threshold to identify surges
threshold_difference = 150

# Identify surges based on differences and threshold
surge_indices = np.where(differences > threshold_difference)[0]

# Print the timestamps of the points right before a surge
for index in surge_indices:
    if index > 0:
        surge_start_timestamp = time_index_test[index]
```

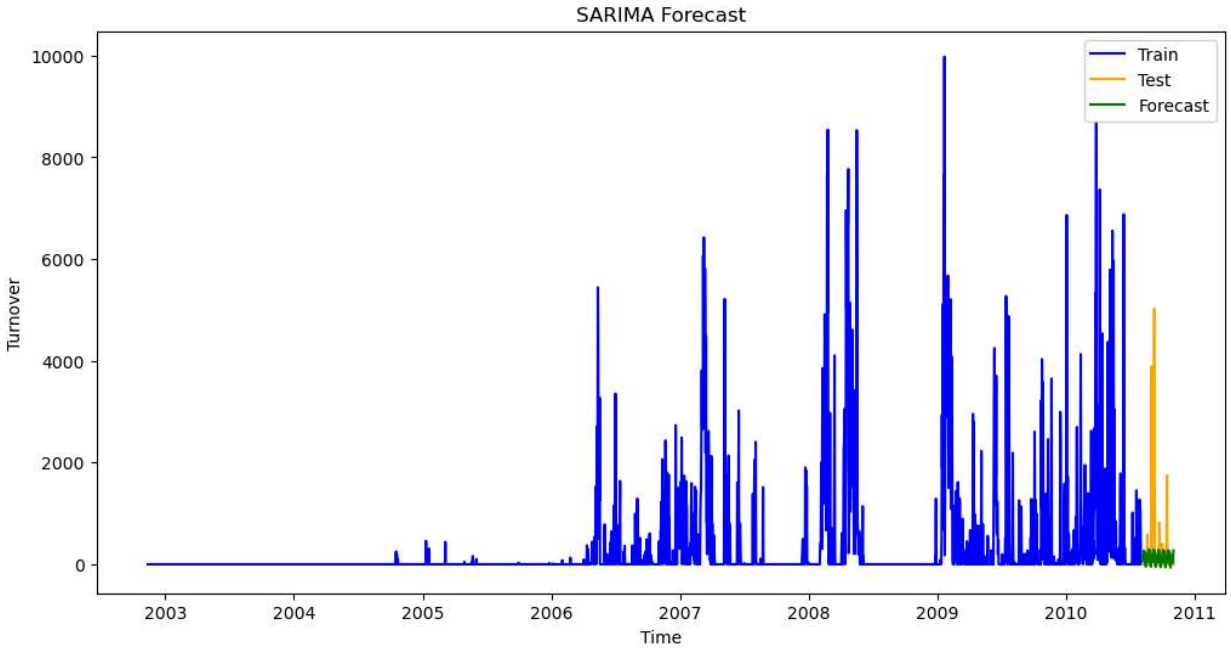
```
print(f"PG Detected right before: {surge_start_timestamp}")
```

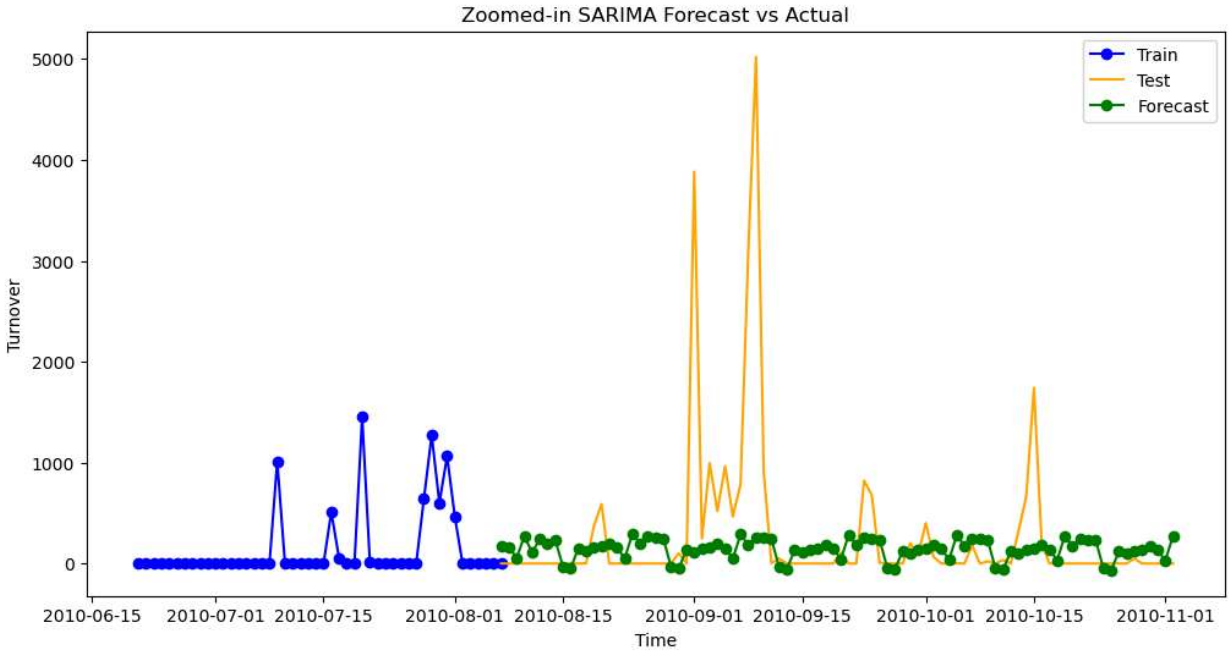


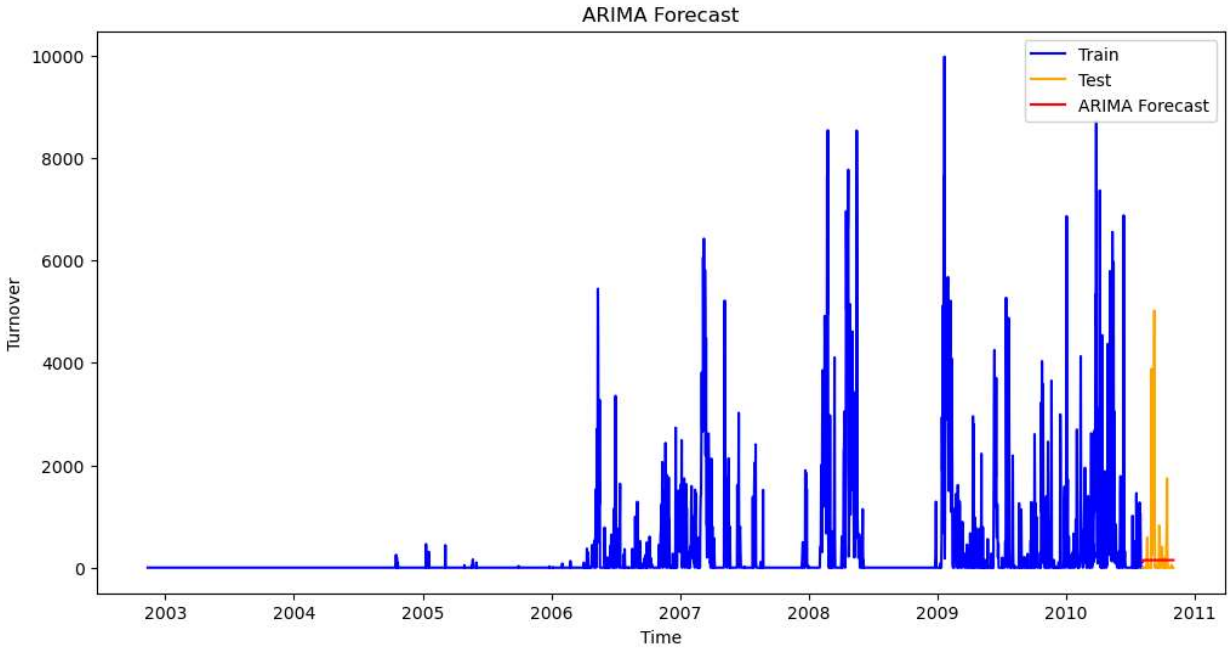




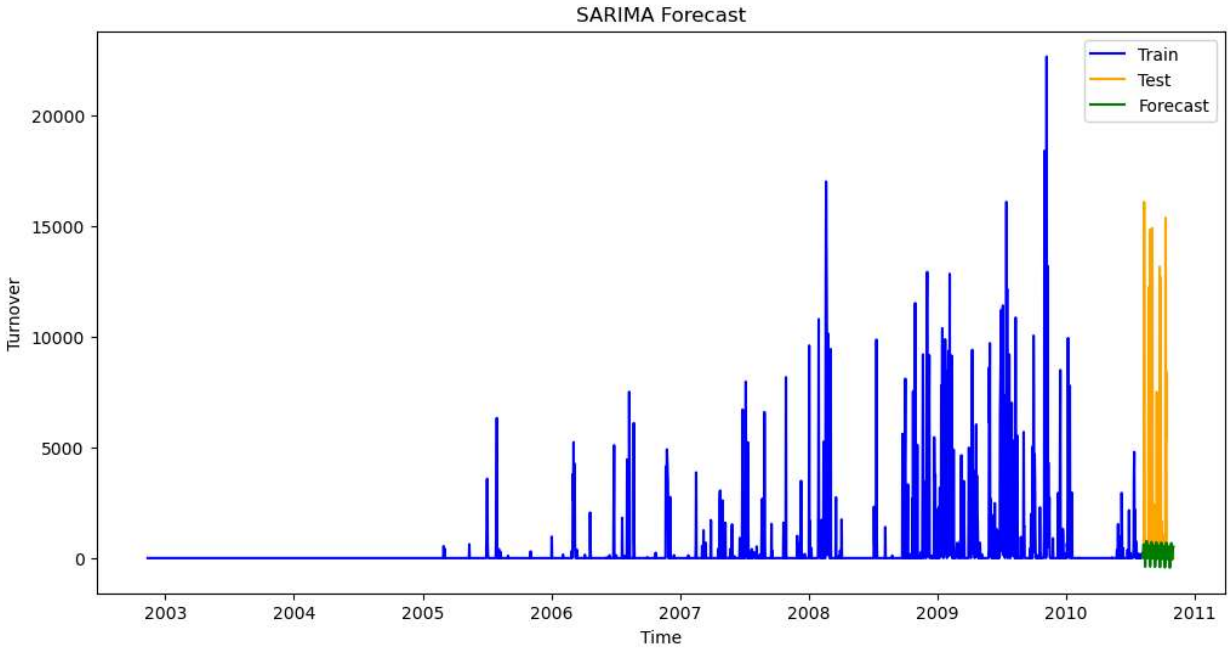

```
ARIMA Model Results (User 868583):
RMSE (ARIMA): 22.40285256300589
MAE (ARIMA): 22.396887663461186
==============================
User 868583:
SARIMAX RMSE : 102.5766159469979
SARIMAX MAE : 79.83388625079733
==============================
PG Detected right before: 2010-08-13 00:00:00
PG Detected right before: 2010-08-15 00:00:00
PG Detected right before: 2010-08-27 00:00:00
PG Detected right before: 2010-08-29 00:00:00
PG Detected right before: 2010-09-10 00:00:00
PG Detected right before: 2010-09-12 00:00:00
PG Detected right before: 2010-09-24 00:00:00
PG Detected right before: 2010-09-26 00:00:00
PG Detected right before: 2010-10-08 00:00:00
PG Detected right before: 2010-10-10 00:00:00
PG Detected right before: 2010-10-22 00:00:00
PG Detected right before: 2010-10-24 00:00:00
```

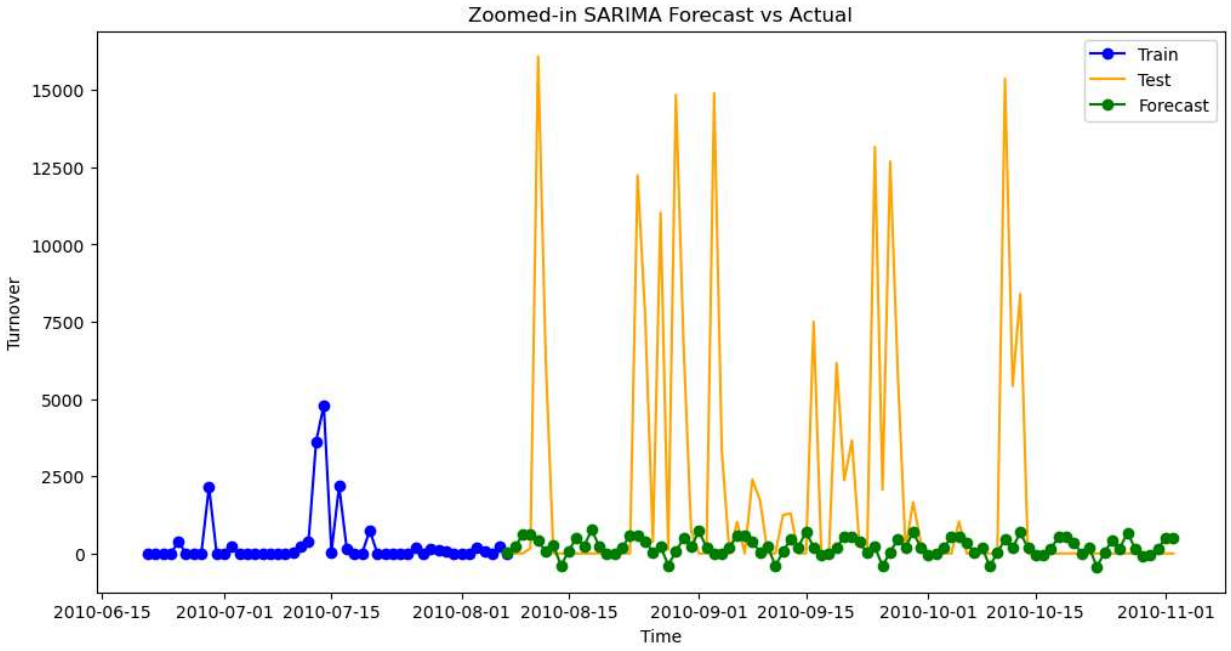

SARIMA Forecast
Train
Test
Forecast
Turnover
Time
10000
8000
6000
4000
2000
0
2003
2004
2005
2006
2007
2008
2009
2010
2011

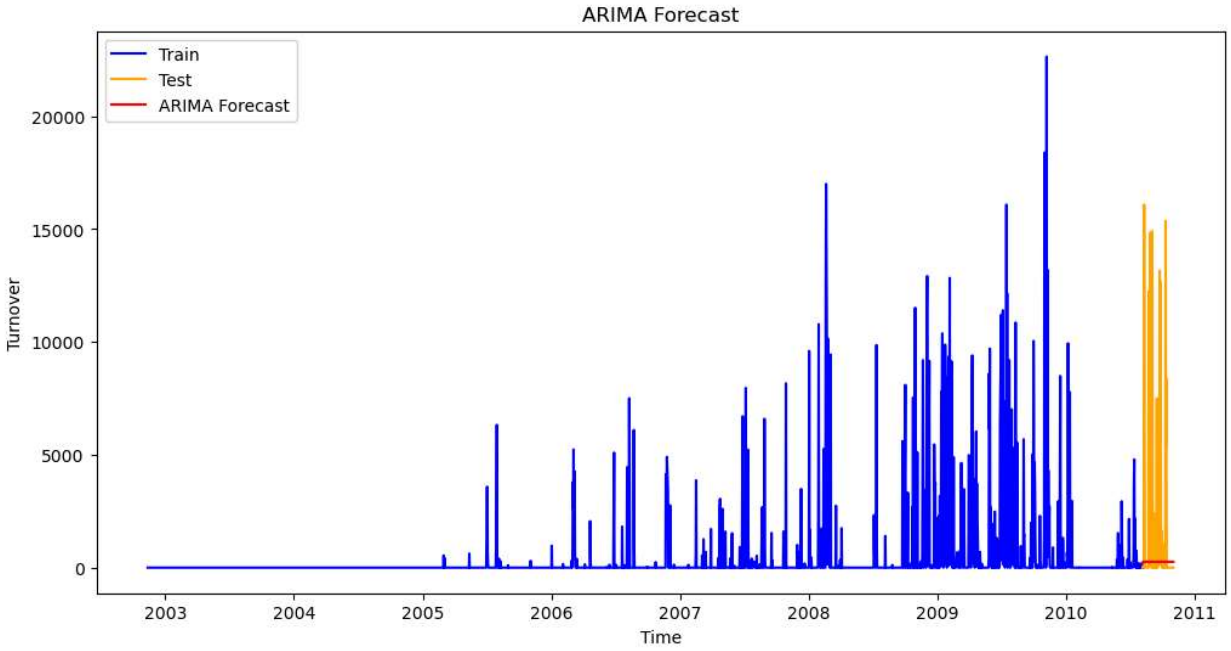

Zoomed-in SARIMA Forecast vs Actual
Train
Test
Forecast
Turnover
Time
5000
4000
3000
2000
1000
0
2010-06-15
2010-07-01
2010-07-15
2010-08-01
2010-08-15
2010-09-01
2010-09-15
2010-10-01
2010-10-15
2010-11-01



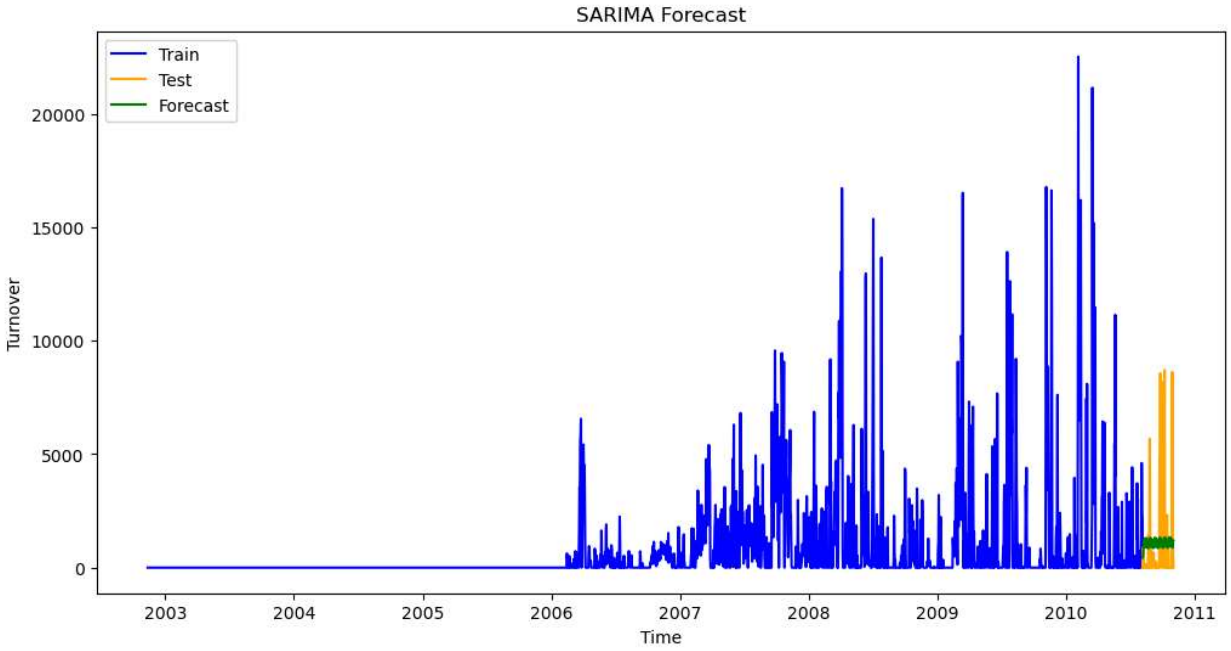

```
ARIMA Model Results (User 1175809):
RMSE (ARIMA): 784.7687886925128
MAE (ARIMA): 331.9608386066314
================================
User 1175809:
SARIMAX RMSE : 777.9760114331281
SARIMAX MAE : 336.02694007952886
================================
PG Detected right before: 2010-08-09 00:00:00
PG Detected right before: 2010-08-16 00:00:00
PG Detected right before: 2010-08-23 00:00:00
PG Detected right before: 2010-08-30 00:00:00
PG Detected right before: 2010-09-06 00:00:00
PG Detected right before: 2010-09-13 00:00:00
PG Detected right before: 2010-09-20 00:00:00
PG Detected right before: 2010-09-27 00:00:00
PG Detected right before: 2010-10-04 00:00:00
PG Detected right before: 2010-10-11 00:00:00
PG Detected right before: 2010-10-18 00:00:00
PG Detected right before: 2010-10-25 00:00:00
PG Detected right before: 2010-11-01 00:00:00
```

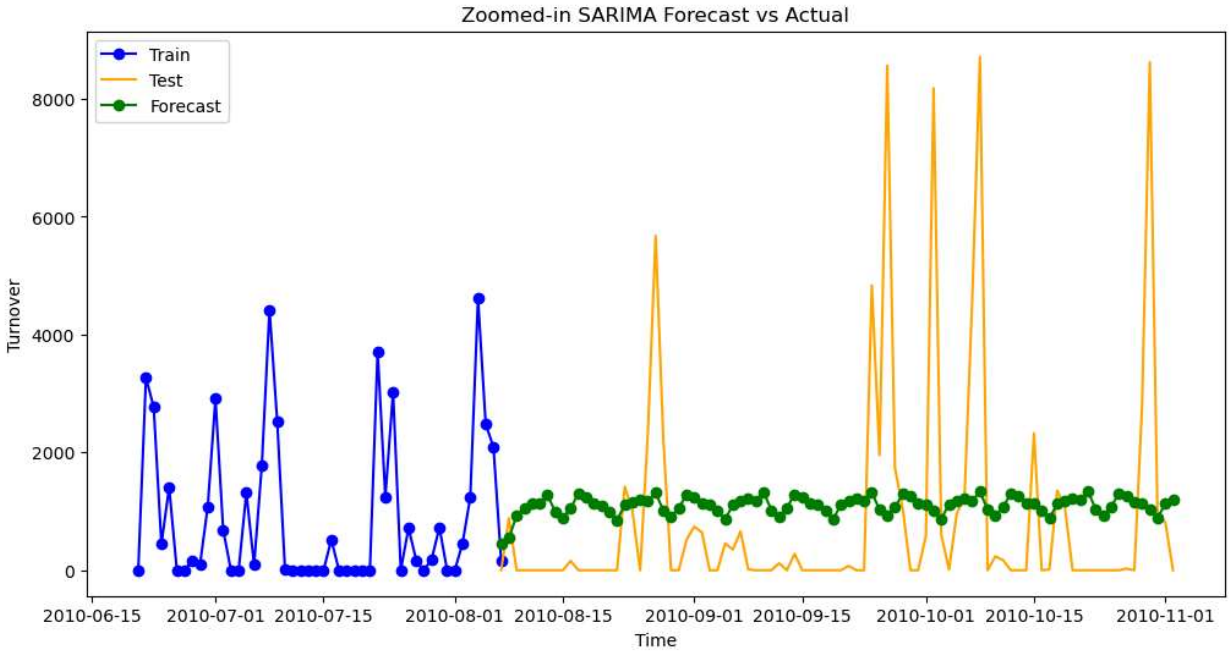

SARIMA Forecast
Train
Test
Forecast
Turnover
Time
0
5000
10000
15000
20000
2003
2004
2005
2006
2007
2008
2009
2010
2011

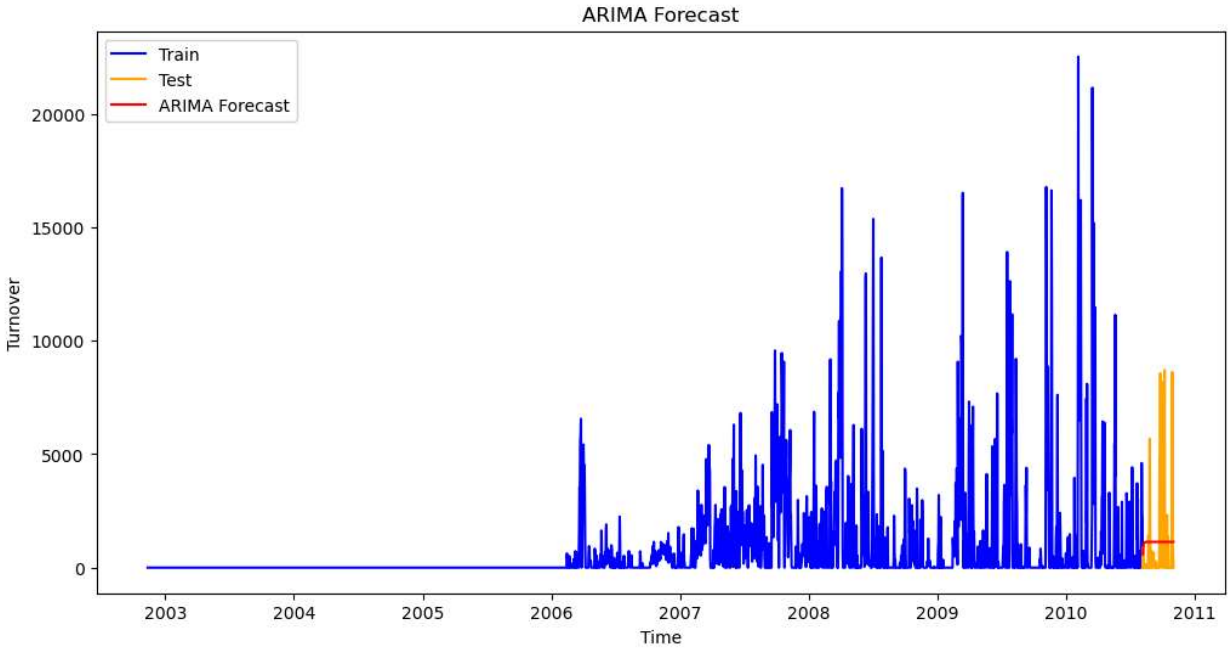

Zoomed-in SARIMA Forecast vs Actual
Train
Test
Forecast
Turnover
Time
0
2500
5000
7500
10000
12500
15000
2010-06-15
2010-07-01
2010-07-15
2010-08-01
2010-08-15
2010-09-01
2010-09-15
2010-10-01
2010-10-15
2010-11-01


ARIMA Forecast
Train
Test
ARIMA Forecast
Turnover
20000
15000
10000
5000
0
2003
2004
2005
2006
2007
2008
2009
2010
2011
Time

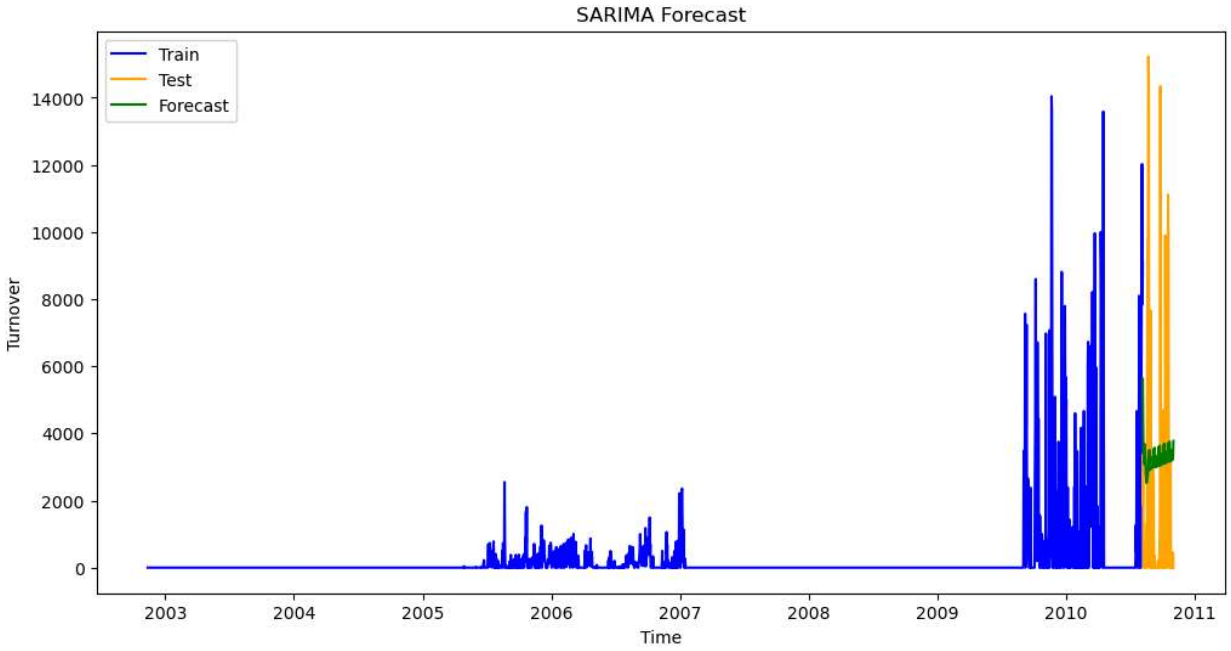
```
ARIMA Model Results (User 1411743):
RMSE (ARIMA): 4616.1579093677465
MAE (ARIMA): 2204.101215242136
==============================
User 1411743:
SARIMAX RMSE : 4613.733648122473
SARIMAX MAE : 2212.505857385308
==============================
PG Detected right before: 2010-08-08 00:00:00
PG Detected right before: 2010-08-12 00:00:00
PG Detected right before: 2010-08-14 00:00:00
PG Detected right before: 2010-08-15 00:00:00
PG Detected right before: 2010-08-17 00:00:00
PG Detected right before: 2010-08-21 00:00:00
PG Detected right before: 2010-08-22 00:00:00
PG Detected right before: 2010-08-26 00:00:00
PG Detected right before: 2010-08-28 00:00:00
PG Detected right before: 2010-08-29 00:00:00
PG Detected right before: 2010-08-31 00:00:00
PG Detected right before: 2010-09-04 00:00:00
PG Detected right before: 2010-09-05 00:00:00
PG Detected right before: 2010-09-09 00:00:00
PG Detected right before: 2010-09-11 00:00:00
PG Detected right before: 2010-09-12 00:00:00
PG Detected right before: 2010-09-14 00:00:00
PG Detected right before: 2010-09-18 00:00:00
PG Detected right before: 2010-09-19 00:00:00
PG Detected right before: 2010-09-23 00:00:00
PG Detected right before: 2010-09-25 00:00:00
PG Detected right before: 2010-09-26 00:00:00
PG Detected right before: 2010-09-28 00:00:00
PG Detected right before: 2010-10-02 00:00:00
PG Detected right before: 2010-10-03 00:00:00
PG Detected right before: 2010-10-07 00:00:00
PG Detected right before: 2010-10-09 00:00:00
PG Detected right before: 2010-10-10 00:00:00
PG Detected right before: 2010-10-12 00:00:00
PG Detected right before: 2010-10-16 00:00:00
PG Detected right before: 2010-10-17 00:00:00
PG Detected right before: 2010-10-21 00:00:00
PG Detected right before: 2010-10-23 00:00:00
PG Detected right before: 2010-10-24 00:00:00
PG Detected right before: 2010-10-26 00:00:00
PG Detected right before: 2010-10-30 00:00:00
PG Detected right before: 2010-10-31 00:00:00
```

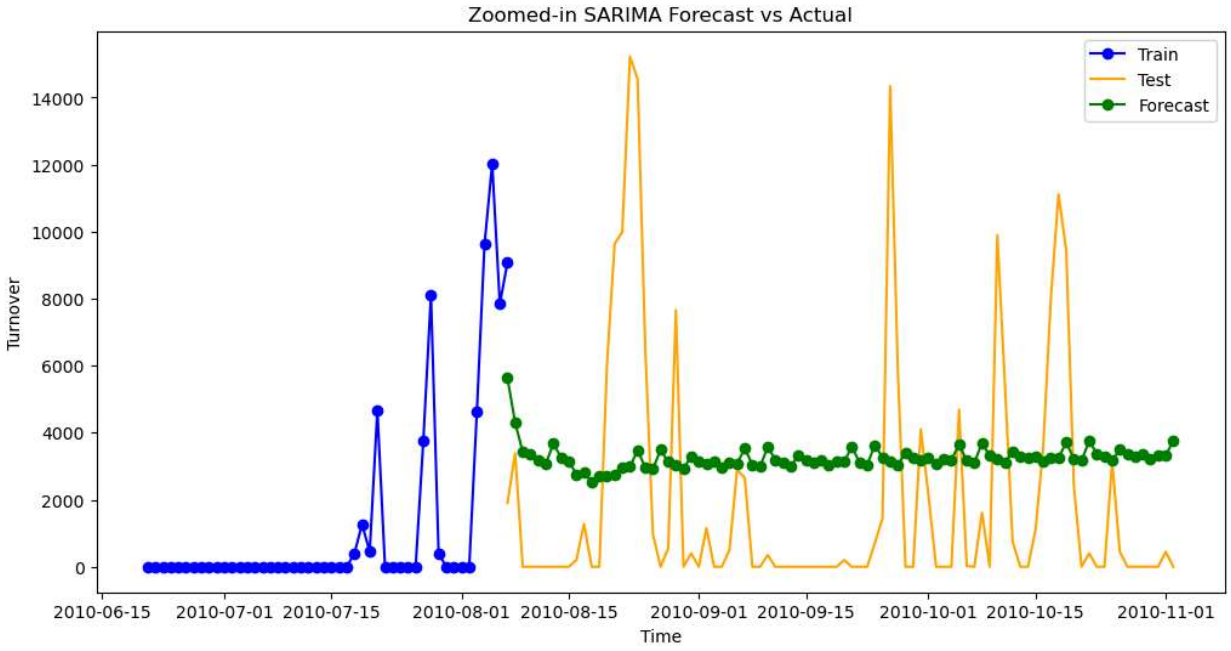

SARIMA Forecast
Train
Test
Forecast
Turnover
Time
0
5000
10000
15000
20000
2003
2004
2005
2006
2007
2008
2009
2010
2011

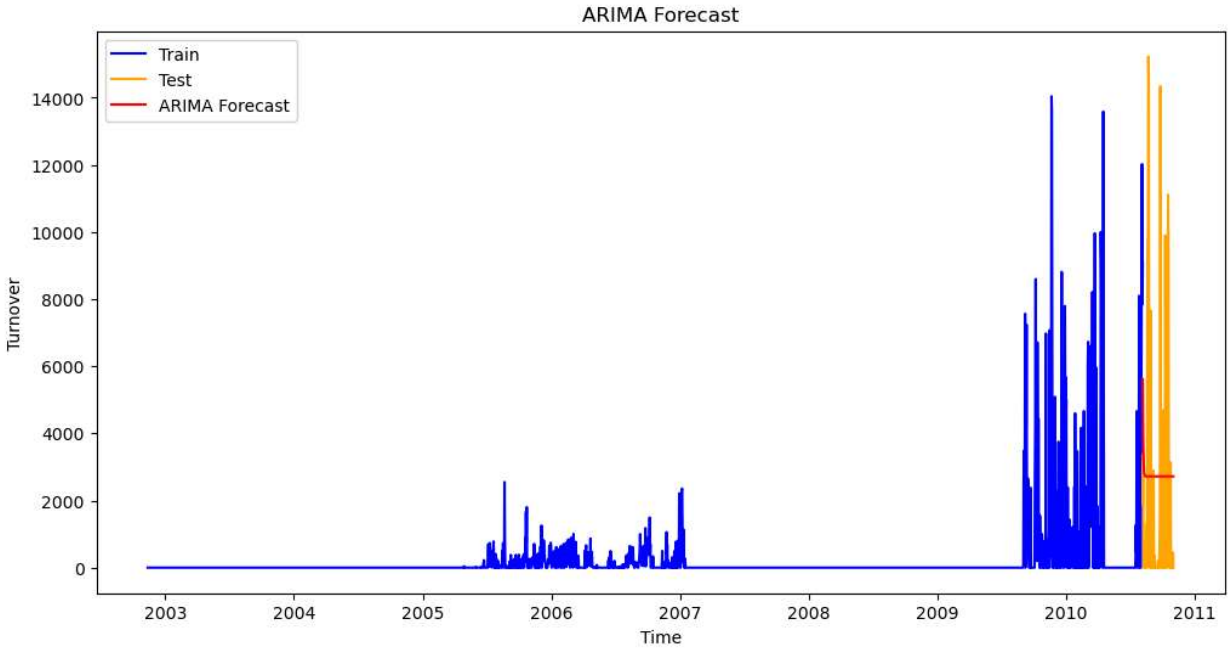

Zoomed-in SARIMA Forecast vs Actual
Train
Test
Forecast
Turnover
Time
0
2000
4000
6000
8000
2010-06-15
2010-07-01
2010-07-15
2010-08-01
2010-08-15
2010-09-01
2010-09-15
2010-10-01
2010-10-15
2010-11-01



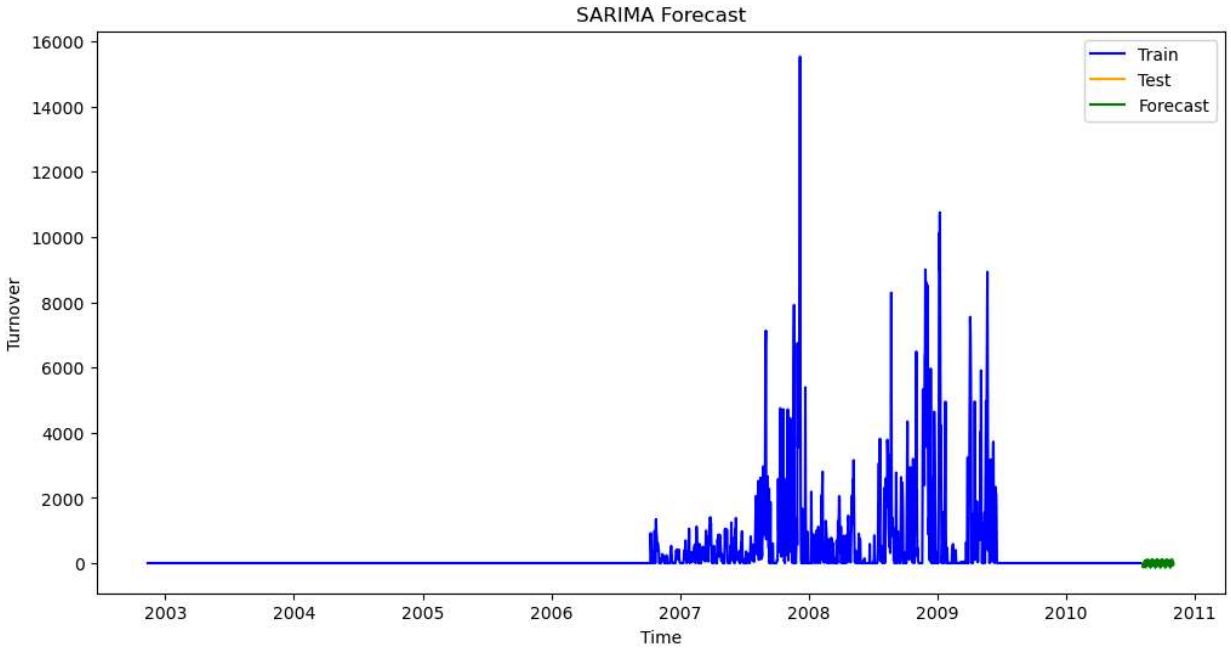

```
ARIMA Model Results (User 1457496):
RMSE (ARIMA): 1978.891627932139
MAE (ARIMA): 1324.9782438380369
==============================
User 1457496:
SARIMAX RMSE : 1977.6625563070231
SARIMAX MAE : 1313.9087263358097
==============================
PG Detected right before: 2010-08-08 00:00:00
PG Detected right before: 2010-08-12 00:00:00
PG Detected right before: 2010-08-15 00:00:00
PG Detected right before: 2010-08-16 00:00:00
PG Detected right before: 2010-08-22 00:00:00
PG Detected right before: 2010-08-30 00:00:00
PG Detected right before: 2010-09-05 00:00:00
PG Detected right before: 2010-09-13 00:00:00
PG Detected right before: 2010-09-19 00:00:00
PG Detected right before: 2010-09-27 00:00:00
PG Detected right before: 2010-10-03 00:00:00
PG Detected right before: 2010-10-11 00:00:00
PG Detected right before: 2010-10-17 00:00:00
PG Detected right before: 2010-10-25 00:00:00
PG Detected right before: 2010-10-31 00:00:00
```

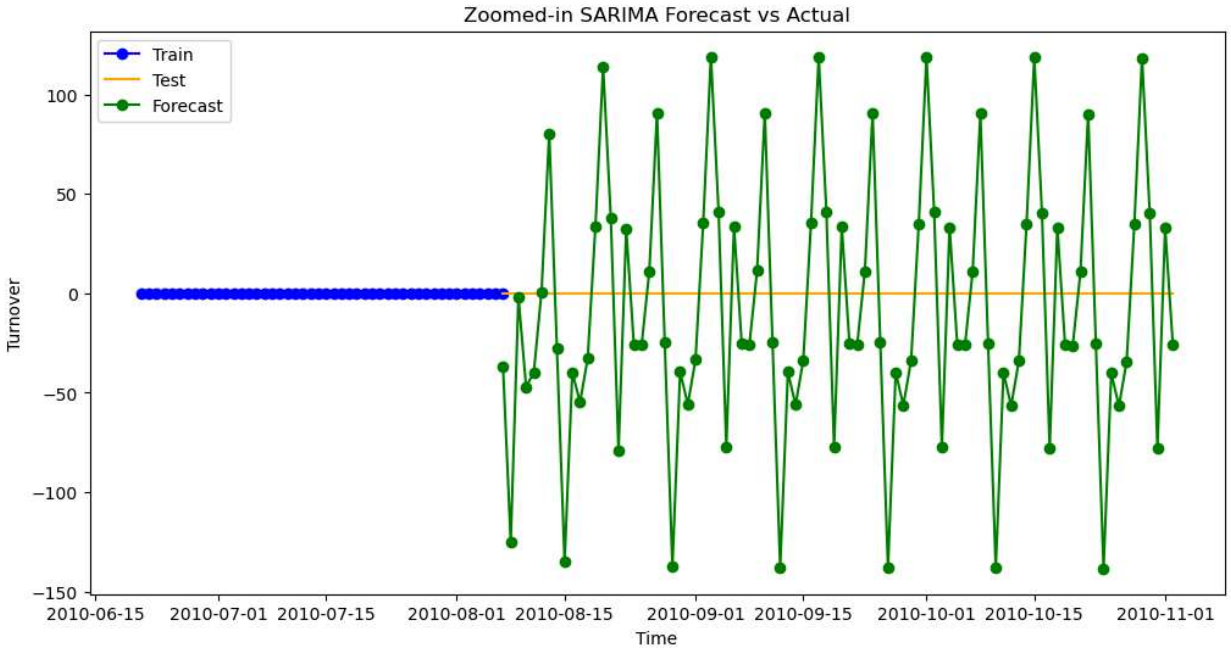

SARIMA Forecast
Train
Test
Forecast
Turnover
Time
0
2000
4000
6000
8000
10000
12000
14000
2003
2004
2005
2006
2007
2008
2009
2010
2011

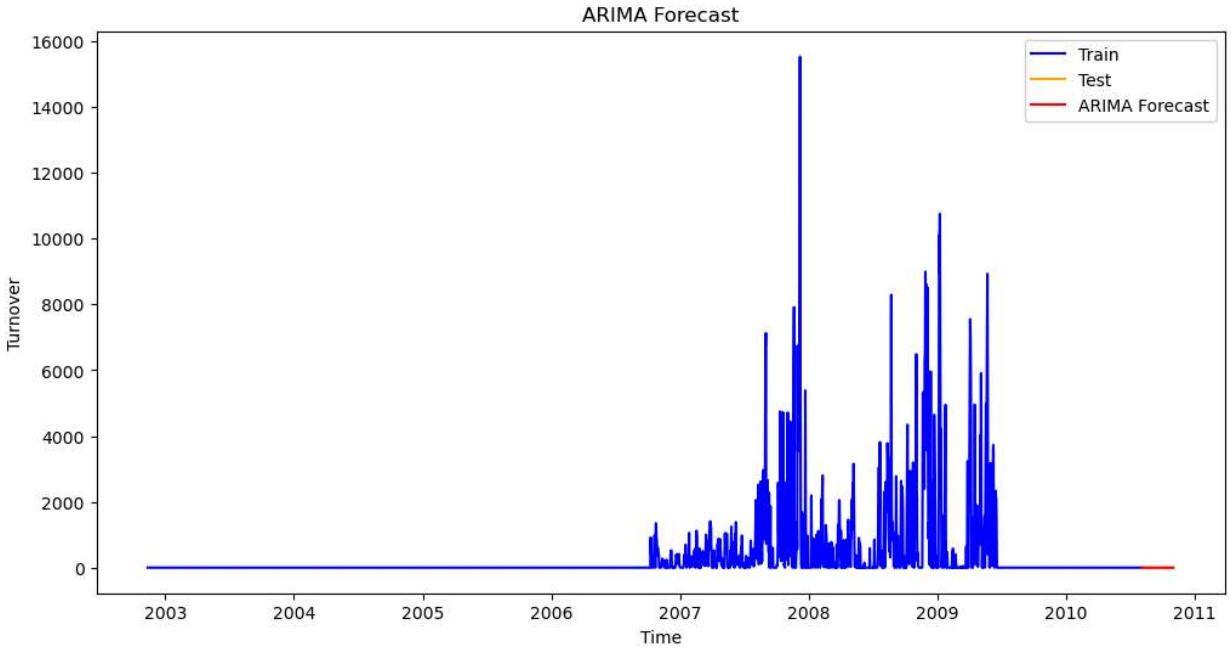

Zoomed-in SARIMA Forecast vs Actual
Train
Test
Forecast
Turnover
Time
0
2000
4000
6000
8000
10000
12000
14000
2010-06-15
2010-07-01
2010-07-15
2010-08-01
2010-08-15
2010-09-01
2010-09-15
2010-10-01
2010-10-15
2010-11-01




```
ARIMA Model Results (User 1486136):
RMSE (ARIMA): 3753.6733323996104
MAE (ARIMA): 3049.424238509906
==============================
User 1486136:
SARIMAX RMSE : 3891.2559967302295
SARIMAX MAE : 3338.0576744058267
==============================
PG Detected right before: 2010-08-12 00:00:00
PG Detected right before: 2010-08-18 00:00:00
PG Detected right before: 2010-08-21 00:00:00
PG Detected right before: 2010-08-23 00:00:00
PG Detected right before: 2010-08-26 00:00:00
PG Detected right before: 2010-08-30 00:00:00
PG Detected right before: 2010-09-06 00:00:00
PG Detected right before: 2010-09-09 00:00:00
PG Detected right before: 2010-09-13 00:00:00
PG Detected right before: 2010-09-20 00:00:00
PG Detected right before: 2010-09-23 00:00:00
PG Detected right before: 2010-09-27 00:00:00
PG Detected right before: 2010-10-04 00:00:00
PG Detected right before: 2010-10-07 00:00:00
PG Detected right before: 2010-10-11 00:00:00
PG Detected right before: 2010-10-18 00:00:00
PG Detected right before: 2010-10-21 00:00:00
PG Detected right before: 2010-10-25 00:00:00
PG Detected right before: 2010-11-01 00:00:00
```

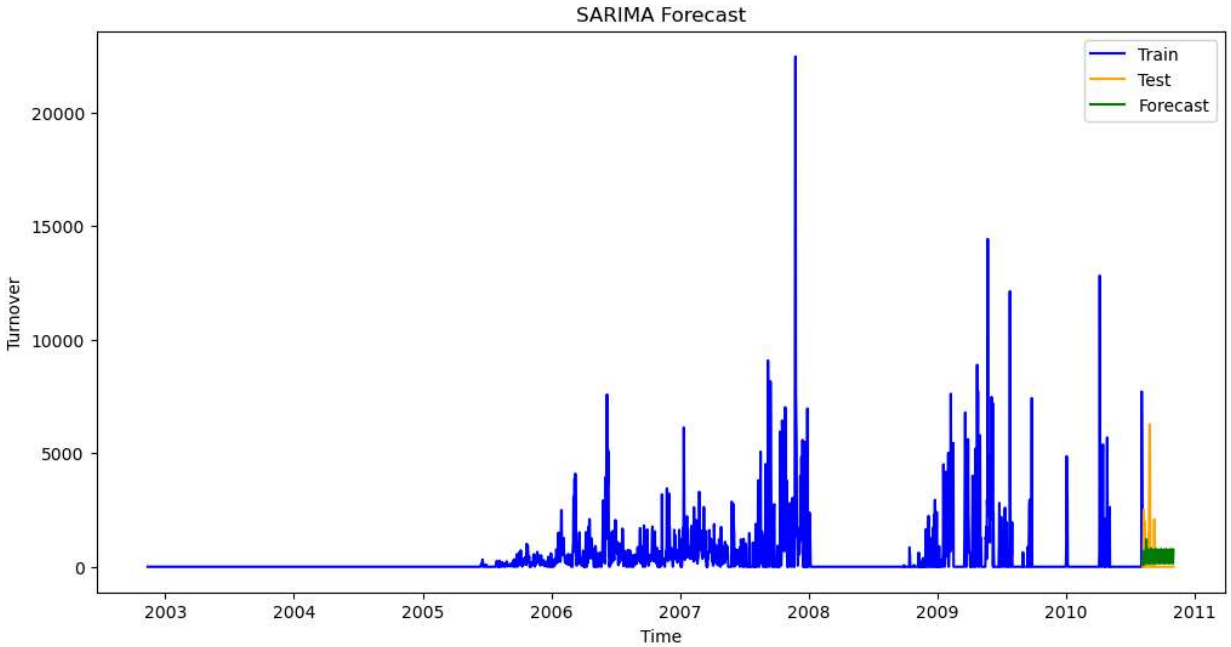

SARIMA Forecast
Train
Test
Forecast
Turnover
Time
16000
14000
12000
10000
8000
6000
4000
2000
0
2003
2004
2005
2006
2007
2008
2009
2010
2011

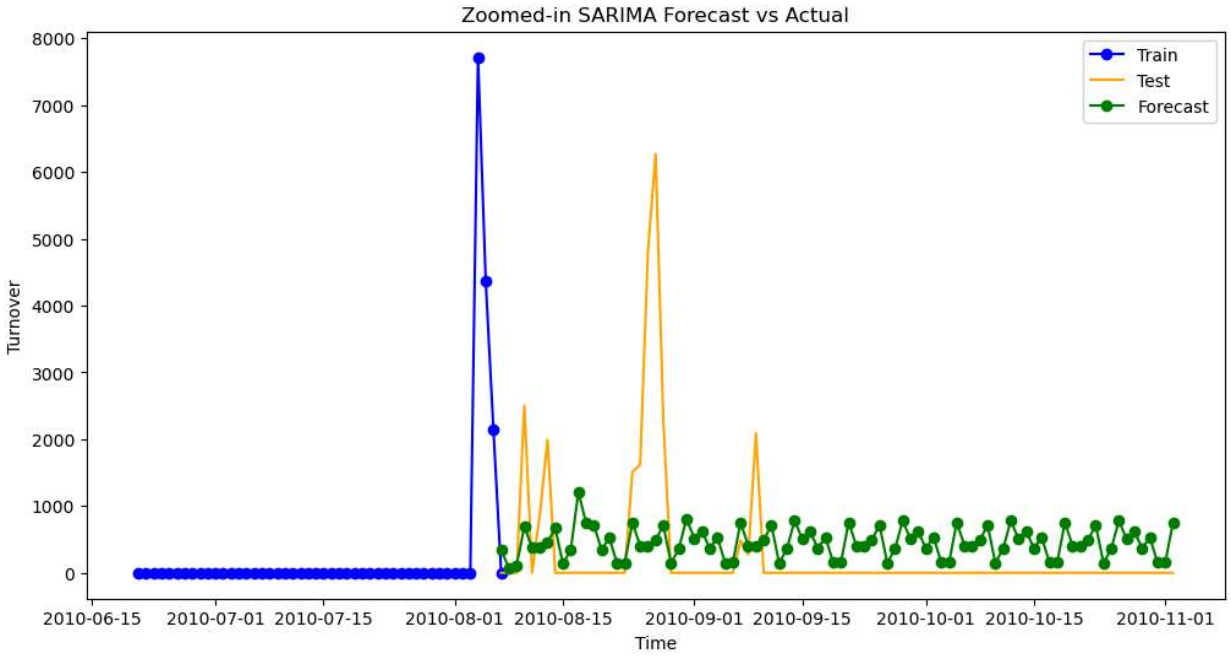

Zoomed-in SARIMA Forecast vs Actual
Train
Test
Forecast
Turnover
Time
100
50
0
−50
−100
−150
2010-06-15
2010-07-01
2010-07-15
2010-08-01
2010-08-15
2010-09-01
2010-09-15
2010-10-01
2010-10-15
2010-11-01

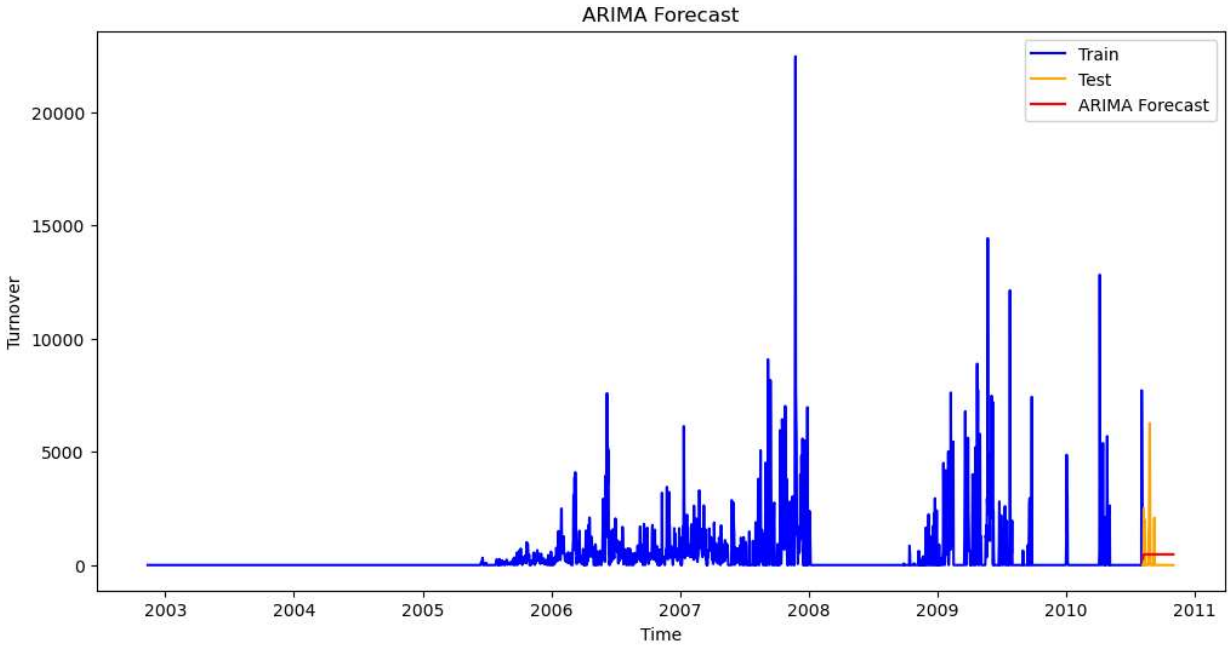



```
ARIMA Model Results (User 1662632):
RMSE (ARIMA): 0.34753857826167966
MAE (ARIMA): 0.3459736756983271
================================
User 1662632:
SARIMAX RMSE : 65.0293645083209
SARIMAX MAE : 53.443114830836
================================
```

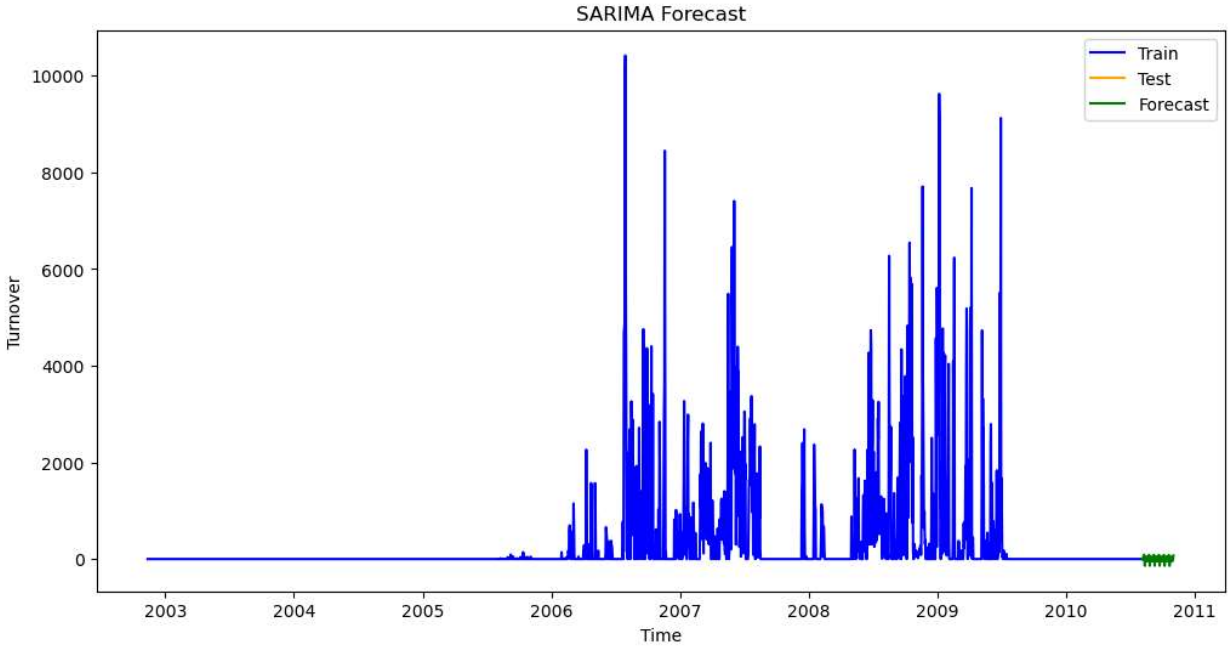
```
C:\Users\abhiv\AppData\Local\anaconda3\Lib\site-packages\statsmodels\base\model.py:604: ConvergenceWarning: Maximum Likelihood optimization failed to converge. Check mle_retvals
  warnings.warn("Maximum Likelihood optimization failed to "
```

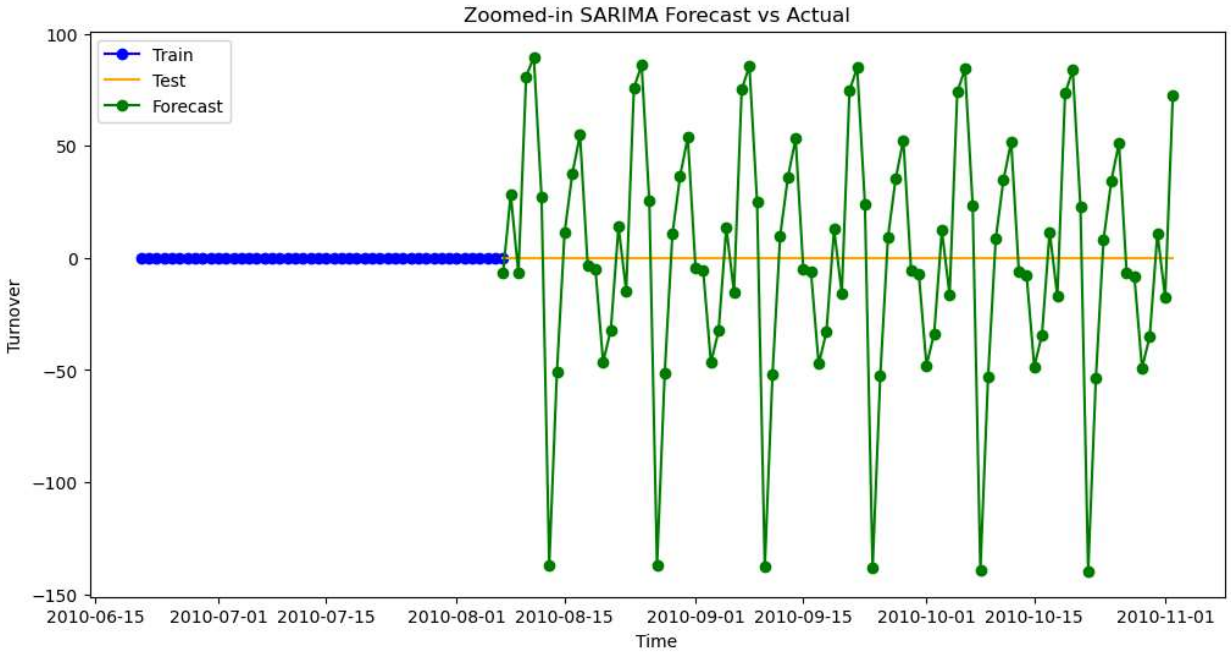


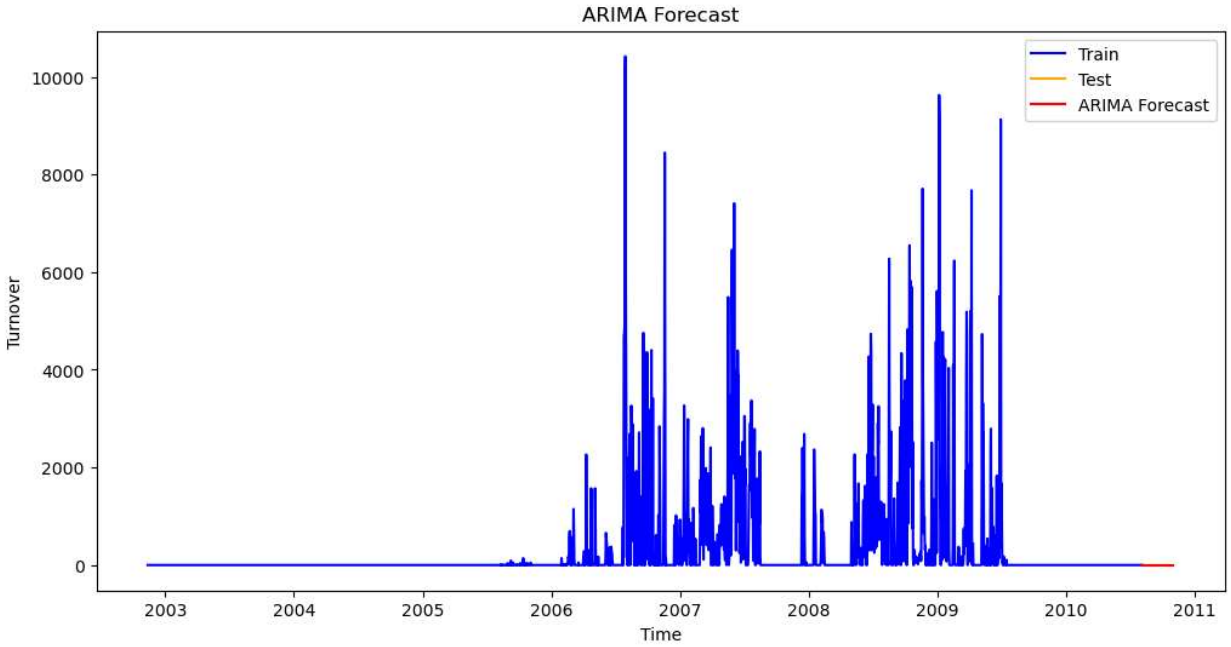

Zoomed-in SARIMA Forecast vs Actual
Train
Test
Forecast
Turnover
Time
8000
7000
6000
5000
4000
3000
2000
1000
0
2010-06-15
2010-07-01
2010-07-15
2010-08-01
2010-08-15
2010-09-01
2010-09-15
2010-10-01
2010-10-15
2010-11-01


ARIMA Forecast
Train
Test
ARIMA Forecast
Turnover
Time
20000
15000
10000
5000
0
2003
2004
2005
2006
2007
2008
2009
2010
2011

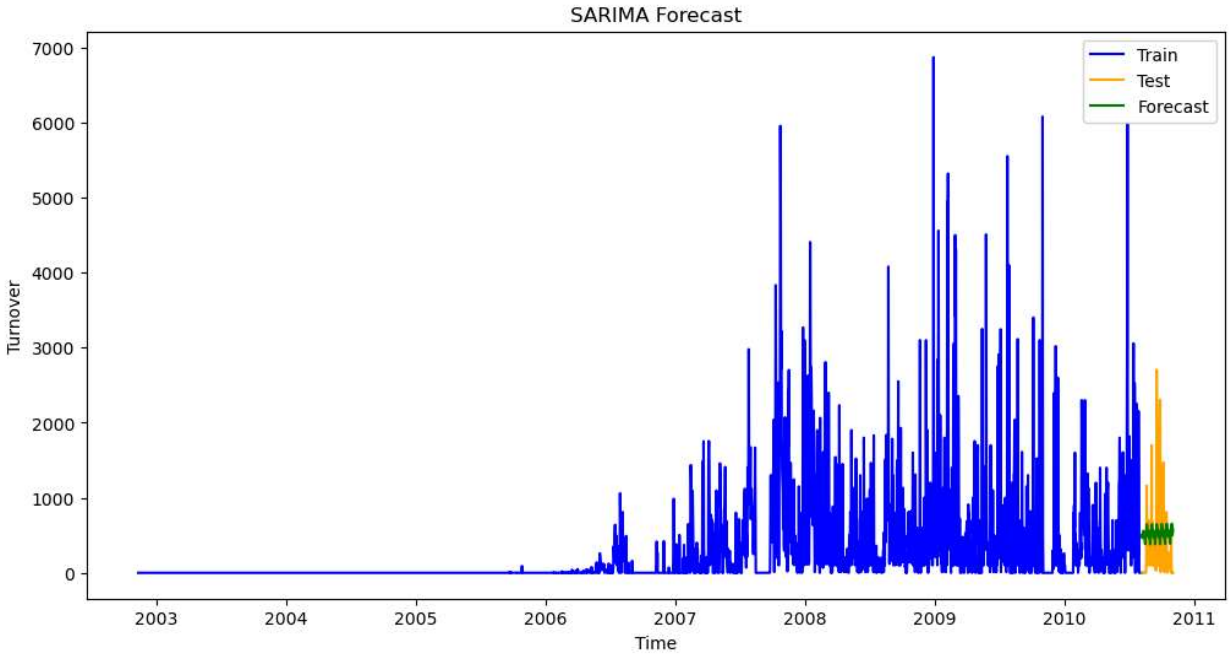
```
ARIMA Model Results (User 1679490):
RMSE (ARIMA): 978.1364539081785
MAE (ARIMA): 636.1494596739383
==============================
User 1679490:
SARIMAX RMSE : 985.5062523969005
SARIMAX MAE : 612.500708985471
==============================
PG Detected right before: 2010-08-09 00:00:00
PG Detected right before: 2010-08-13 00:00:00
PG Detected right before: 2010-08-15 00:00:00
PG Detected right before: 2010-08-16 00:00:00
PG Detected right before: 2010-08-20 00:00:00
PG Detected right before: 2010-08-23 00:00:00
PG Detected right before: 2010-08-27 00:00:00
PG Detected right before: 2010-08-29 00:00:00
PG Detected right before: 2010-08-30 00:00:00
PG Detected right before: 2010-09-03 00:00:00
PG Detected right before: 2010-09-06 00:00:00
PG Detected right before: 2010-09-10 00:00:00
PG Detected right before: 2010-09-12 00:00:00
PG Detected right before: 2010-09-13 00:00:00
PG Detected right before: 2010-09-17 00:00:00
PG Detected right before: 2010-09-20 00:00:00
PG Detected right before: 2010-09-24 00:00:00
PG Detected right before: 2010-09-26 00:00:00
PG Detected right before: 2010-09-27 00:00:00
PG Detected right before: 2010-10-01 00:00:00
PG Detected right before: 2010-10-04 00:00:00
PG Detected right before: 2010-10-08 00:00:00
PG Detected right before: 2010-10-10 00:00:00
PG Detected right before: 2010-10-11 00:00:00
PG Detected right before: 2010-10-15 00:00:00
PG Detected right before: 2010-10-18 00:00:00
PG Detected right before: 2010-10-22 00:00:00
PG Detected right before: 2010-10-24 00:00:00
PG Detected right before: 2010-10-25 00:00:00
PG Detected right before: 2010-10-29 00:00:00
PG Detected right before: 2010-11-01 00:00:00
```

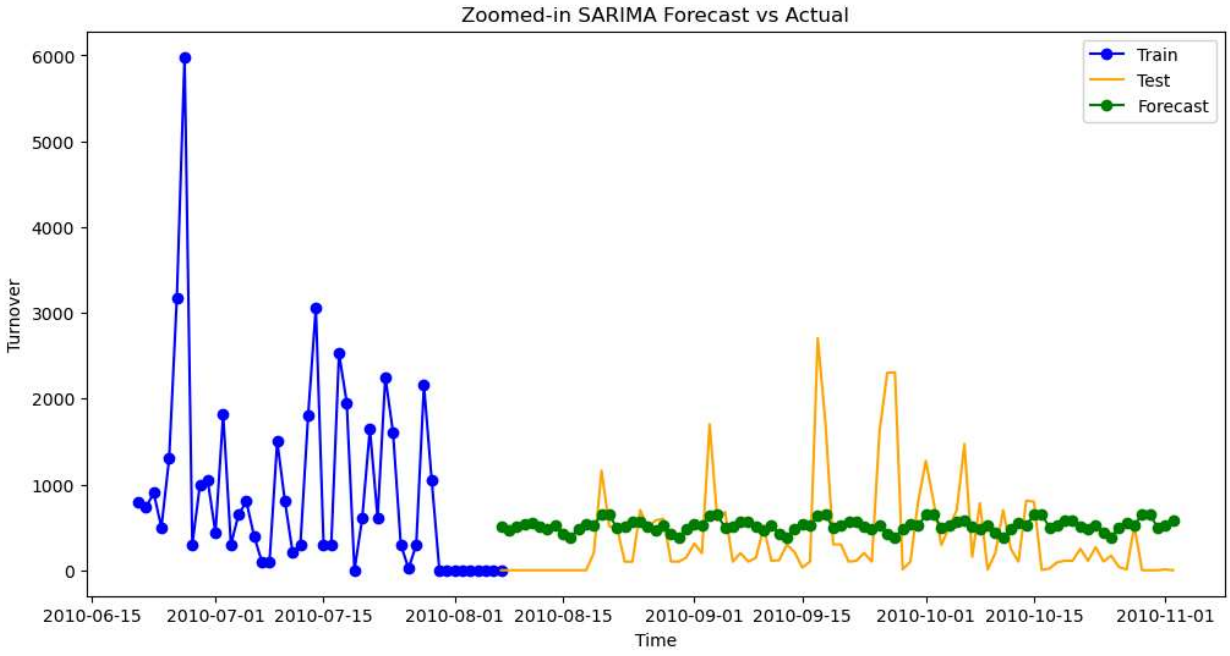


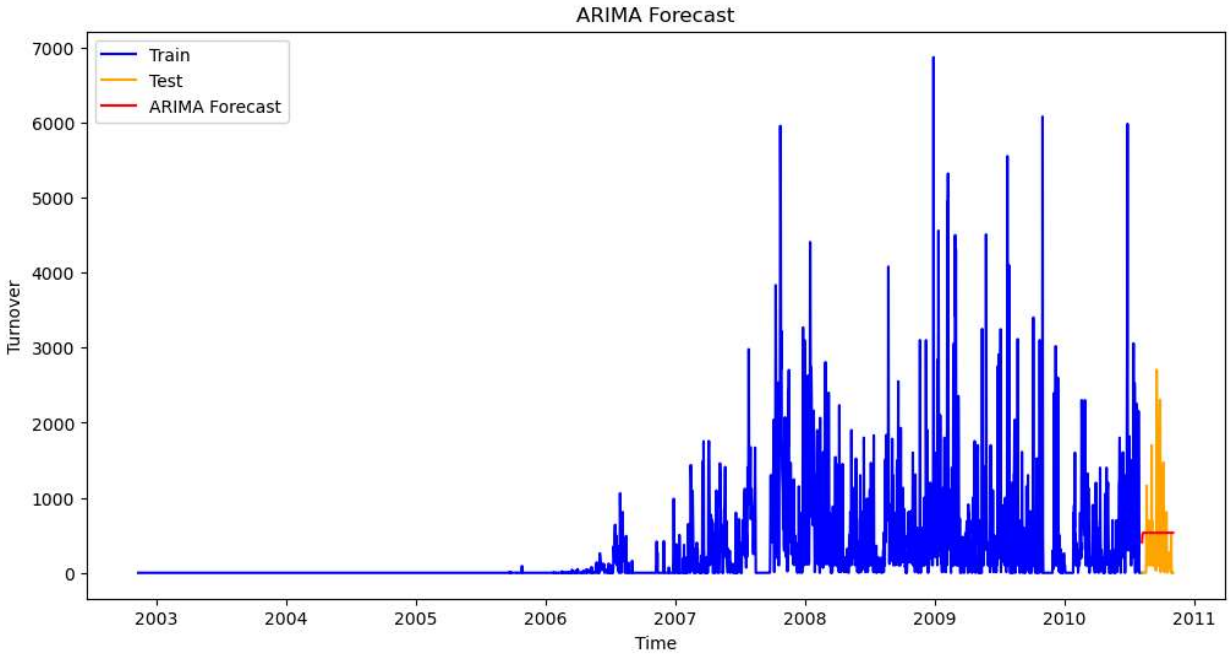




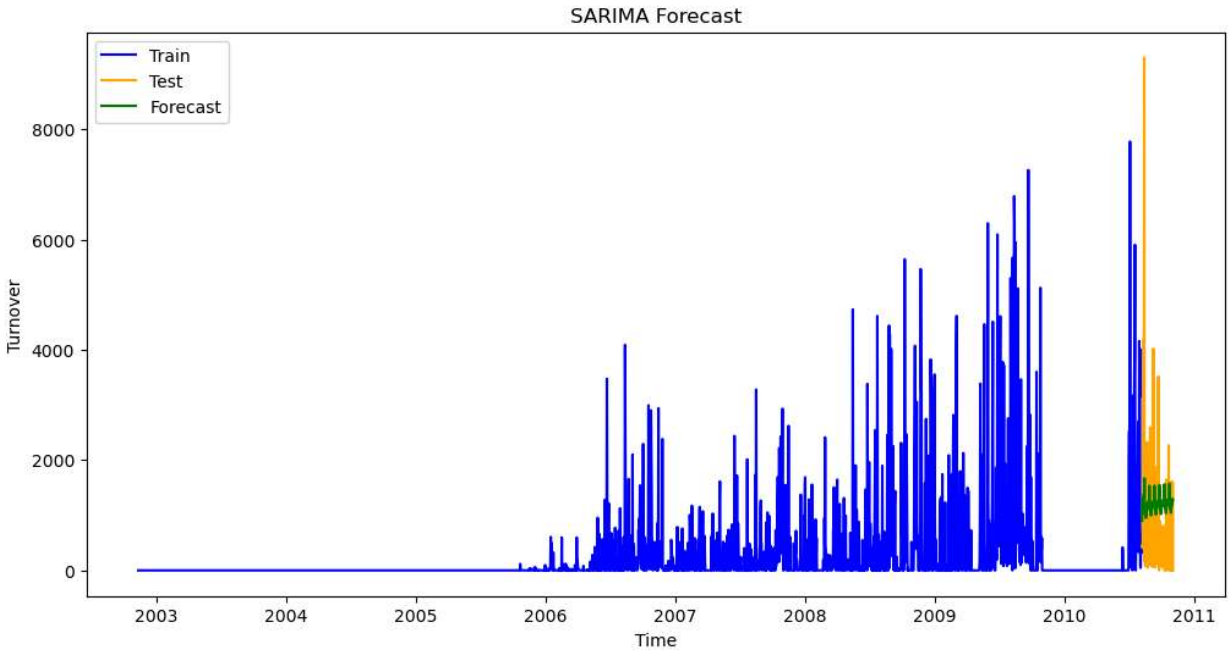

```
ARIMA Model Results (User 1790848):
RMSE (ARIMA): 0.0012906643653054297
MAE (ARIMA): 0.0012876457490887852
==============================
User 1790848:
SARIMAX RMSE : 55.19393943266972
SARIMAX MAE : 41.99327265944823
==============================
```

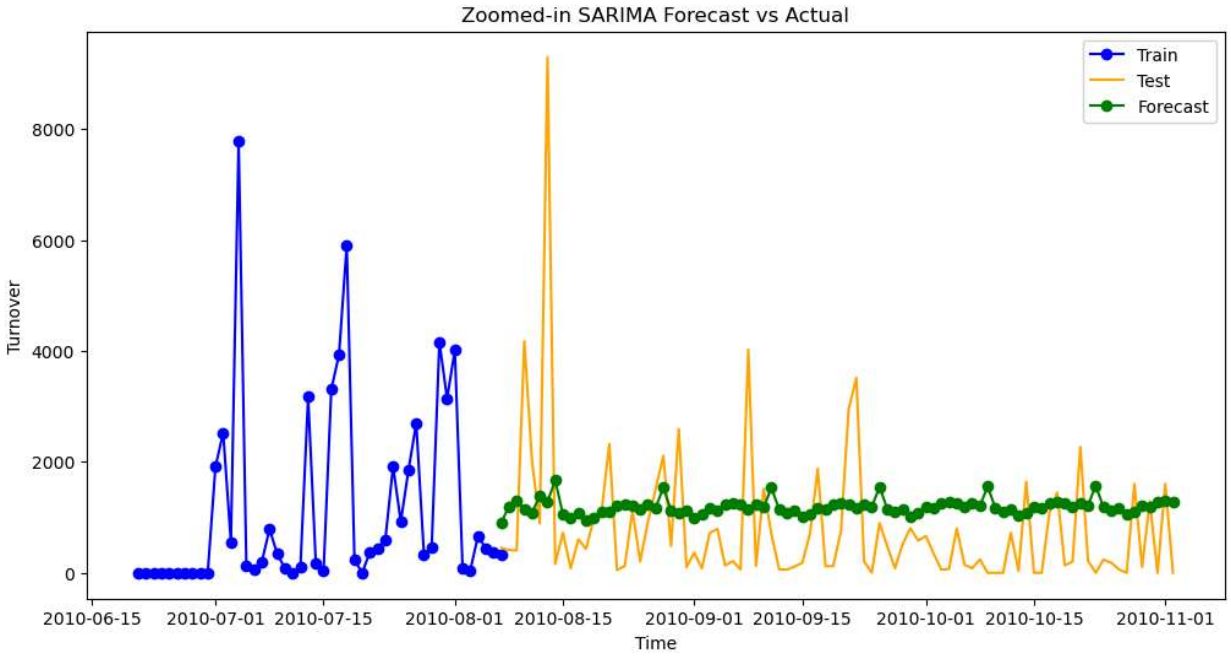

SARIMA Forecast
Train
Test
Forecast
Turnover
Time
0
1000
2000
3000
4000
5000
6000
7000
2003
2004
2005
2006
2007
2008
2009
2010
2011

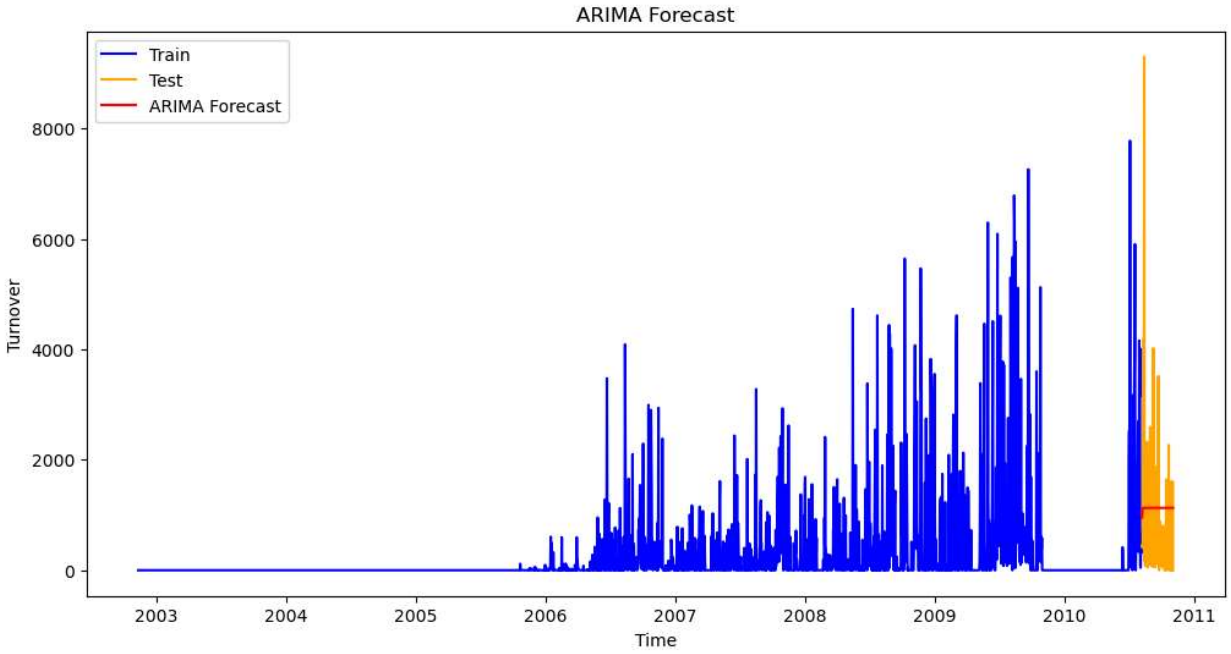

Zoomed-in SARIMA Forecast vs Actual
Train
Test
Forecast
Turnover
Time
0
1000
2000
3000
4000
5000
6000
2010-06-15
2010-07-01
2010-07-15
2010-08-01
2010-08-15
2010-09-01
2010-09-15
2010-10-01
2010-10-15
2010-11-01




```
ARIMA Model Results (User 1921204):
RMSE (ARIMA): 573.9839527146021
MAE (ARIMA): 453.59999894681533
==============================
User 1921204:
SARIMAX RMSE : 564.2913244910224
SARIMAX MAE : 437.39373955004817
==============================
```

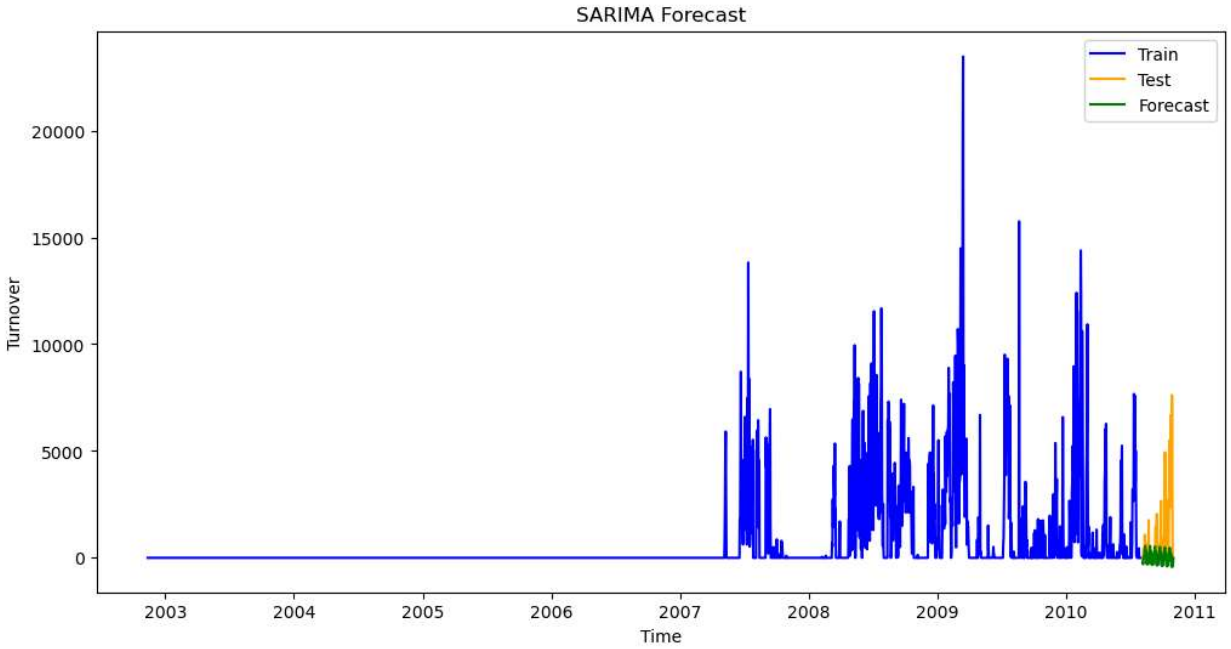


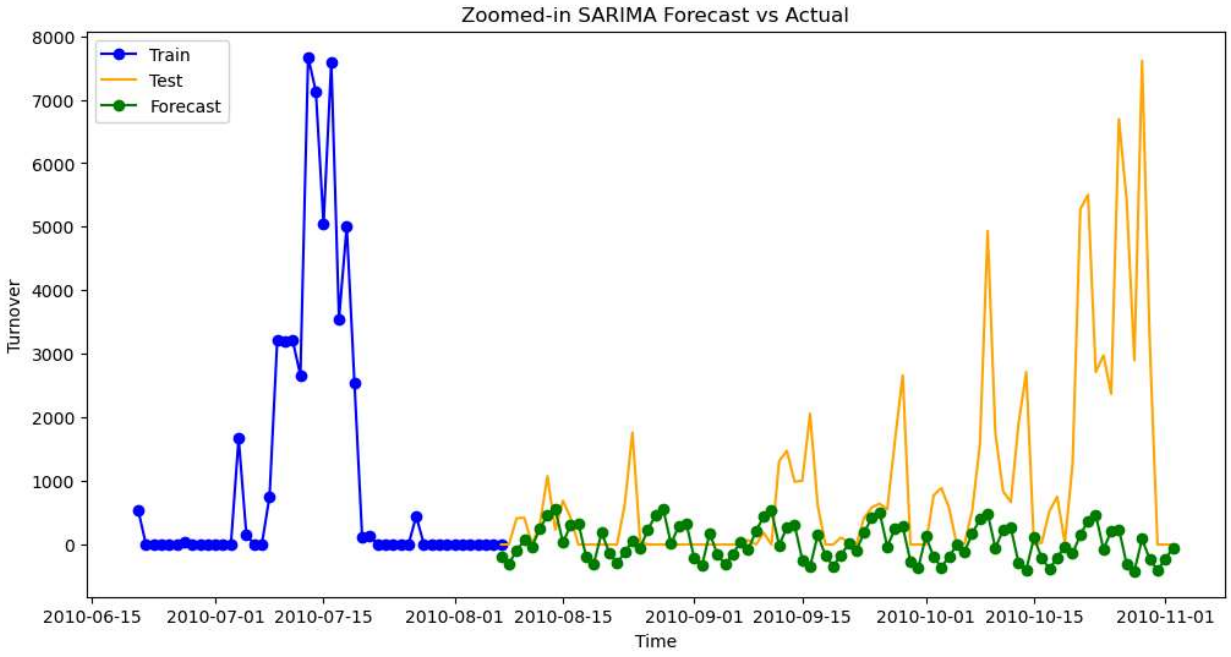


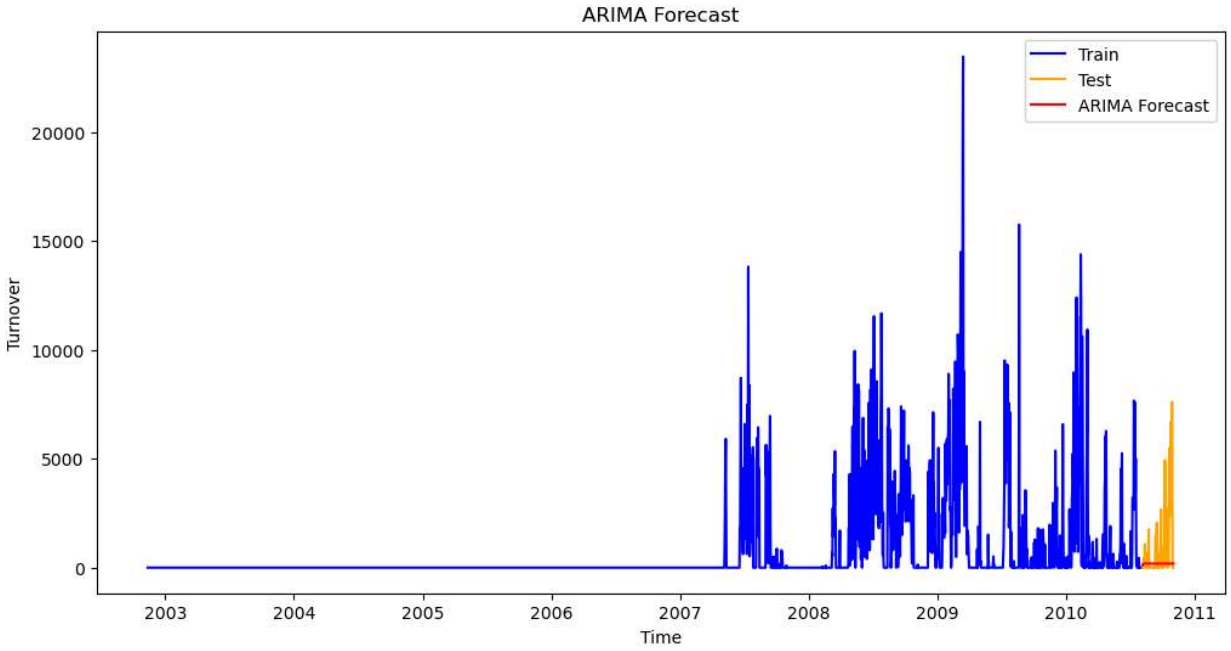



```
ARIMA Model Results (User 2070894):
RMSE (ARIMA): 1318.9026564972567
MAE (ARIMA): 939.9894409160116
==============================
User 2070894:
SARIMAX RMSE : 1338.5668002348875
SARIMAX MAE : 975.8631625342256
==============================
PG Detected right before: 2010-08-11 00:00:00
PG Detected right before: 2010-08-13 00:00:00
PG Detected right before: 2010-08-27 00:00:00
PG Detected right before: 2010-09-10 00:00:00
PG Detected right before: 2010-09-24 00:00:00
PG Detected right before: 2010-10-08 00:00:00
PG Detected right before: 2010-10-22 00:00:00
```

```
C:\Users\abhiv\AppData\Local\anaconda3\Lib\site-packages\statsmodels\base\model.py:604: ConvergenceWarning: Maximum Likelihood optimization failed to converge. Check mle_retvals
  warnings.warn("Maximum Likelihood optimization failed to "
```






ARIMA Forecast
Train
Test
ARIMA Forecast
Turnover
20000
15000
10000
5000
0
2003
2004
2005
2006
2007
2008
2009
2010
2011
Time

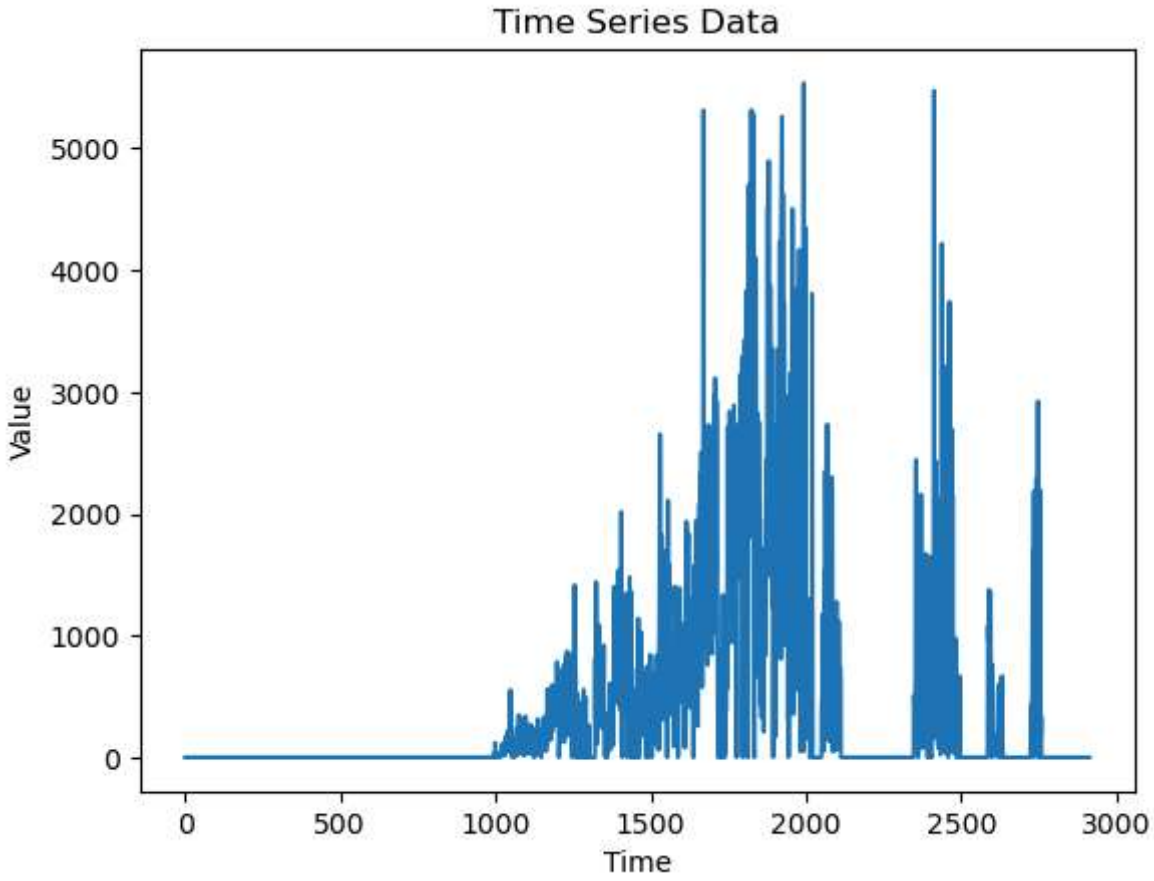
```
ARIMA Model Results (User 4754125):
RMSE (ARIMA): 1771.2240395544623
MAE (ARIMA): 946.1813865979672
==============================
User 4754125:
SARIMAX RMSE : 1841.9634368882164
SARIMAX MAE : 1036.2222142263788
==============================
PG Detected right before: 2010-08-08 00:00:00
PG Detected right before: 2010-08-09 00:00:00
PG Detected right before: 2010-08-11 00:00:00
PG Detected right before: 2010-08-12 00:00:00
PG Detected right before: 2010-08-15 00:00:00
PG Detected right before: 2010-08-19 00:00:00
PG Detected right before: 2010-08-22 00:00:00
PG Detected right before: 2010-08-23 00:00:00
PG Detected right before: 2010-08-25 00:00:00
PG Detected right before: 2010-08-26 00:00:00
PG Detected right before: 2010-08-29 00:00:00
PG Detected right before: 2010-09-02 00:00:00
PG Detected right before: 2010-09-05 00:00:00
PG Detected right before: 2010-09-06 00:00:00
PG Detected right before: 2010-09-08 00:00:00
PG Detected right before: 2010-09-09 00:00:00
PG Detected right before: 2010-09-12 00:00:00
PG Detected right before: 2010-09-16 00:00:00
PG Detected right before: 2010-09-19 00:00:00
PG Detected right before: 2010-09-20 00:00:00
PG Detected right before: 2010-09-22 00:00:00
PG Detected right before: 2010-09-23 00:00:00
PG Detected right before: 2010-09-26 00:00:00
PG Detected right before: 2010-09-30 00:00:00
PG Detected right before: 2010-10-03 00:00:00
PG Detected right before: 2010-10-04 00:00:00
PG Detected right before: 2010-10-06 00:00:00
PG Detected right before: 2010-10-07 00:00:00
PG Detected right before: 2010-10-10 00:00:00
PG Detected right before: 2010-10-14 00:00:00
PG Detected right before: 2010-10-17 00:00:00
PG Detected right before: 2010-10-18 00:00:00
PG Detected right before: 2010-10-20 00:00:00
PG Detected right before: 2010-10-21 00:00:00
PG Detected right before: 2010-10-24 00:00:00
PG Detected right before: 2010-10-28 00:00:00
PG Detected right before: 2010-10-31 00:00:00
PG Detected right before: 2010-11-01 00:00:00
```

LSTM for non stationary users

In [51]:
```
#arimausers =
lstmusers=is_stationary_df[(is_stationary_df['is_stationary'] == 0) | (is_stationary_d
lstmusers=lstmusers.reset_index()
lstmusers['UserID']
```

Out[51]:
```
0    1776178
1    3852889
2    3904422
3    4371320
4    4412550
5    4495603
6    5308271
7    6158120
8    6239380
9    6985339
Name: UserID, dtype: int64
```

In [52]:
```
nonstationary_single_user_data=user_data_3d['Turnover'][user_data_3d.index==lstmusers[
nonstationary_single_user_data


x_single_user_turnover_ravel=nonstationary_single_user_data.values.ravel()
x_single_user_turnover_ravel


import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from statsmodels.tsa.stattools import adfuller


# Plot the time series to visualize it
plt.plot(x_single_user_turnover_ravel)
plt.xlabel('Time')
plt.ylabel('Value')
plt.title('Time Series Data')
plt.show()
```



GRID SEARCH FOR BEST PARAMS

In [53]:
```
# import numpy as np
# import pandas as pd
# import matplotlib.pyplot as plt
# from sklearn.preprocessing import MinMaxScaler
# from keras.models import Sequential
# from keras.layers import Dense, LSTM
# from keras.optimizers import Adam
# from sklearn.metrics import mean_squared_error

# # Function to create and train an LSTM model
# def create_lstm_model(units, learning_rate):
#     model = Sequential()
#     model.add(LSTM(units=units, return_sequences=True, input_shape=(x_train.shape[1]
#     model.add(LSTM(units=units, return_sequences=False))
#     model.add(Dense(units=25))
#     model.add(Dense(units=1))

#     optimizer = Adam(learning_rate=learning_rate)
#     model.compile(optimizer=optimizer, loss='mean_squared_error')

#     return model

# # Extract statistics from the data
# row_variance = nonstationary_single_user_data.var(axis=1, skipna=True).iloc[0]
# row_mean = nonstationary_single_user_data.mean(axis=1).iloc[0]
# row_std = nonstationary_single_user_data.std(axis=1).iloc[0]

# print(f'Mean: {row_mean}')
```

```
# print(f'Standard Deviation: {row_std}')
# print(f'Variance: {row_variance}')

# # Flatten the data
# nonstationary_single_user_data_ravel = nonstationary_single_user_data.values.ravel()

# test_weeks = 3

# # Set the length of training data
# training_data_monthly_len = len(nonstationary_single_user_data_ravel) - 7 * test_wee

# # Scale the data
# scaler = MinMaxScaler(feature_range=(0, 1))
# scaled_data = scaler.fit_transform(nonstationary_single_user_data_ravel.reshape(-1,
# train_data_monthly = scaled_data[0:int(training_data_monthly_len), :]

# # Prepare the training data
# x_train = []
# y_train = []

# for i in range(7 * test_weeks, len(train_data_monthly)):
#     x_train.append(train_data_monthly[i - 7 * test_weeks:i, 0])
#     y_train.append(train_data_monthly[i, 0])

# x_train, y_train = np.array(x_train), np.array(y_train)

# # Reshape the data
# x_train = np.reshape(x_train, (x_train.shape[0], x_train.shape[1], 1))

# # Define hyperparameters for tuning
# units_values = [64, 128, 256]
# learning_rate_values = [0.01, 0.001, 0.0001]

# best_rmse = float('inf')
# best_params = None

# # Perform grid search
# for units in units_values:
#     for learning_rate in learning_rate_values:
#         model = create_lstm_model(units, learning_rate)

#         # Train the model
#         model.fit(x_train, y_train, batch_size=1, epochs=3, verbose=0)

#         # Prepare the testing data
#         test_data = scaled_data[training_data_monthly_len - 7 * test_weeks:, :]
#         x_test = []

#         for i in range(7 * test_weeks, len(test_data)):
#             x_test.append(test_data[i - (7 * test_weeks):i, 0])

#         x_test = np.array(x_test)
#         x_test = np.reshape(x_test, (x_test.shape[0], x_test.shape[1], 1))

#         # Get predictions
#         predictions = model.predict(x_test)
#         predictions = scaler.inverse_transform(predictions)

#         # Get the root mean squared error (RMSE)
#         rmse = np.sqrt(mean_squared_error(predictions, nonstationary_single_user_dat
```

```
#         print(f"Units: {units}, Learning Rate: {learning_rate}, Test RMSE: {rmse}")

#         # Update the best parameters if RMSE is improved
#         if rmse < best_rmse:
#             best_rmse = rmse
#             best_params = {'units': units, 'learning_rate': learning_rate}

# # Print the best hyperparameters
# print(f"Best Hyperparameters: {best_params}, Best RMSE: {best_rmse}")
```

Saved results of LSTM Hyperparameter tuning for better reducing computation time.

In [54]:
```
# Mean: 35.68018188057653
# Standard Deviation: 122.09556811748713
# Variance: 14907.32775393194
# 1/1 [==============================] - 1s 853ms/step
# Units: 64, Learning Rate: 0.01, Test RMSE: 375.2553253207926
# 1/1 [==============================] - 1s 840ms/step
# Units: 64, Learning Rate: 0.001, Test RMSE: 231.66883181925292
# 1/1 [==============================] - 1s 907ms/step
# Units: 64, Learning Rate: 0.0001, Test RMSE: 299.90232400037746
# 1/1 [==============================] - 1s 902ms/step
# Units: 128, Learning Rate: 0.01, Test RMSE: 398.97323688410444
# 1/1 [==============================] - 1s 879ms/step
# Units: 128, Learning Rate: 0.001, Test RMSE: 241.82620950011136
# 1/1 [==============================] - 1s 854ms/step
# Units: 128, Learning Rate: 0.0001, Test RMSE: 279.9525872191042
# 1/1 [==============================] - 1s 886ms/step
# Units: 256, Learning Rate: 0.01, Test RMSE: 398.22153453188594
# 1/1 [==============================] - 1s 876ms/step
# Units: 256, Learning Rate: 0.001, Test RMSE: 289.69737162552474
# 1/1 [==============================] - 1s 879ms/step
# Units: 256, Learning Rate: 0.0001, Test RMSE: 266.59826253364014
# Best Hyperparameters: {'units': 64, 'learning_rate': 0.001}, Best RMSE: 231.66883181
```

Model Fitting of LSTM and Evaluation in Loop

In [55]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.preprocessing import MinMaxScaler
from sklearn.metrics import mean_absolute_error
from keras.models import Sequential
from keras.layers import Dense, LSTM
from keras.optimizers import Adam
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt



for lstmuser in lstmusers['UserID']:
```

```
nonstationary_single_user_data=user_data_3d['Turnover'][user_data_3d.index==lstmus
nonstationary_single_user_data
```

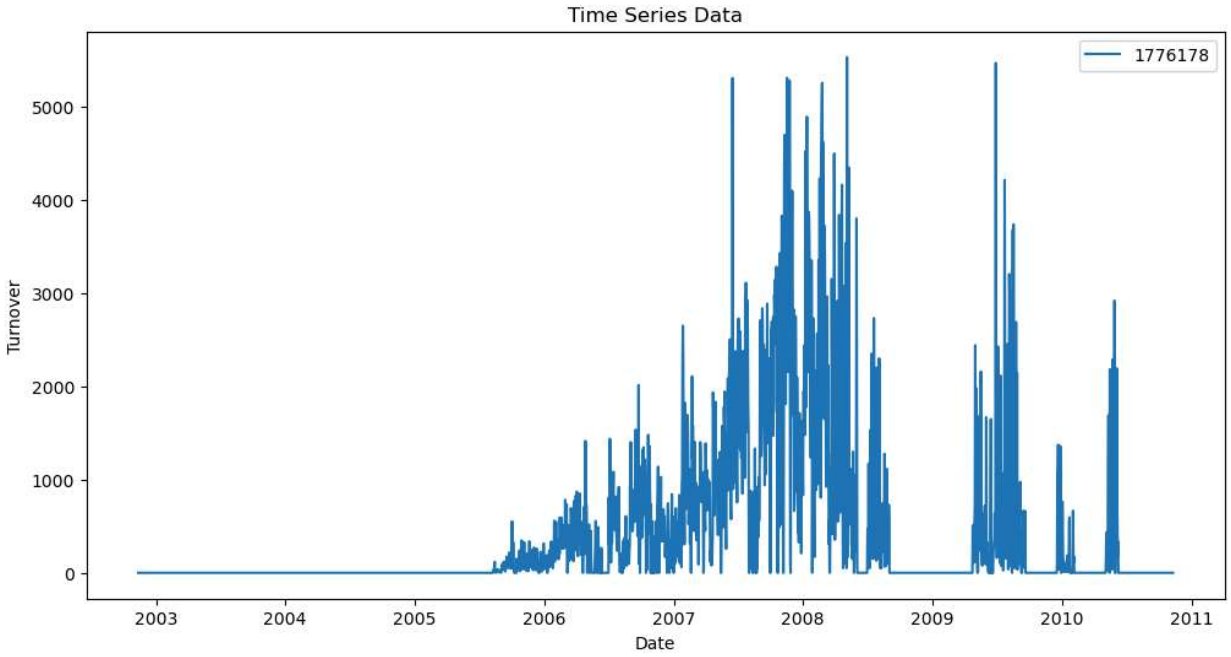
```
x_single_user_turnover_ravel=nonstationary_single_user_data.values.ravel()
x_single_user_turnover_ravel
```

```
nonstationary_single_user_data_transposed = nonstationary_single_user_data.T

# Plot the time series with dates as columns
plt.figure(figsize=(12, 6))
for column in nonstationary_single_user_data_transposed.columns:
    plt.plot(nonstationary_single_user_data_transposed.index, nonstationary_single

plt.xlabel('Date')
plt.ylabel('Turnover')
plt.title('Time Series Data')
plt.legend()
plt.show()
```

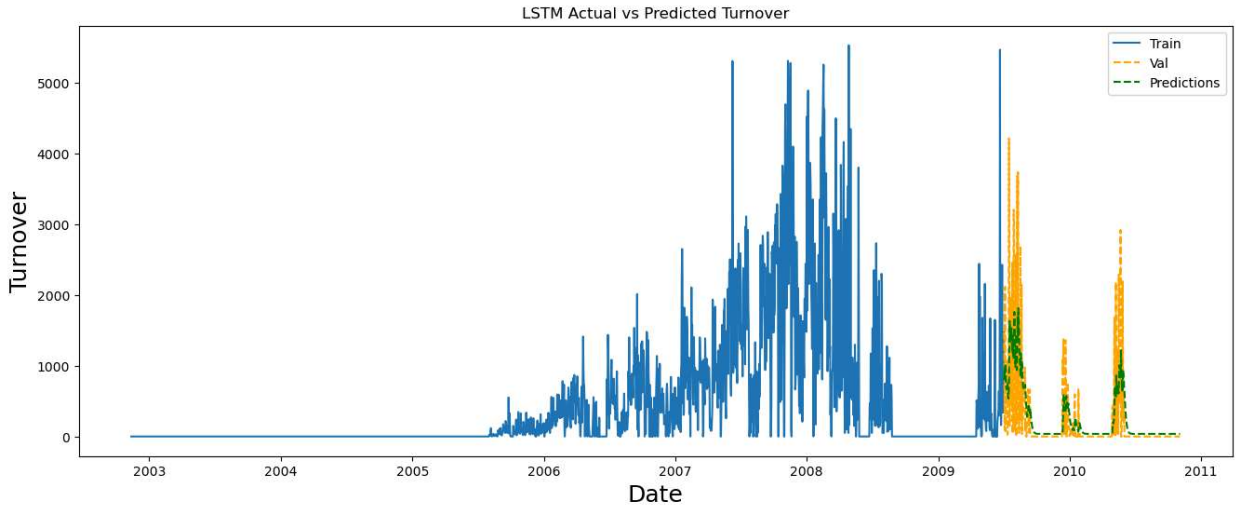
```
# Extract statistics from the data
row_variance = nonstationary_single_user_data.var(axis=1, skipna=True).iloc[0]
row_mean = nonstationary_single_user_data.mean(axis=1).iloc[0]
row_std = nonstationary_single_user_data.std(axis=1).iloc[0]

print(f'Mean: {row_mean}')
print(f'Standard Deviation: {row_std}')
print(f'Variance: {row_variance}')

# Flatten the data
nonstationary_single_user_data_ravel = nonstationary_single_user_data.values.ravel

test_weeks=70

# Set the length of training data
training_data_monthly_len = len(nonstationary_single_user_data_ravel) - 7*test_wee

# Scale the data
scaler = MinMaxScaler(feature_range=(0, 1))
scaled_data = scaler.fit_transform(nonstationary_single_user_data_ravel.reshape(-1
train_data_monthly = scaled_data[0:int(training_data_monthly_len), :]

# Prepare the training data
x_train = []
y_train = []

for i in range(7*test_weeks, len(train_data_monthly)):
    x_train.append(train_data_monthly[i-7*test_weeks:i, 0])
```

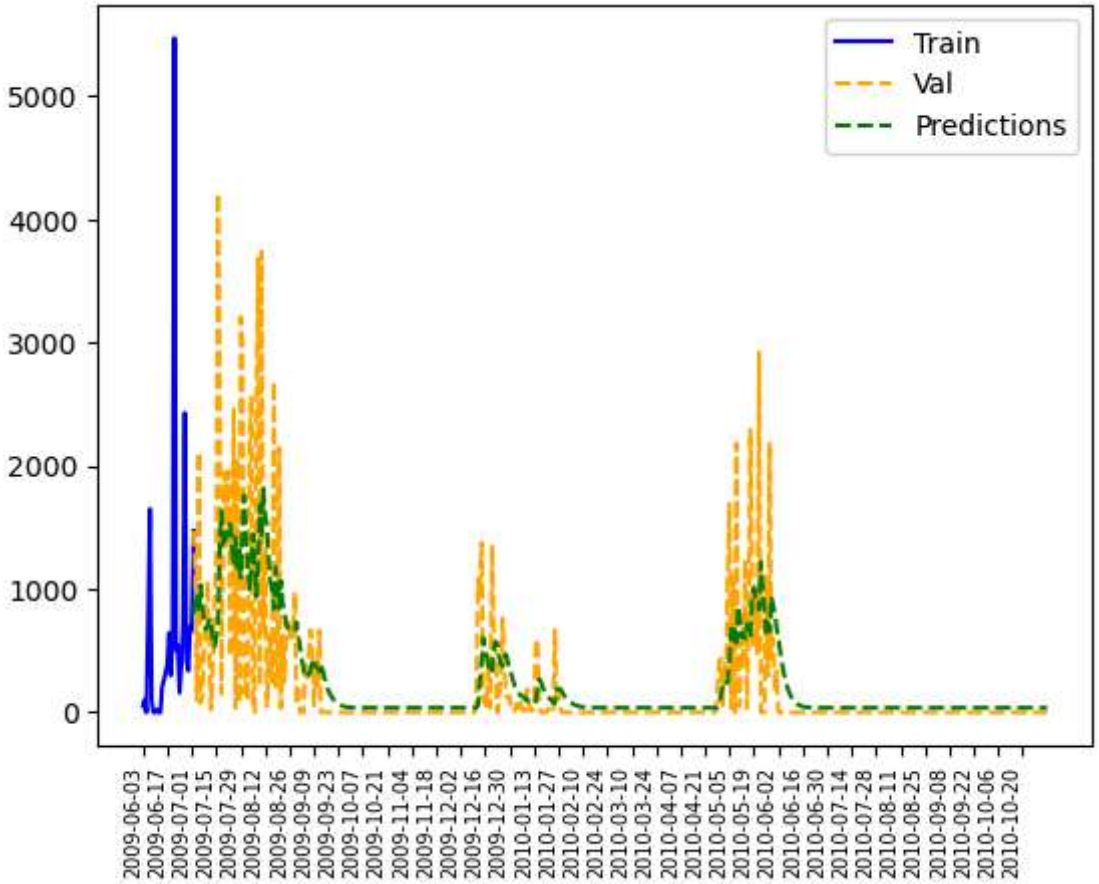
```
    y_train.append(train_data_monthly[i, 0])

x_train, y_train = np.array(x_train), np.array(y_train)

# Reshape the data
x_train = np.reshape(x_train, (x_train.shape[0], x_train.shape[1], 1))

# Define the best hyperparameters
best_units = 64
best_learning_rate = 0.001

# Build and compile the LSTM model with the best hyperparameters
best_model = Sequential()
best_model.add(LSTM(units=best_units, return_sequences=True, input_shape=(x_train.
best_model.add(LSTM(units=best_units, return_sequences=False))
best_model.add(Dense(units=25))
best_model.add(Dense(units=1))

optimizer = Adam(learning_rate=best_learning_rate)
best_model.compile(optimizer=optimizer, loss='mean_squared_error')

# Train the model
best_model.fit(x_train, y_train, batch_size=1, epochs=1)  # Adjust epochs as neede

# Prepare the testing data
test_data = scaled_data[training_data_monthly_len - (7*test_weeks):, :]
x_test = []

for i in range(7*test_weeks, len(test_data)):
    x_test.append(test_data[i-(7*test_weeks):i, 0])

x_test = np.array(x_test)
x_test = np.reshape(x_test, (x_test.shape[0], x_test.shape[1], 1))

# Get predictions
predictions = best_model.predict(x_test)
predictions = scaler.inverse_transform(predictions)

# Get the root mean squared error (RMSE)
rmse = np.sqrt(np.mean(((predictions - nonstationary_single_user_data_ravel[traini
print("Test RMSE:", rmse)

# Get the mean absolute error (MAE)
mae = mean_absolute_error(nonstationary_single_user_data_ravel[training_data_month
print("Test MAE:", mae)



from datetime import datetime, timedelta
# Generate time indices
start_date = '2002-11-12'
end_date = '2010-11-10'
#test_window_lastweek='2010-11-03'


end_date_dt = datetime.strptime(end_date, '%Y-%m-%d')

time_index_train = pd.date_range(start=start_date,  periods=training_data_monthly_
time_index_test = pd.date_range(start=end_date_dt - timedelta(days=(7 * (test_week
```

```
# Visualize the data
train = nonstationary_single_user_data_ravel[:training_data_monthly_len+(1)]
valid = pd.DataFrame({'x': nonstationary_single_user_data_ravel[training_data_mont
valid['Predictions'] = predictions

plt.figure(figsize=(16, 6))
plt.title('LSTM Actual vs Predicted Turnover')
plt.xlabel('Date', fontsize=18)
plt.ylabel('Turnover', fontsize=18)
plt.plot(time_index_train, train[:])
plt.plot(time_index_test, valid['x'], linestyle='--', color='orange')
plt.plot(time_index_test, valid['Predictions'], linestyle='--', color='green')
plt.legend(['Train', 'Val', 'Predictions'], loc='upper right')
plt.show()



# Plot only the last 30 values
last_30_values = -30

# Plot training data
plt.plot(time_index_train[-30:], train[last_30_values:], linestyle='-', color='blu

# Plot validation data
plt.plot(time_index_test, valid['x'], linestyle='--', color='orange')  # Use x_tes
plt.plot(time_index_test, valid['Predictions'], linestyle='--', color='green')  #

plt.legend(['Train', 'Val', 'Predictions'], loc='upper right')

# Rotate x-axis labels
plt.xticks(rotation=90, ha='right')
# Increase the number of x-axis ticks
plt.xticks(np.arange(time_index_train[-30], time_index_test[-1], timedelta(days=14


plt.show()


# Compute differences between consecutive predicted values
differences = np.diff(predictions.flatten())

# Set a threshold to identify surges
threshold_difference = 0.5

# Identify surges based on differences and threshold
surge_indices = np.where(differences > threshold_difference)[0]

# Print the timestamps of the points right before a surge
for index in surge_indices:
    if index > 0:
        surge_start_timestamp = time_index_test[index]
        print(f"PG started right before: {surge_start_timestamp}")
```



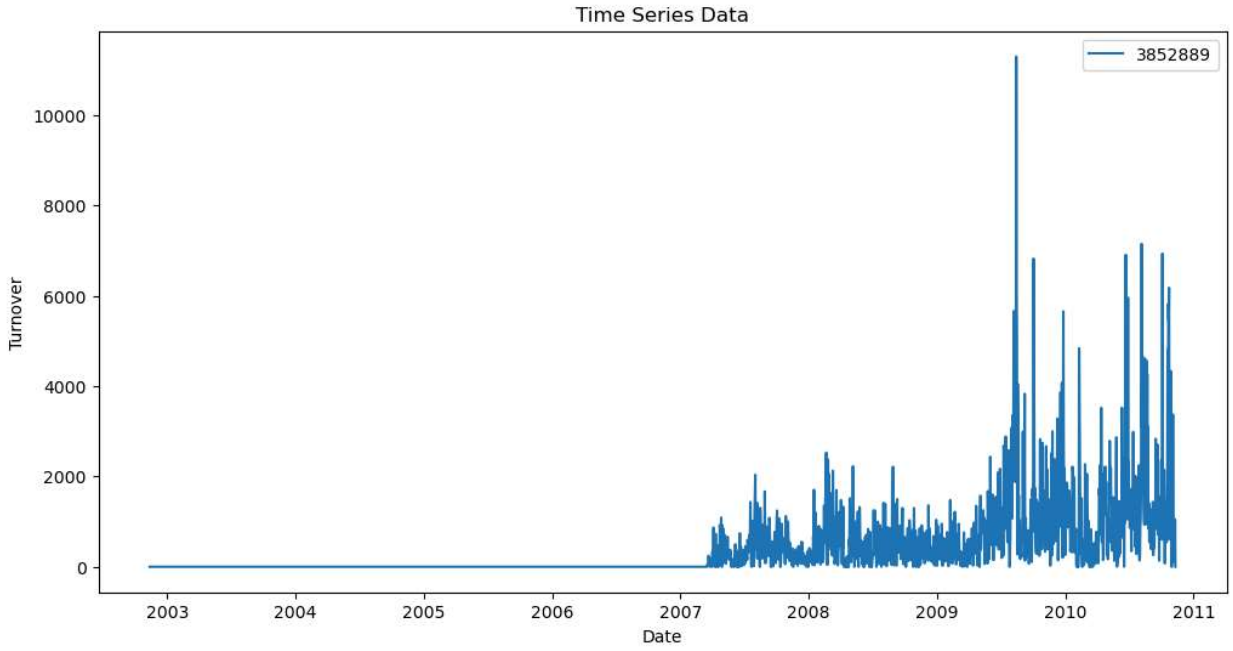

```
Mean: 386.32035549073436
Standard Deviation: 798.9267000819931
Variance: 638283.872103903
1934/1934 [==============================] - 286s 146ms/step - loss: 0.0121
16/16 [==============================] - 4s 144ms/step
Test RMSE: 699.6919454731204
Test MAE: 226.75674804189254
```



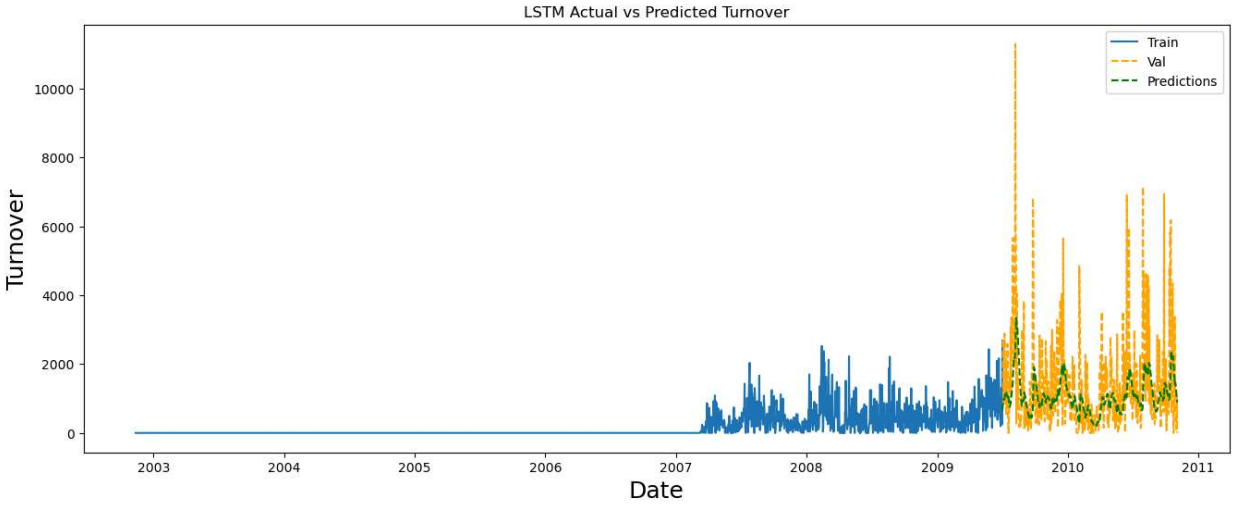

Train
Val
Predictions
5000
4000
3000
2000
1000
0
2009-06-03
2009-06-17
2009-07-01
2009-07-15
2009-07-29
2009-08-12
2009-08-26
2009-09-09
2009-09-23
2009-10-07
2009-10-21
2009-11-04
2009-11-18
2009-12-02
2009-12-16
2009-12-30
2010-01-13
2010-01-27
2010-02-10
2010-02-24
2010-03-10
2010-03-24
2010-04-07
2010-04-21
2010-05-05
2010-05-19
2010-06-02
2010-06-16
2010-06-30
2010-07-14
2010-07-28
2010-08-11
2010-08-25
2010-09-08
2010-09-22
2010-10-06
2010-10-20

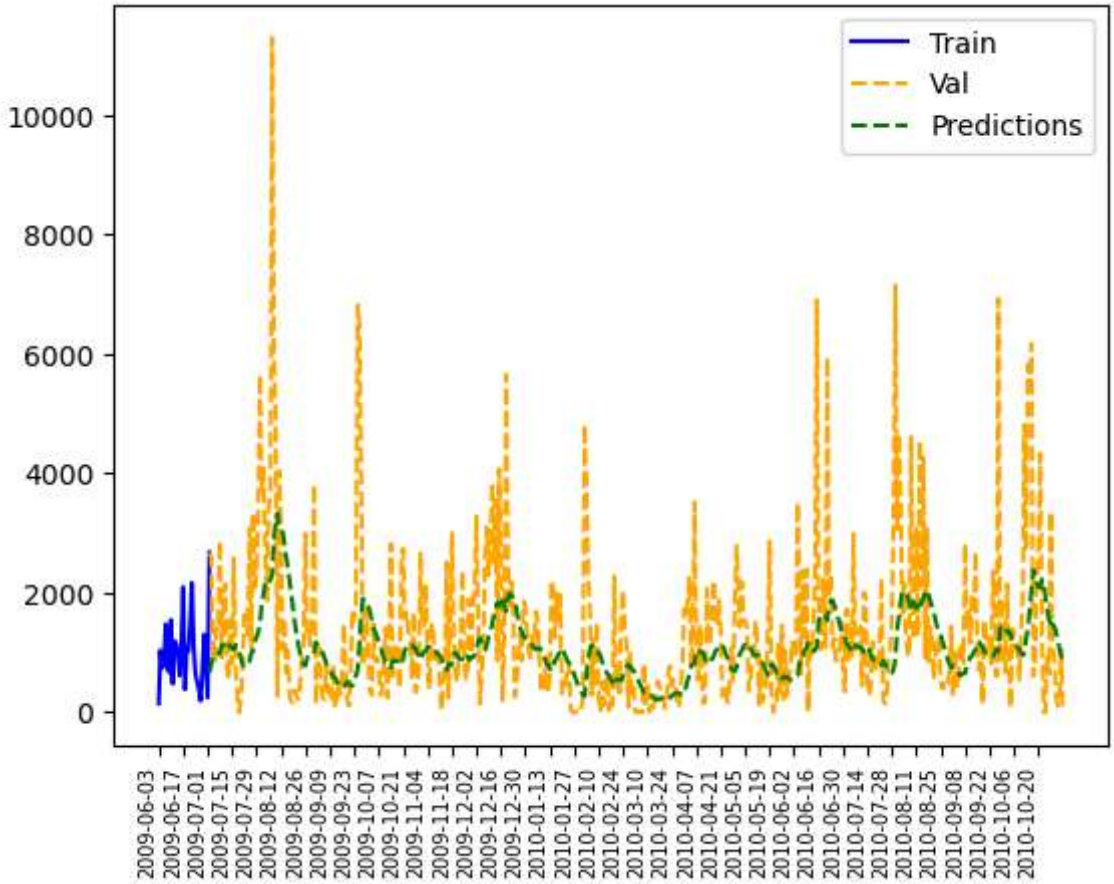
```
PG started right before: 2009-07-03 00:00:00
PG started right before: 2009-07-05 00:00:00
PG started right before: 2009-07-09 00:00:00
PG started right before: 2009-07-10 00:00:00
PG started right before: 2009-07-14 00:00:00
PG started right before: 2009-07-15 00:00:00
PG started right before: 2009-07-16 00:00:00
PG started right before: 2009-07-17 00:00:00
PG started right before: 2009-07-20 00:00:00
PG started right before: 2009-07-22 00:00:00
PG started right before: 2009-07-25 00:00:00
PG started right before: 2009-07-27 00:00:00
PG started right before: 2009-07-29 00:00:00
PG started right before: 2009-07-30 00:00:00
PG started right before: 2009-08-03 00:00:00
PG started right before: 2009-08-04 00:00:00
PG started right before: 2009-08-07 00:00:00
PG started right before: 2009-08-08 00:00:00
PG started right before: 2009-08-10 00:00:00
PG started right before: 2009-08-17 00:00:00
PG started right before: 2009-08-20 00:00:00
PG started right before: 2009-08-25 00:00:00
PG started right before: 2009-08-27 00:00:00
PG started right before: 2009-08-29 00:00:00
PG started right before: 2009-09-04 00:00:00
PG started right before: 2009-09-05 00:00:00
PG started right before: 2009-09-06 00:00:00
PG started right before: 2009-09-07 00:00:00
PG started right before: 2009-09-08 00:00:00
PG started right before: 2009-09-12 00:00:00
PG started right before: 2009-12-11 00:00:00
PG started right before: 2009-12-12 00:00:00
PG started right before: 2009-12-13 00:00:00
PG started right before: 2009-12-14 00:00:00
PG started right before: 2009-12-18 00:00:00
PG started right before: 2009-12-20 00:00:00
PG started right before: 2009-12-21 00:00:00
PG started right before: 2009-12-26 00:00:00
PG started right before: 2009-12-27 00:00:00
PG started right before: 2010-01-09 00:00:00
PG started right before: 2010-01-10 00:00:00
PG started right before: 2010-01-14 00:00:00
PG started right before: 2010-01-15 00:00:00
PG started right before: 2010-01-25 00:00:00
PG started right before: 2010-01-26 00:00:00
PG started right before: 2010-01-27 00:00:00
PG started right before: 2010-04-28 00:00:00
PG started right before: 2010-04-29 00:00:00
PG started right before: 2010-04-30 00:00:00
PG started right before: 2010-05-02 00:00:00
PG started right before: 2010-05-03 00:00:00
PG started right before: 2010-05-04 00:00:00
PG started right before: 2010-05-05 00:00:00
PG started right before: 2010-05-07 00:00:00
PG started right before: 2010-05-09 00:00:00
PG started right before: 2010-05-12 00:00:00
PG started right before: 2010-05-14 00:00:00
PG started right before: 2010-05-16 00:00:00
PG started right before: 2010-05-17 00:00:00
PG started right before: 2010-05-18 00:00:00
```

```
PG started right before: 2010-05-22 00:00:00
PG started right before: 2010-05-26 00:00:00
PG started right before: 2010-05-28 00:00:00
```




```
Mean: 365.90467207501774
Standard Deviation: 775.0542406959867
Variance: 600709.0760208324
1934/1934 [==============================] - 303s 154ms/step - loss: 6.9536e-04
16/16 [==============================] - 3s 110ms/step
Test RMSE: 1465.6427761608077
Test MAE: 844.4668944654891
```



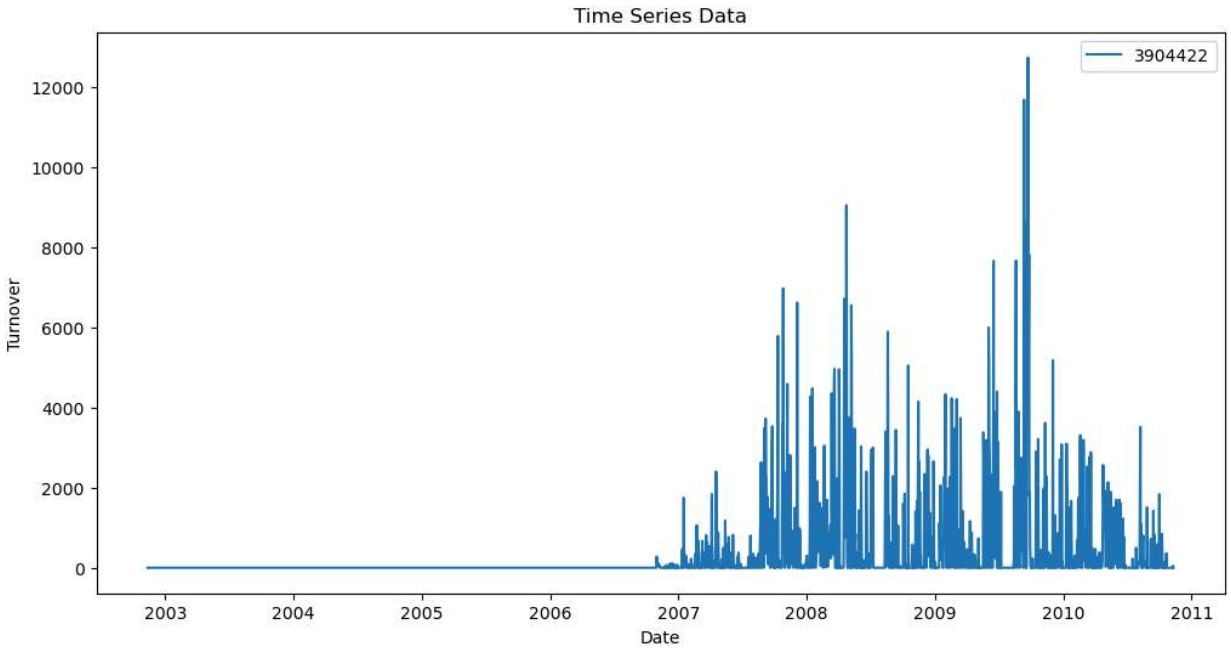

Train
Val
Predictions
10000
8000
6000
4000
2000
0
2009-06-03
2009-06-17
2009-07-01
2009-07-15
2009-07-29
2009-08-12
2009-08-26
2009-09-09
2009-09-23
2009-10-07
2009-10-21
2009-11-04
2009-11-18
2009-12-02
2009-12-16
2009-12-30
2010-01-13
2010-01-27
2010-02-10
2010-02-24
2010-03-10
2010-03-24
2010-04-07
2010-04-21
2010-05-05
2010-05-19
2010-06-02
2010-06-16
2010-06-30
2010-07-14
2010-07-28
2010-08-11
2010-08-25
2010-09-08
2010-09-22
2010-10-06
2010-10-20

```
PG started right before: 2009-07-03 00:00:00
PG started right before: 2009-07-04 00:00:00
PG started right before: 2009-07-06 00:00:00
PG started right before: 2009-07-07 00:00:00
PG started right before: 2009-07-08 00:00:00
PG started right before: 2009-07-10 00:00:00
PG started right before: 2009-07-11 00:00:00
PG started right before: 2009-07-16 00:00:00
PG started right before: 2009-07-22 00:00:00
PG started right before: 2009-07-23 00:00:00
PG started right before: 2009-07-24 00:00:00
PG started right before: 2009-07-25 00:00:00
PG started right before: 2009-07-26 00:00:00
PG started right before: 2009-07-27 00:00:00
PG started right before: 2009-07-28 00:00:00
PG started right before: 2009-07-29 00:00:00
PG started right before: 2009-07-30 00:00:00
PG started right before: 2009-07-31 00:00:00
PG started right before: 2009-08-01 00:00:00
PG started right before: 2009-08-02 00:00:00
PG started right before: 2009-08-03 00:00:00
PG started right before: 2009-08-05 00:00:00
PG started right before: 2009-08-06 00:00:00
PG started right before: 2009-08-07 00:00:00
PG started right before: 2009-08-08 00:00:00
PG started right before: 2009-08-09 00:00:00
PG started right before: 2009-08-26 00:00:00
PG started right before: 2009-08-27 00:00:00
PG started right before: 2009-08-28 00:00:00
PG started right before: 2009-08-30 00:00:00
PG started right before: 2009-08-31 00:00:00
PG started right before: 2009-09-17 00:00:00
PG started right before: 2009-09-18 00:00:00
PG started right before: 2009-09-21 00:00:00
PG started right before: 2009-09-22 00:00:00
PG started right before: 2009-09-23 00:00:00
PG started right before: 2009-09-24 00:00:00
PG started right before: 2009-09-25 00:00:00
PG started right before: 2009-09-26 00:00:00
PG started right before: 2009-09-27 00:00:00
PG started right before: 2009-10-11 00:00:00
PG started right before: 2009-10-14 00:00:00
PG started right before: 2009-10-17 00:00:00
PG started right before: 2009-10-20 00:00:00
PG started right before: 2009-10-21 00:00:00
PG started right before: 2009-10-22 00:00:00
PG started right before: 2009-10-23 00:00:00
PG started right before: 2009-10-24 00:00:00
PG started right before: 2009-10-27 00:00:00
PG started right before: 2009-10-31 00:00:00
PG started right before: 2009-11-01 00:00:00
PG started right before: 2009-11-02 00:00:00
PG started right before: 2009-11-03 00:00:00
PG started right before: 2009-11-07 00:00:00
PG started right before: 2009-11-15 00:00:00
PG started right before: 2009-11-17 00:00:00
PG started right before: 2009-11-18 00:00:00
PG started right before: 2009-11-19 00:00:00
PG started right before: 2009-11-23 00:00:00
PG started right before: 2009-11-24 00:00:00
```

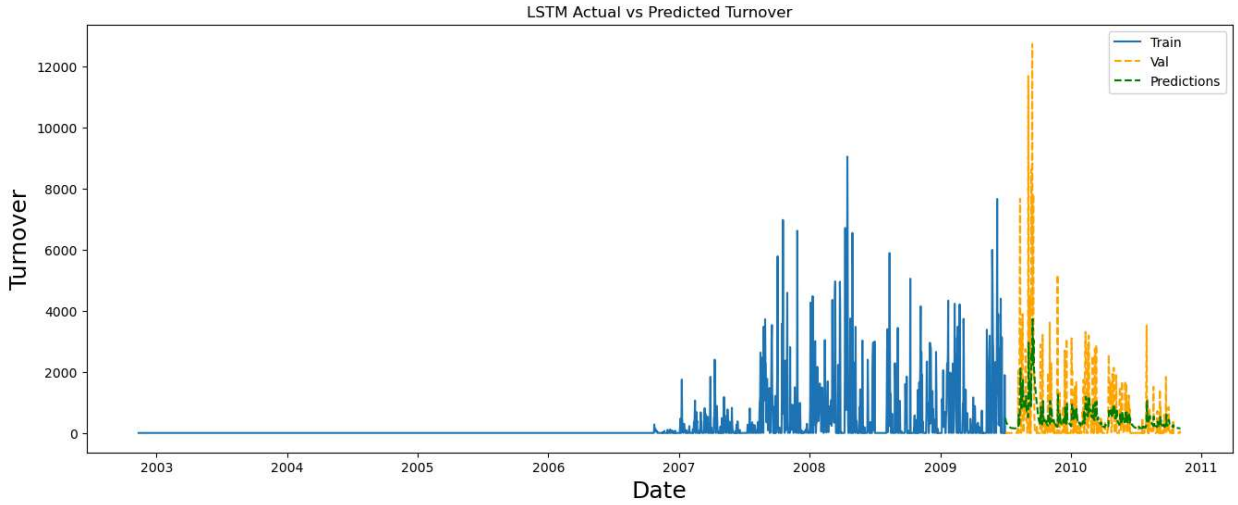
```
PG started right before: 2009-11-28 00:00:00
PG started right before: 2009-11-30 00:00:00
PG started right before: 2009-12-01 00:00:00
PG started right before: 2009-12-02 00:00:00
PG started right before: 2009-12-03 00:00:00
PG started right before: 2009-12-07 00:00:00
PG started right before: 2009-12-08 00:00:00
PG started right before: 2009-12-09 00:00:00
PG started right before: 2009-12-10 00:00:00
PG started right before: 2009-12-11 00:00:00
PG started right before: 2009-12-12 00:00:00
PG started right before: 2009-12-13 00:00:00
PG started right before: 2009-12-15 00:00:00
PG started right before: 2009-12-16 00:00:00
PG started right before: 2009-12-19 00:00:00
PG started right before: 2009-12-20 00:00:00
PG started right before: 2009-12-30 00:00:00
PG started right before: 2010-01-05 00:00:00
PG started right before: 2010-01-06 00:00:00
PG started right before: 2010-01-14 00:00:00
PG started right before: 2010-01-15 00:00:00
PG started right before: 2010-01-16 00:00:00
PG started right before: 2010-01-17 00:00:00
PG started right before: 2010-01-18 00:00:00
PG started right before: 2010-01-19 00:00:00
PG started right before: 2010-02-02 00:00:00
PG started right before: 2010-02-03 00:00:00
PG started right before: 2010-02-04 00:00:00
PG started right before: 2010-02-05 00:00:00
PG started right before: 2010-02-09 00:00:00
PG started right before: 2010-02-19 00:00:00
PG started right before: 2010-02-20 00:00:00
PG started right before: 2010-02-21 00:00:00
PG started right before: 2010-02-23 00:00:00
PG started right before: 2010-02-24 00:00:00
PG started right before: 2010-02-25 00:00:00
PG started right before: 2010-02-26 00:00:00
PG started right before: 2010-03-08 00:00:00
PG started right before: 2010-03-16 00:00:00
PG started right before: 2010-03-17 00:00:00
PG started right before: 2010-03-22 00:00:00
PG started right before: 2010-03-23 00:00:00
PG started right before: 2010-03-24 00:00:00
PG started right before: 2010-03-25 00:00:00
PG started right before: 2010-03-26 00:00:00
PG started right before: 2010-03-30 00:00:00
PG started right before: 2010-03-31 00:00:00
PG started right before: 2010-04-01 00:00:00
PG started right before: 2010-04-02 00:00:00
PG started right before: 2010-04-03 00:00:00
PG started right before: 2010-04-04 00:00:00
PG started right before: 2010-04-05 00:00:00
PG started right before: 2010-04-06 00:00:00
PG started right before: 2010-04-07 00:00:00
PG started right before: 2010-04-09 00:00:00
PG started right before: 2010-04-13 00:00:00
PG started right before: 2010-04-14 00:00:00
PG started right before: 2010-04-15 00:00:00
PG started right before: 2010-04-16 00:00:00
PG started right before: 2010-04-17 00:00:00
```

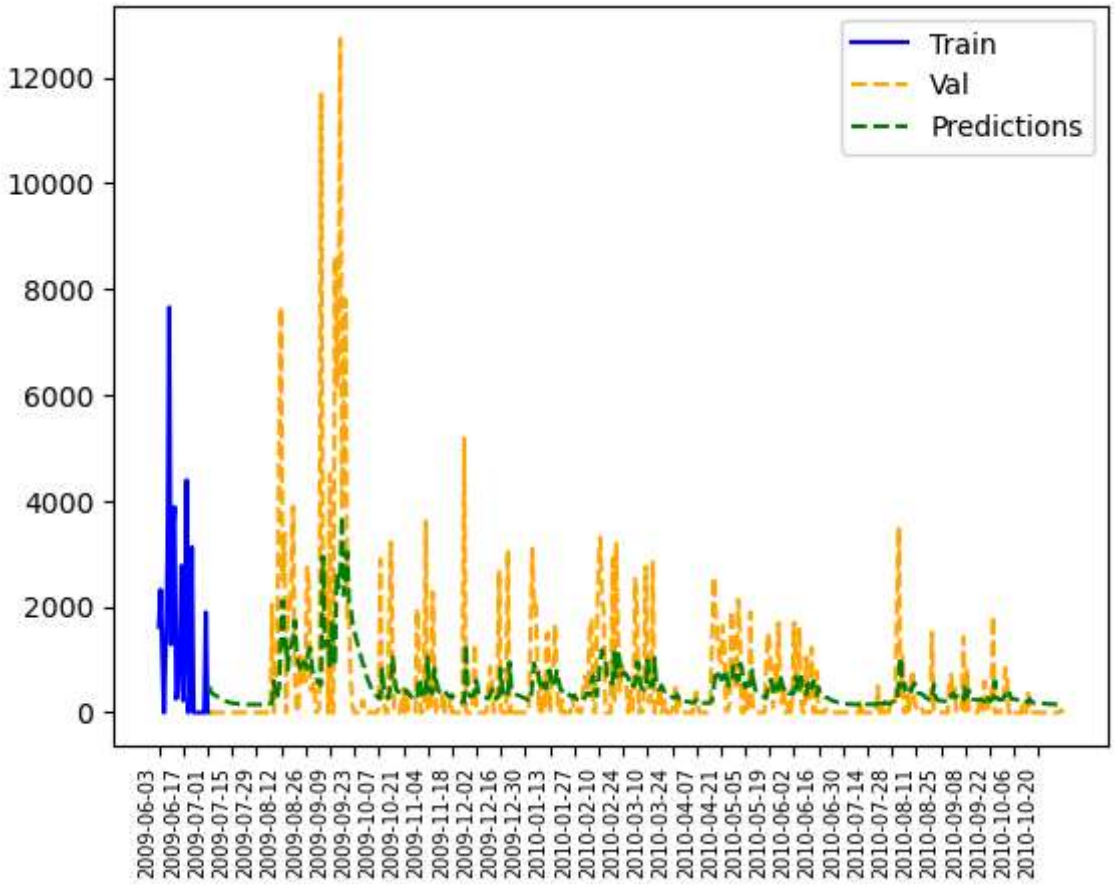
```
PG started right before: 2010-04-18 00:00:00
PG started right before: 2010-04-19 00:00:00
PG started right before: 2010-04-20 00:00:00
PG started right before: 2010-04-21 00:00:00
PG started right before: 2010-04-29 00:00:00
PG started right before: 2010-04-30 00:00:00
PG started right before: 2010-05-01 00:00:00
PG started right before: 2010-05-02 00:00:00
PG started right before: 2010-05-03 00:00:00
PG started right before: 2010-05-04 00:00:00
PG started right before: 2010-05-05 00:00:00
PG started right before: 2010-05-06 00:00:00
PG started right before: 2010-05-11 00:00:00
PG started right before: 2010-05-18 00:00:00
PG started right before: 2010-05-19 00:00:00
PG started right before: 2010-05-24 00:00:00
PG started right before: 2010-05-26 00:00:00
PG started right before: 2010-05-28 00:00:00
PG started right before: 2010-05-31 00:00:00
PG started right before: 2010-06-01 00:00:00
PG started right before: 2010-06-02 00:00:00
PG started right before: 2010-06-03 00:00:00
PG started right before: 2010-06-04 00:00:00
PG started right before: 2010-06-05 00:00:00
PG started right before: 2010-06-07 00:00:00
PG started right before: 2010-06-08 00:00:00
PG started right before: 2010-06-09 00:00:00
PG started right before: 2010-06-13 00:00:00
PG started right before: 2010-06-14 00:00:00
PG started right before: 2010-06-15 00:00:00
PG started right before: 2010-06-16 00:00:00
PG started right before: 2010-06-19 00:00:00
PG started right before: 2010-06-21 00:00:00
PG started right before: 2010-06-22 00:00:00
PG started right before: 2010-06-23 00:00:00
PG started right before: 2010-07-06 00:00:00
PG started right before: 2010-07-10 00:00:00
PG started right before: 2010-07-12 00:00:00
PG started right before: 2010-07-13 00:00:00
PG started right before: 2010-07-22 00:00:00
PG started right before: 2010-07-27 00:00:00
PG started right before: 2010-07-28 00:00:00
PG started right before: 2010-07-29 00:00:00
PG started right before: 2010-07-30 00:00:00
PG started right before: 2010-07-31 00:00:00
PG started right before: 2010-08-01 00:00:00
PG started right before: 2010-08-02 00:00:00
PG started right before: 2010-08-08 00:00:00
PG started right before: 2010-08-11 00:00:00
PG started right before: 2010-08-13 00:00:00
PG started right before: 2010-08-14 00:00:00
PG started right before: 2010-08-15 00:00:00
PG started right before: 2010-08-17 00:00:00
PG started right before: 2010-08-31 00:00:00
PG started right before: 2010-09-05 00:00:00
PG started right before: 2010-09-06 00:00:00
PG started right before: 2010-09-07 00:00:00
PG started right before: 2010-09-08 00:00:00
PG started right before: 2010-09-09 00:00:00
PG started right before: 2010-09-10 00:00:00
```

```
PG started right before: 2010-09-11 00:00:00
PG started right before: 2010-09-12 00:00:00
PG started right before: 2010-09-13 00:00:00
PG started right before: 2010-09-14 00:00:00
PG started right before: 2010-09-15 00:00:00
PG started right before: 2010-09-24 00:00:00
PG started right before: 2010-09-25 00:00:00
PG started right before: 2010-09-27 00:00:00
PG started right before: 2010-09-28 00:00:00
PG started right before: 2010-09-29 00:00:00
PG started right before: 2010-10-02 00:00:00
PG started right before: 2010-10-11 00:00:00
PG started right before: 2010-10-12 00:00:00
PG started right before: 2010-10-13 00:00:00
PG started right before: 2010-10-14 00:00:00
PG started right before: 2010-10-15 00:00:00
PG started right before: 2010-10-16 00:00:00
PG started right before: 2010-10-21 00:00:00
PG started right before: 2010-10-27 00:00:00
```




```
Mean: 297.5882071771934
Standard Deviation: 912.9051402201292
Variance: 833395.7950403338
1934/1934 [==============================] - 258s 132ms/step - loss: 0.0041
16/16 [==============================] - 3s 114ms/step
Test RMSE: 1438.35631447229
Test MAE: 607.3983705502761
```

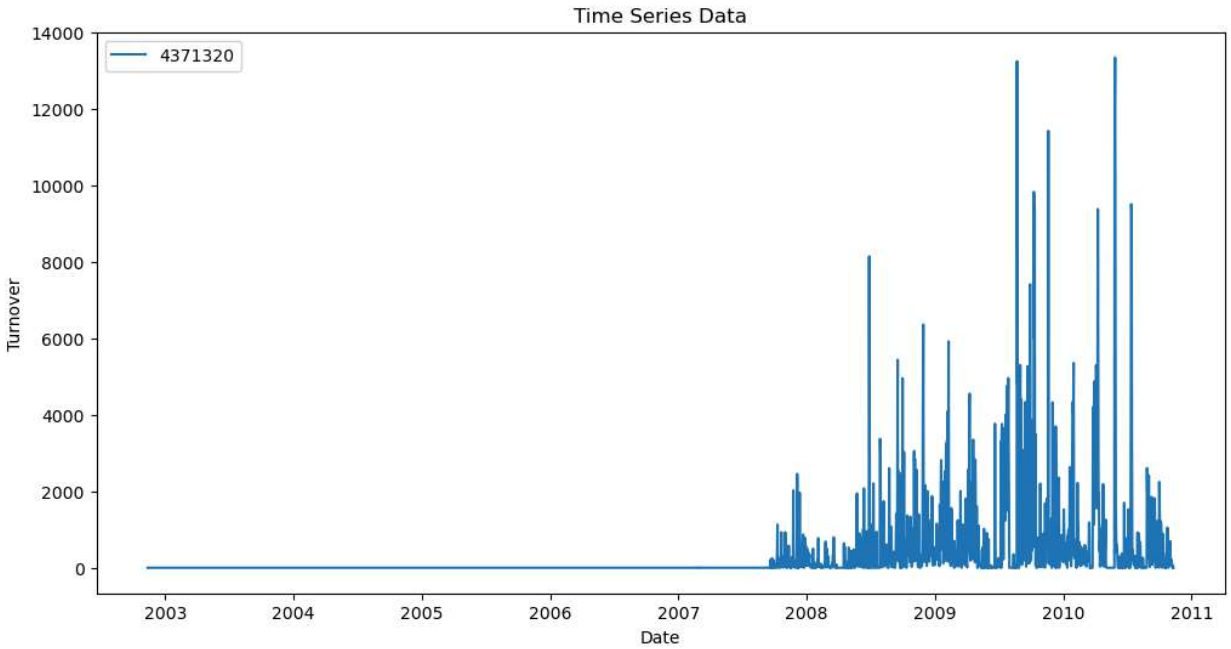

LSTM Actual vs Predicted Turnover
Train
Val
Predictions
Turnover
Date
0
2000
4000
6000
8000
10000
12000
2003
2004
2005
2006
2007
2008
2009
2010
2011


Train
Val
Predictions
0
2000
4000
6000
8000
10000
12000
2009-06-03
2009-06-17
2009-07-01
2009-07-15
2009-07-29
2009-08-12
2009-08-26
2009-09-09
2009-09-23
2009-10-07
2009-10-21
2009-11-04
2009-11-18
2009-12-02
2009-12-16
2009-12-30
2010-01-13
2010-01-27
2010-02-10
2010-02-24
2010-03-10
2010-03-24
2010-04-07
2010-04-21
2010-05-05
2010-05-19
2010-06-02
2010-06-16
2010-06-30
2010-07-14
2010-07-28
2010-08-11
2010-08-25
2010-09-08
2010-09-22
2010-10-06
2010-10-20

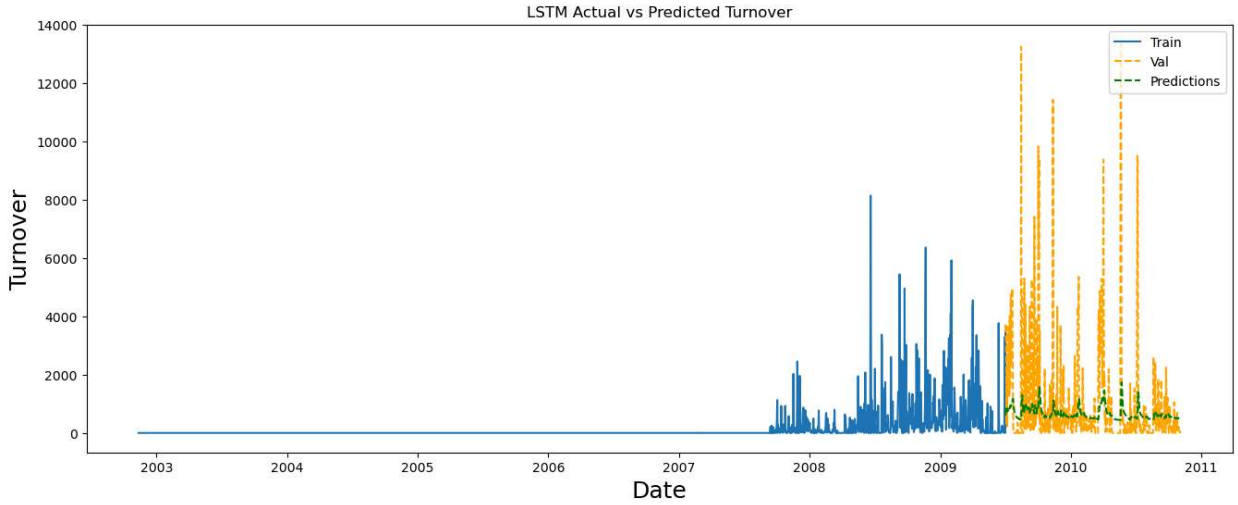
```
PG started right before: 2009-08-05 00:00:00
PG started right before: 2009-08-06 00:00:00
PG started right before: 2009-08-07 00:00:00
PG started right before: 2009-08-09 00:00:00
PG started right before: 2009-08-10 00:00:00
PG started right before: 2009-08-11 00:00:00
PG started right before: 2009-08-12 00:00:00
PG started right before: 2009-08-14 00:00:00
PG started right before: 2009-08-16 00:00:00
PG started right before: 2009-08-17 00:00:00
PG started right before: 2009-08-18 00:00:00
PG started right before: 2009-08-19 00:00:00
PG started right before: 2009-08-23 00:00:00
PG started right before: 2009-08-25 00:00:00
PG started right before: 2009-08-27 00:00:00
PG started right before: 2009-08-31 00:00:00
PG started right before: 2009-09-04 00:00:00
PG started right before: 2009-09-07 00:00:00
PG started right before: 2009-09-09 00:00:00
PG started right before: 2009-09-12 00:00:00
PG started right before: 2009-09-14 00:00:00
PG started right before: 2009-09-15 00:00:00
PG started right before: 2009-09-17 00:00:00
PG started right before: 2009-09-18 00:00:00
PG started right before: 2009-10-08 00:00:00
PG started right before: 2009-10-13 00:00:00
PG started right before: 2009-10-14 00:00:00
PG started right before: 2009-10-18 00:00:00
PG started right before: 2009-10-20 00:00:00
PG started right before: 2009-10-21 00:00:00
PG started right before: 2009-10-23 00:00:00
PG started right before: 2009-10-29 00:00:00
PG started right before: 2009-11-02 00:00:00
PG started right before: 2009-11-03 00:00:00
PG started right before: 2009-11-07 00:00:00
PG started right before: 2009-11-12 00:00:00
PG started right before: 2009-11-13 00:00:00
PG started right before: 2009-11-17 00:00:00
PG started right before: 2009-11-25 00:00:00
PG started right before: 2009-11-29 00:00:00
PG started right before: 2009-11-30 00:00:00
PG started right before: 2009-12-01 00:00:00
PG started right before: 2009-12-09 00:00:00
PG started right before: 2009-12-10 00:00:00
PG started right before: 2009-12-13 00:00:00
PG started right before: 2009-12-14 00:00:00
PG started right before: 2009-12-15 00:00:00
PG started right before: 2009-12-19 00:00:00
PG started right before: 2009-12-20 00:00:00
PG started right before: 2010-01-02 00:00:00
PG started right before: 2010-01-03 00:00:00
PG started right before: 2010-01-05 00:00:00
PG started right before: 2010-01-08 00:00:00
PG started right before: 2010-01-11 00:00:00
PG started right before: 2010-01-12 00:00:00
PG started right before: 2010-01-14 00:00:00
PG started right before: 2010-01-15 00:00:00
PG started right before: 2010-01-16 00:00:00
PG started right before: 2010-01-20 00:00:00
PG started right before: 2010-01-26 00:00:00
```

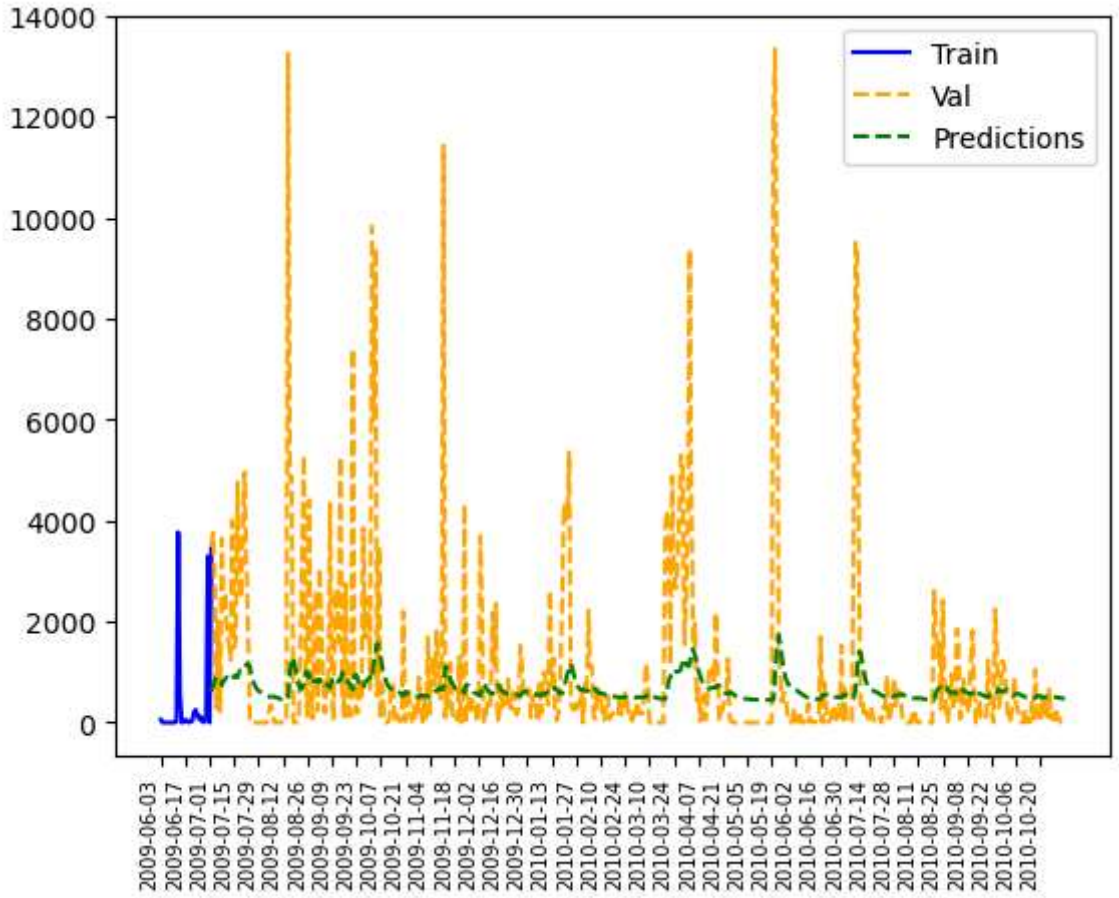
```
PG started right before: 2010-02-01 00:00:00
PG started right before: 2010-02-02 00:00:00
PG started right before: 2010-02-03 00:00:00
PG started right before: 2010-02-05 00:00:00
PG started right before: 2010-02-06 00:00:00
PG started right before: 2010-02-09 00:00:00
PG started right before: 2010-02-10 00:00:00
PG started right before: 2010-02-11 00:00:00
PG started right before: 2010-02-18 00:00:00
PG started right before: 2010-02-20 00:00:00
PG started right before: 2010-02-23 00:00:00
PG started right before: 2010-02-28 00:00:00
PG started right before: 2010-03-02 00:00:00
PG started right before: 2010-03-03 00:00:00
PG started right before: 2010-03-08 00:00:00
PG started right before: 2010-03-09 00:00:00
PG started right before: 2010-03-12 00:00:00
PG started right before: 2010-03-13 00:00:00
PG started right before: 2010-03-17 00:00:00
PG started right before: 2010-03-20 00:00:00
PG started right before: 2010-03-25 00:00:00
PG started right before: 2010-03-26 00:00:00
PG started right before: 2010-03-28 00:00:00
PG started right before: 2010-04-06 00:00:00
PG started right before: 2010-04-07 00:00:00
PG started right before: 2010-04-14 00:00:00
PG started right before: 2010-04-15 00:00:00
PG started right before: 2010-04-16 00:00:00
PG started right before: 2010-04-17 00:00:00
PG started right before: 2010-04-22 00:00:00
PG started right before: 2010-04-26 00:00:00
PG started right before: 2010-04-27 00:00:00
PG started right before: 2010-05-01 00:00:00
PG started right before: 2010-05-05 00:00:00
PG started right before: 2010-05-08 00:00:00
PG started right before: 2010-05-17 00:00:00
PG started right before: 2010-05-18 00:00:00
PG started right before: 2010-05-22 00:00:00
PG started right before: 2010-05-24 00:00:00
PG started right before: 2010-05-27 00:00:00
PG started right before: 2010-05-30 00:00:00
PG started right before: 2010-06-01 00:00:00
PG started right before: 2010-06-02 00:00:00
PG started right before: 2010-06-05 00:00:00
PG started right before: 2010-06-09 00:00:00
PG started right before: 2010-06-12 00:00:00
PG started right before: 2010-06-15 00:00:00
PG started right before: 2010-06-20 00:00:00
PG started right before: 2010-07-10 00:00:00
PG started right before: 2010-07-14 00:00:00
PG started right before: 2010-07-20 00:00:00
PG started right before: 2010-07-23 00:00:00
PG started right before: 2010-07-27 00:00:00
PG started right before: 2010-07-28 00:00:00
PG started right before: 2010-07-30 00:00:00
PG started right before: 2010-07-31 00:00:00
PG started right before: 2010-08-01 00:00:00
PG started right before: 2010-08-03 00:00:00
PG started right before: 2010-08-06 00:00:00
PG started right before: 2010-08-08 00:00:00
```

```
PG started right before: 2010-08-09 00:00:00
PG started right before: 2010-08-20 00:00:00
PG started right before: 2010-08-30 00:00:00
PG started right before: 2010-08-31 00:00:00
PG started right before: 2010-09-02 00:00:00
PG started right before: 2010-09-07 00:00:00
PG started right before: 2010-09-09 00:00:00
PG started right before: 2010-09-16 00:00:00
PG started right before: 2010-09-19 00:00:00
PG started right before: 2010-09-22 00:00:00
PG started right before: 2010-09-24 00:00:00
PG started right before: 2010-09-28 00:00:00
PG started right before: 2010-09-29 00:00:00
PG started right before: 2010-09-30 00:00:00
PG started right before: 2010-10-01 00:00:00
PG started right before: 2010-10-05 00:00:00
PG started right before: 2010-10-11 00:00:00
PG started right before: 2010-10-14 00:00:00
PG started right before: 2010-11-02 00:00:00
```




```
Mean: 276.44176282570834
Standard Deviation: 949.0362785545946
Variance: 900669.858012754
1934/1934 [==============================] - 326s 167ms/step - loss: 0.0016
16/16 [==============================] - 5s 211ms/step
Test RMSE: 1904.0475907483478
Test MAE: 938.5964909608357
```

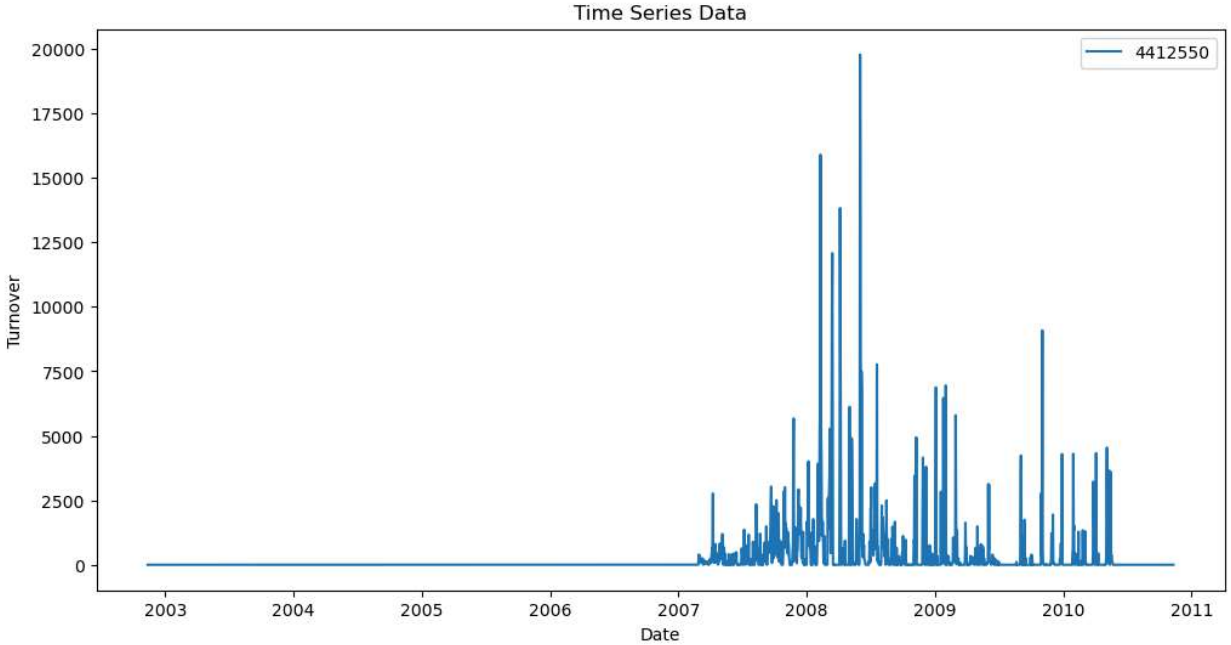

LSTM Actual vs Predicted Turnover
Train
Val
Predictions
Turnover
Date
14000
12000
10000
8000
6000
4000
2000
0
2003
2004
2005
2006
2007
2008
2009
2010
2011


Train
Val
Predictions
14000
12000
10000
8000
6000
4000
2000
0
2009-06-03
2009-06-17
2009-07-01
2009-07-15
2009-07-29
2009-08-12
2009-08-26
2009-09-09
2009-09-23
2009-10-07
2009-10-21
2009-11-04
2009-11-18
2009-12-02
2009-12-16
2009-12-30
2010-01-13
2010-01-27
2010-02-10
2010-02-24
2010-03-10
2010-03-24
2010-04-07
2010-04-21
2010-05-05
2010-05-19
2010-06-02
2010-06-16
2010-06-30
2010-07-14
2010-07-28
2010-08-11
2010-08-25
2010-09-08
2010-09-22
2010-10-06
2010-10-20

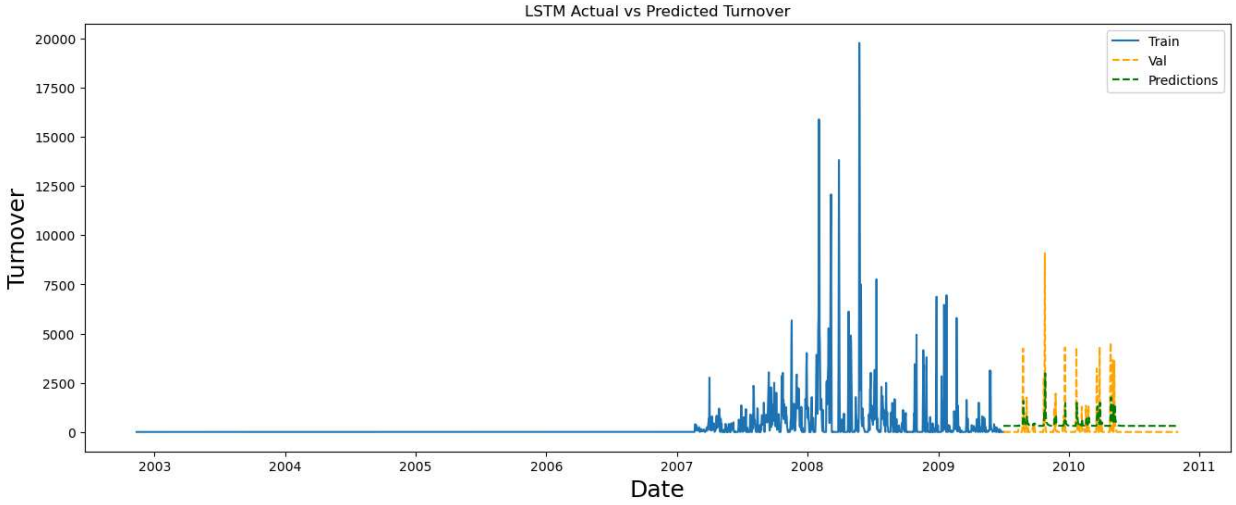
```
PG started right before: 2009-07-03 00:00:00
PG started right before: 2009-07-04 00:00:00
PG started right before: 2009-07-08 00:00:00
PG started right before: 2009-07-09 00:00:00
PG started right before: 2009-07-10 00:00:00
PG started right before: 2009-07-14 00:00:00
PG started right before: 2009-07-17 00:00:00
PG started right before: 2009-07-18 00:00:00
PG started right before: 2009-07-20 00:00:00
PG started right before: 2009-07-21 00:00:00
PG started right before: 2009-07-22 00:00:00
PG started right before: 2009-08-05 00:00:00
PG started right before: 2009-08-15 00:00:00
PG started right before: 2009-08-16 00:00:00
PG started right before: 2009-08-17 00:00:00
PG started right before: 2009-08-22 00:00:00
PG started right before: 2009-08-23 00:00:00
PG started right before: 2009-08-24 00:00:00
PG started right before: 2009-08-25 00:00:00
PG started right before: 2009-08-27 00:00:00
PG started right before: 2009-08-30 00:00:00
PG started right before: 2009-08-31 00:00:00
PG started right before: 2009-09-02 00:00:00
PG started right before: 2009-09-03 00:00:00
PG started right before: 2009-09-08 00:00:00
PG started right before: 2009-09-11 00:00:00
PG started right before: 2009-09-12 00:00:00
PG started right before: 2009-09-13 00:00:00
PG started right before: 2009-09-14 00:00:00
PG started right before: 2009-09-17 00:00:00
PG started right before: 2009-09-21 00:00:00
PG started right before: 2009-09-22 00:00:00
PG started right before: 2009-09-27 00:00:00
PG started right before: 2009-09-28 00:00:00
PG started right before: 2009-09-29 00:00:00
PG started right before: 2009-09-30 00:00:00
PG started right before: 2009-10-02 00:00:00
PG started right before: 2009-10-03 00:00:00
PG started right before: 2009-10-04 00:00:00
PG started right before: 2009-10-13 00:00:00
PG started right before: 2009-10-14 00:00:00
PG started right before: 2009-10-20 00:00:00
PG started right before: 2009-10-29 00:00:00
PG started right before: 2009-10-30 00:00:00
PG started right before: 2009-11-01 00:00:00
PG started right before: 2009-11-03 00:00:00
PG started right before: 2009-11-05 00:00:00
PG started right before: 2009-11-06 00:00:00
PG started right before: 2009-11-07 00:00:00
PG started right before: 2009-11-08 00:00:00
PG started right before: 2009-11-09 00:00:00
PG started right before: 2009-11-11 00:00:00
PG started right before: 2009-11-12 00:00:00
PG started right before: 2009-11-21 00:00:00
PG started right before: 2009-11-24 00:00:00
PG started right before: 2009-12-03 00:00:00
PG started right before: 2009-12-04 00:00:00
PG started right before: 2009-12-10 00:00:00
PG started right before: 2009-12-11 00:00:00
PG started right before: 2009-12-12 00:00:00
```

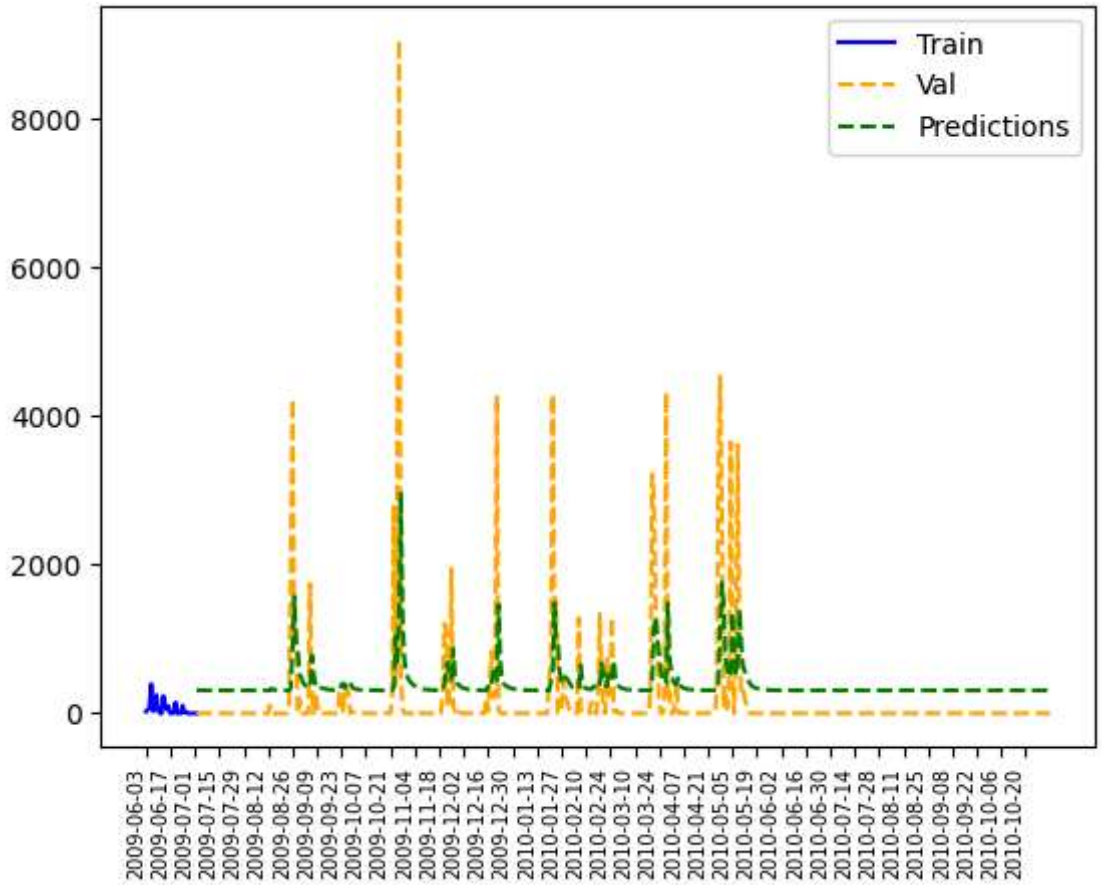
```
PG started right before: 2009-12-19 00:00:00
PG started right before: 2009-12-22 00:00:00
PG started right before: 2009-12-26 00:00:00
PG started right before: 2009-12-27 00:00:00
PG started right before: 2010-01-03 00:00:00
PG started right before: 2010-01-04 00:00:00
PG started right before: 2010-01-06 00:00:00
PG started right before: 2010-01-08 00:00:00
PG started right before: 2010-01-09 00:00:00
PG started right before: 2010-01-12 00:00:00
PG started right before: 2010-01-13 00:00:00
PG started right before: 2010-01-14 00:00:00
PG started right before: 2010-01-18 00:00:00
PG started right before: 2010-01-19 00:00:00
PG started right before: 2010-01-20 00:00:00
PG started right before: 2010-01-21 00:00:00
PG started right before: 2010-01-22 00:00:00
PG started right before: 2010-01-23 00:00:00
PG started right before: 2010-01-30 00:00:00
PG started right before: 2010-02-01 00:00:00
PG started right before: 2010-02-03 00:00:00
PG started right before: 2010-02-04 00:00:00
PG started right before: 2010-02-14 00:00:00
PG started right before: 2010-02-21 00:00:00
PG started right before: 2010-02-24 00:00:00
PG started right before: 2010-02-25 00:00:00
PG started right before: 2010-03-03 00:00:00
PG started right before: 2010-03-07 00:00:00
PG started right before: 2010-03-08 00:00:00
PG started right before: 2010-03-09 00:00:00
PG started right before: 2010-03-19 00:00:00
PG started right before: 2010-03-20 00:00:00
PG started right before: 2010-03-21 00:00:00
PG started right before: 2010-03-23 00:00:00
PG started right before: 2010-03-24 00:00:00
PG started right before: 2010-03-26 00:00:00
PG started right before: 2010-03-27 00:00:00
PG started right before: 2010-03-28 00:00:00
PG started right before: 2010-03-29 00:00:00
PG started right before: 2010-04-01 00:00:00
PG started right before: 2010-04-02 00:00:00
PG started right before: 2010-04-03 00:00:00
PG started right before: 2010-04-13 00:00:00
PG started right before: 2010-04-14 00:00:00
PG started right before: 2010-04-15 00:00:00
PG started right before: 2010-04-17 00:00:00
PG started right before: 2010-04-23 00:00:00
PG started right before: 2010-04-24 00:00:00
PG started right before: 2010-05-20 00:00:00
PG started right before: 2010-05-21 00:00:00
PG started right before: 2010-05-22 00:00:00
PG started right before: 2010-06-09 00:00:00
PG started right before: 2010-06-15 00:00:00
PG started right before: 2010-06-16 00:00:00
PG started right before: 2010-06-18 00:00:00
PG started right before: 2010-06-23 00:00:00
PG started right before: 2010-06-24 00:00:00
PG started right before: 2010-06-28 00:00:00
PG started right before: 2010-06-30 00:00:00
PG started right before: 2010-07-05 00:00:00
```

```
PG started right before: 2010-07-06 00:00:00
PG started right before: 2010-07-07 00:00:00
PG started right before: 2010-07-24 00:00:00
PG started right before: 2010-07-27 00:00:00
PG started right before: 2010-07-28 00:00:00
PG started right before: 2010-07-29 00:00:00
PG started right before: 2010-07-30 00:00:00
PG started right before: 2010-07-31 00:00:00
PG started right before: 2010-08-01 00:00:00
PG started right before: 2010-08-08 00:00:00
PG started right before: 2010-08-09 00:00:00
PG started right before: 2010-08-20 00:00:00
PG started right before: 2010-08-21 00:00:00
PG started right before: 2010-08-22 00:00:00
PG started right before: 2010-08-23 00:00:00
PG started right before: 2010-08-25 00:00:00
PG started right before: 2010-08-30 00:00:00
PG started right before: 2010-09-02 00:00:00
PG started right before: 2010-09-03 00:00:00
PG started right before: 2010-09-10 00:00:00
PG started right before: 2010-09-11 00:00:00
PG started right before: 2010-09-12 00:00:00
PG started right before: 2010-09-20 00:00:00
PG started right before: 2010-09-21 00:00:00
PG started right before: 2010-09-24 00:00:00
PG started right before: 2010-09-25 00:00:00
PG started right before: 2010-09-28 00:00:00
PG started right before: 2010-09-29 00:00:00
PG started right before: 2010-09-30 00:00:00
PG started right before: 2010-10-05 00:00:00
PG started right before: 2010-10-06 00:00:00
PG started right before: 2010-10-15 00:00:00
PG started right before: 2010-10-17 00:00:00
PG started right before: 2010-10-18 00:00:00
PG started right before: 2010-10-21 00:00:00
PG started right before: 2010-10-23 00:00:00
PG started right before: 2010-10-25 00:00:00
```

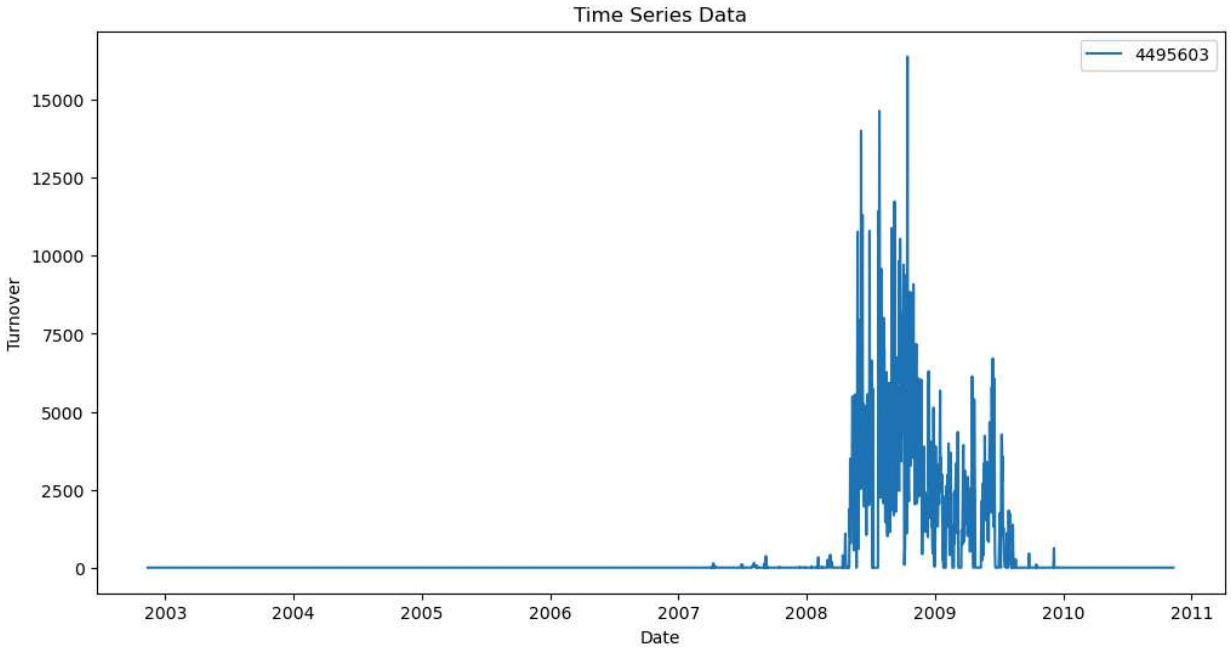


```
Mean: 239.93016815374057
Standard Deviation: 1034.4351019416395
Variance: 1070055.98012901
1934/1934 [==============================] - 397s 202ms/step - loss: 0.0034
16/16 [==============================] - 5s 199ms/step
Test RMSE: 817.9082970425846
Test MAE: 435.18406274663187
```

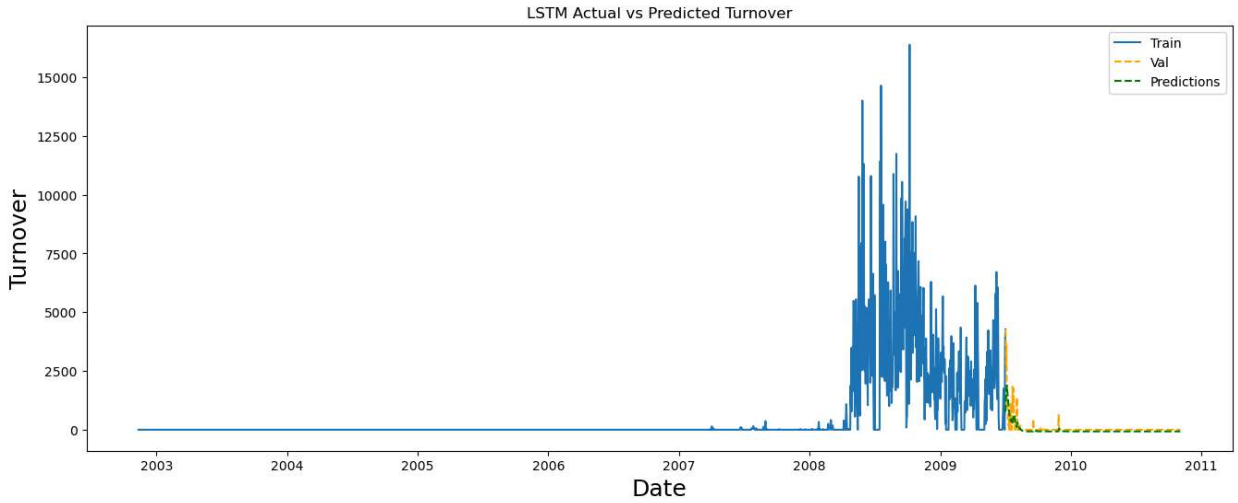


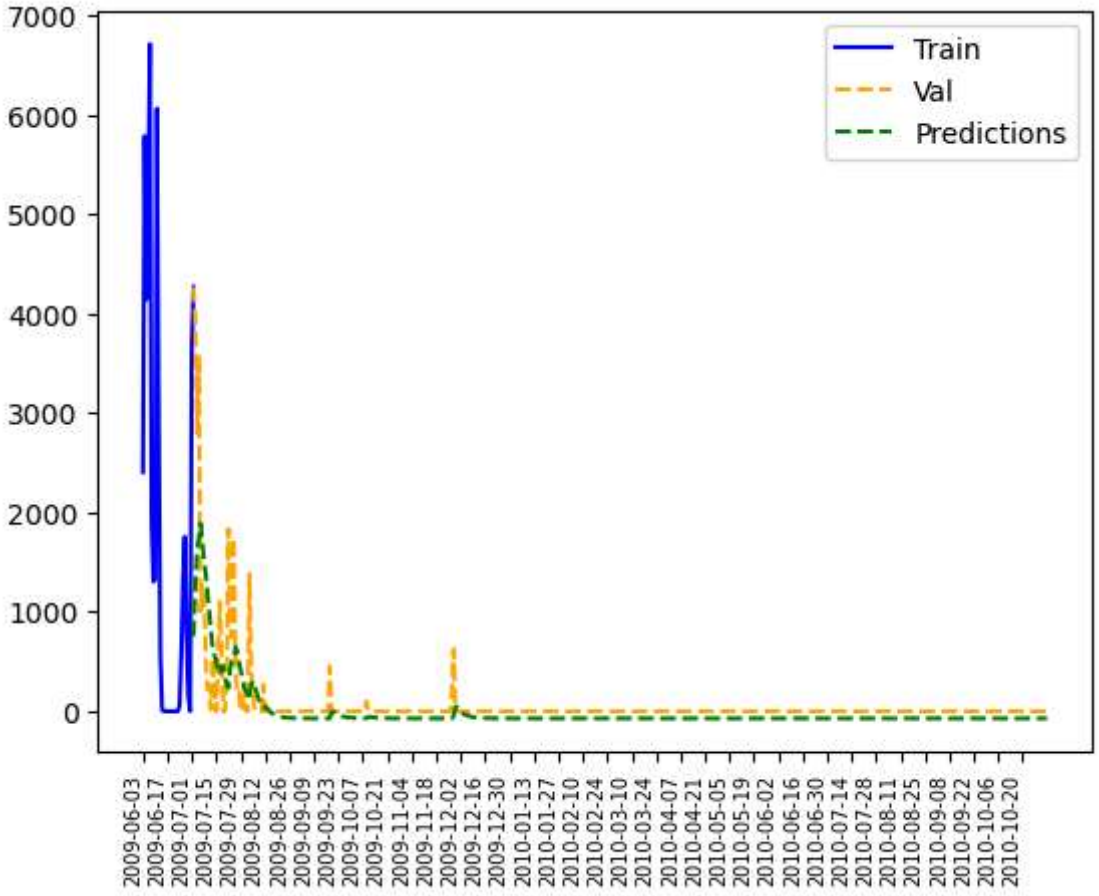


```
PG started right before: 2009-07-16 00:00:00
PG started right before: 2009-08-13 00:00:00
PG started right before: 2009-08-24 00:00:00
PG started right before: 2009-08-25 00:00:00
PG started right before: 2009-08-26 00:00:00
PG started right before: 2009-09-05 00:00:00
PG started right before: 2009-09-08 00:00:00
PG started right before: 2009-09-22 00:00:00
PG started right before: 2009-09-23 00:00:00
PG started right before: 2009-09-25 00:00:00
PG started right before: 2009-09-26 00:00:00
PG started right before: 2009-10-22 00:00:00
PG started right before: 2009-10-23 00:00:00
PG started right before: 2009-10-25 00:00:00
PG started right before: 2009-10-26 00:00:00
PG started right before: 2009-11-17 00:00:00
PG started right before: 2009-11-20 00:00:00
PG started right before: 2009-11-21 00:00:00
PG started right before: 2009-11-22 00:00:00
PG started right before: 2009-11-24 00:00:00
PG started right before: 2009-11-25 00:00:00
PG started right before: 2009-12-15 00:00:00
PG started right before: 2009-12-17 00:00:00
PG started right before: 2009-12-18 00:00:00
PG started right before: 2009-12-21 00:00:00
PG started right before: 2010-01-20 00:00:00
PG started right before: 2010-01-21 00:00:00
PG started right before: 2010-01-22 00:00:00
PG started right before: 2010-01-28 00:00:00
PG started right before: 2010-02-06 00:00:00
PG started right before: 2010-02-13 00:00:00
PG started right before: 2010-02-14 00:00:00
PG started right before: 2010-02-17 00:00:00
PG started right before: 2010-02-18 00:00:00
PG started right before: 2010-02-22 00:00:00
PG started right before: 2010-02-24 00:00:00
PG started right before: 2010-02-25 00:00:00
PG started right before: 2010-03-19 00:00:00
PG started right before: 2010-03-20 00:00:00
PG started right before: 2010-03-21 00:00:00
PG started right before: 2010-03-27 00:00:00
PG started right before: 2010-03-28 00:00:00
PG started right before: 2010-04-02 00:00:00
PG started right before: 2010-04-03 00:00:00
PG started right before: 2010-04-21 00:00:00
PG started right before: 2010-04-25 00:00:00
PG started right before: 2010-04-26 00:00:00
PG started right before: 2010-04-27 00:00:00
PG started right before: 2010-04-28 00:00:00
PG started right before: 2010-05-03 00:00:00
PG started right before: 2010-05-04 00:00:00
PG started right before: 2010-05-07 00:00:00
PG started right before: 2010-05-08 00:00:00
```

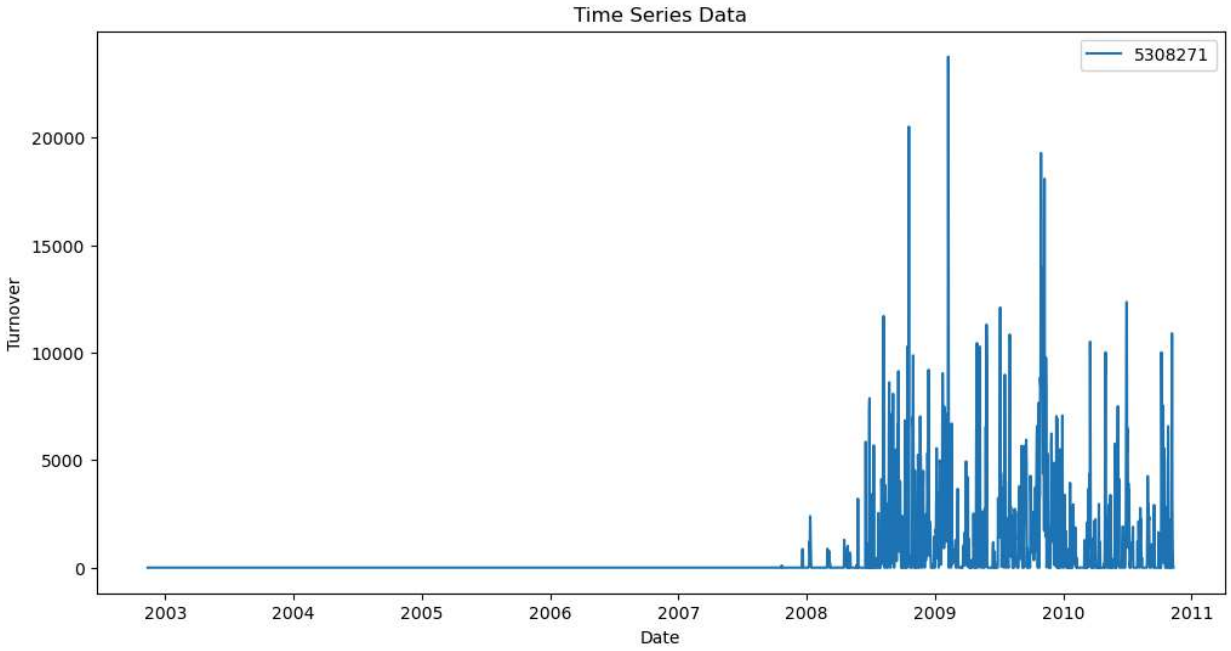



```
Mean: 445.9454516561502
Standard Deviation: 1455.7747813704814
Variance: 2119280.2140742727
1934/1934 [==============================] - 381s 194ms/step - loss: 0.0056
16/16 [==============================] - 5s 201ms/step
Test RMSE: 452.7626105463848
Test MAE: 116.34511226989632
```

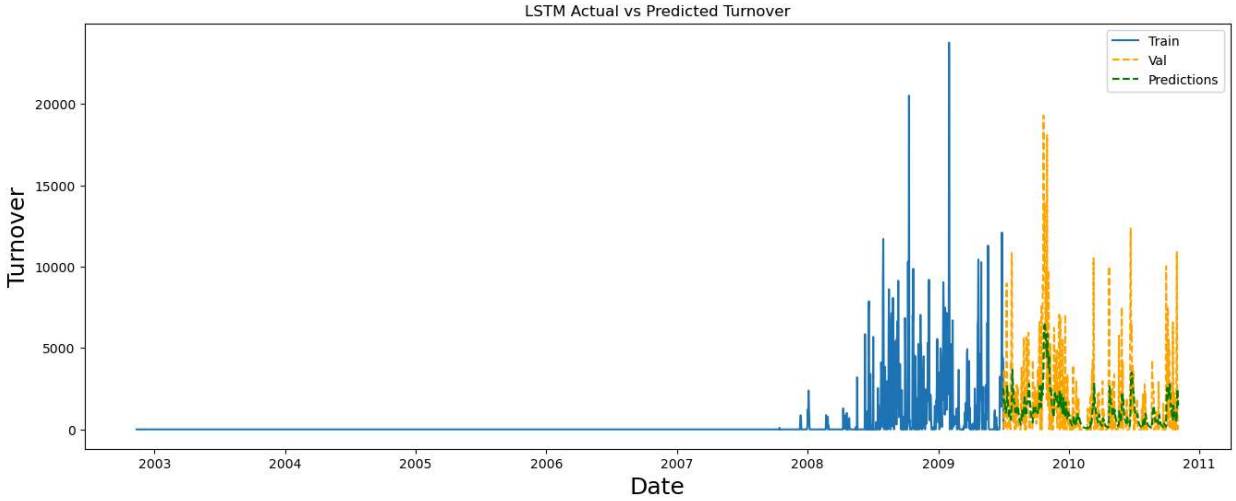


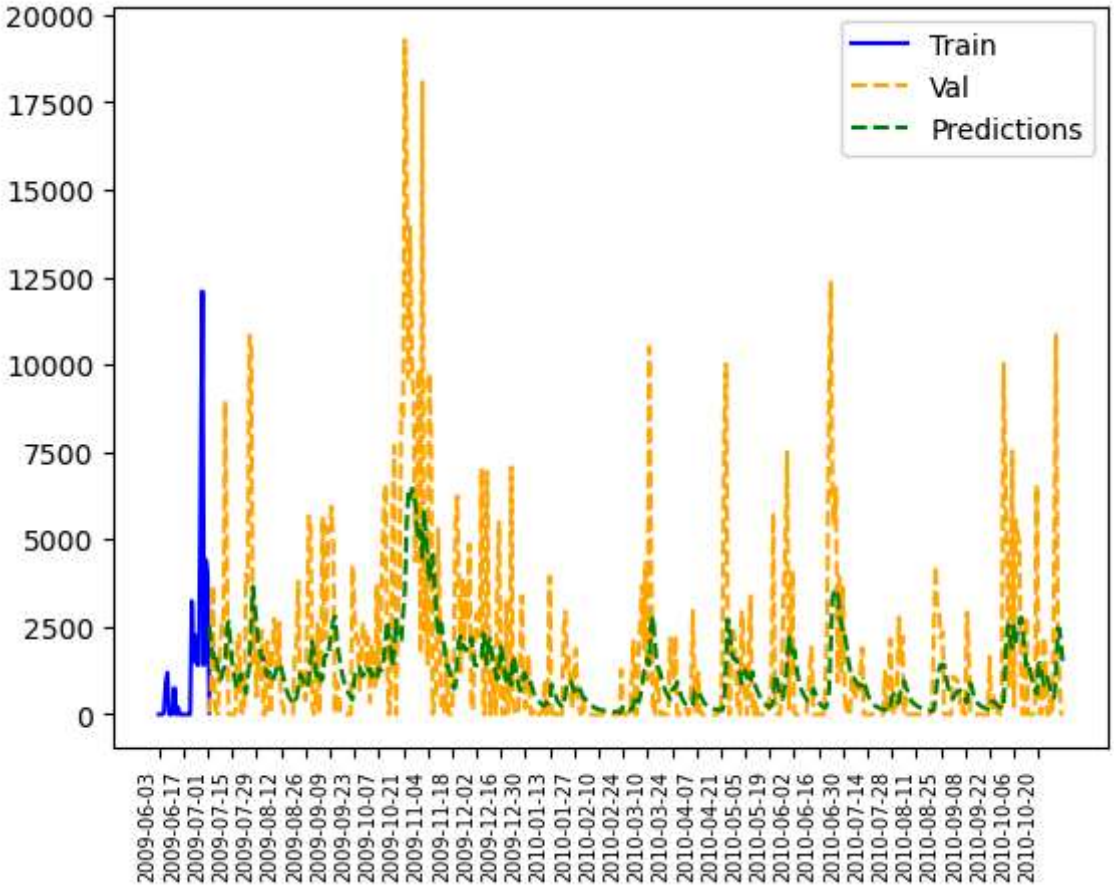



```
PG started right before: 2009-07-03 00:00:00
PG started right before: 2009-07-04 00:00:00
PG started right before: 2009-07-05 00:00:00
PG started right before: 2009-07-17 00:00:00
PG started right before: 2009-07-18 00:00:00
PG started right before: 2009-07-22 00:00:00
PG started right before: 2009-07-23 00:00:00
PG started right before: 2009-07-24 00:00:00
PG started right before: 2009-07-25 00:00:00
PG started right before: 2009-08-03 00:00:00
PG started right before: 2009-08-04 00:00:00
PG started right before: 2009-08-11 00:00:00
PG started right before: 2009-09-18 00:00:00
PG started right before: 2009-09-19 00:00:00
PG started right before: 2009-10-09 00:00:00
PG started right before: 2009-11-27 00:00:00
PG started right before: 2009-11-28 00:00:00
```




```
Mean: 555.0427506863418
Standard Deviation: 1780.5543780564858
Variance: 3170373.893216119
1934/1934 [==============================] - 301s 152ms/step - loss: 0.0038
16/16 [==============================] - 3s 128ms/step
Test RMSE: 2993.7127535504333
Test MAE: 1485.7203724097428
```

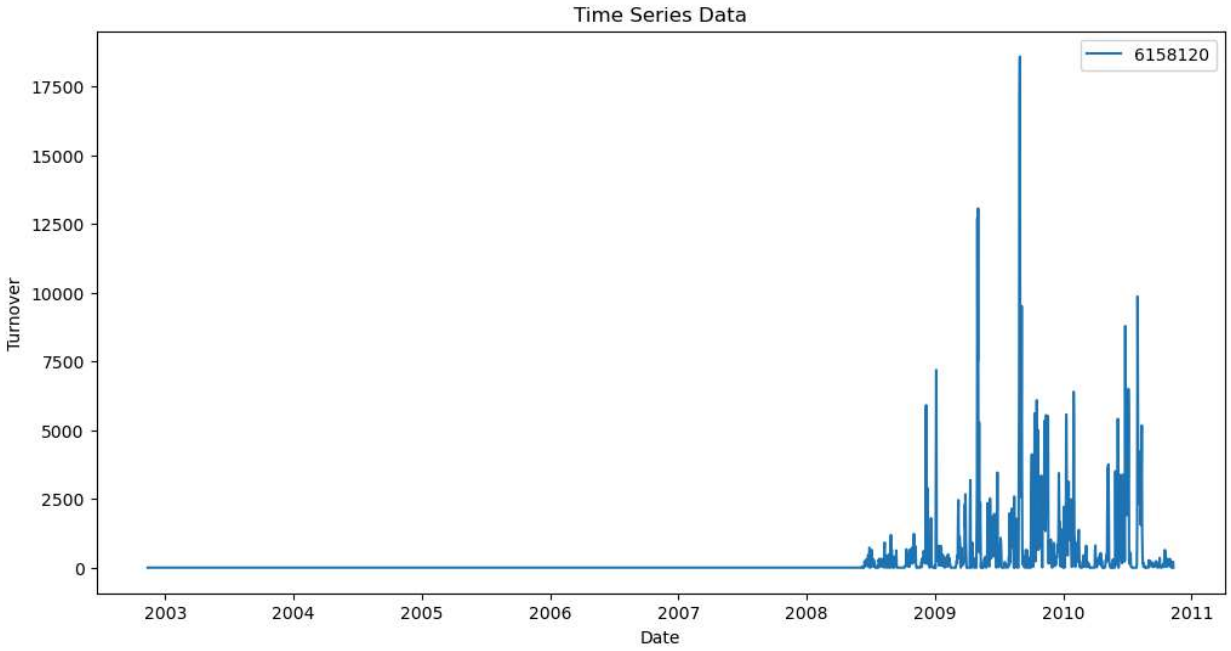



Train
Val
Predictions
20000
17500
15000
12500
10000
7500
5000
2500
0
2009-06-03
2009-06-17
2009-07-01
2009-07-15
2009-07-29
2009-08-12
2009-08-26
2009-09-09
2009-09-23
2009-10-07
2009-10-21
2009-11-04
2009-11-18
2009-12-02
2009-12-16
2009-12-30
2010-01-13
2010-01-27
2010-02-10
2010-02-24
2010-03-10
2010-03-24
2010-04-07
2010-04-21
2010-05-05
2010-05-19
2010-06-02
2010-06-16
2010-06-30
2010-07-14
2010-07-28
2010-08-11
2010-08-25
2010-09-08
2010-09-22
2010-10-06
2010-10-20

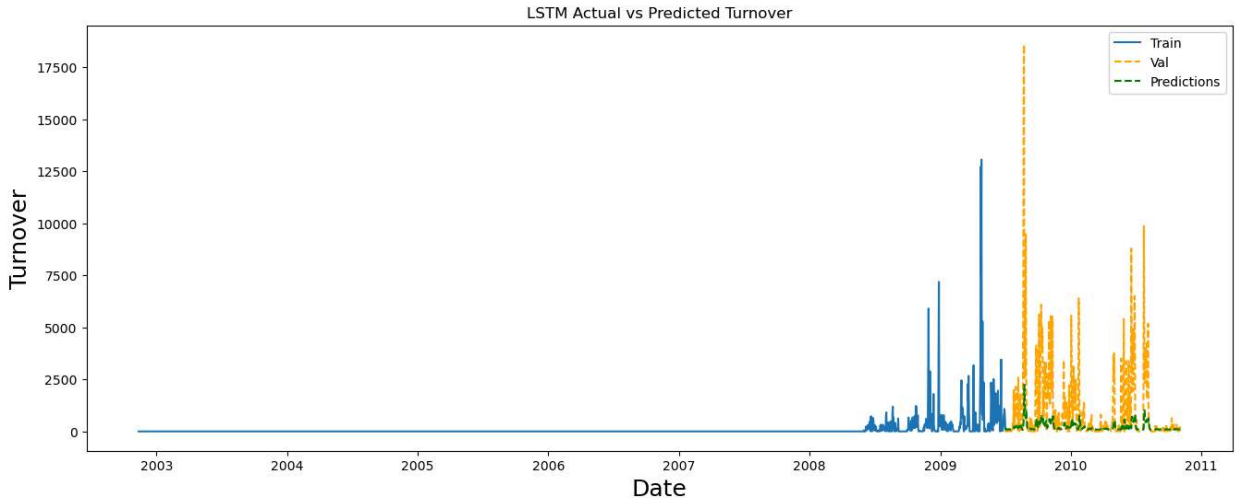
```
PG started right before: 2009-07-04 00:00:00
PG started right before: 2009-07-08 00:00:00
PG started right before: 2009-07-09 00:00:00
PG started right before: 2009-07-10 00:00:00
PG started right before: 2009-07-11 00:00:00
PG started right before: 2009-07-12 00:00:00
PG started right before: 2009-07-18 00:00:00
PG started right before: 2009-07-19 00:00:00
PG started right before: 2009-07-23 00:00:00
PG started right before: 2009-07-24 00:00:00
PG started right before: 2009-07-25 00:00:00
PG started right before: 2009-07-26 00:00:00
PG started right before: 2009-08-01 00:00:00
PG started right before: 2009-08-04 00:00:00
PG started right before: 2009-08-05 00:00:00
PG started right before: 2009-08-08 00:00:00
PG started right before: 2009-08-09 00:00:00
PG started right before: 2009-08-11 00:00:00
PG started right before: 2009-08-20 00:00:00
PG started right before: 2009-08-21 00:00:00
PG started right before: 2009-08-22 00:00:00
PG started right before: 2009-08-27 00:00:00
PG started right before: 2009-08-28 00:00:00
PG started right before: 2009-08-29 00:00:00
PG started right before: 2009-09-02 00:00:00
PG started right before: 2009-09-04 00:00:00
PG started right before: 2009-09-05 00:00:00
PG started right before: 2009-09-06 00:00:00
PG started right before: 2009-09-08 00:00:00
PG started right before: 2009-09-09 00:00:00
PG started right before: 2009-09-10 00:00:00
PG started right before: 2009-09-11 00:00:00
PG started right before: 2009-09-22 00:00:00
PG started right before: 2009-09-23 00:00:00
PG started right before: 2009-09-24 00:00:00
PG started right before: 2009-09-25 00:00:00
PG started right before: 2009-09-28 00:00:00
PG started right before: 2009-09-29 00:00:00
PG started right before: 2009-10-01 00:00:00
PG started right before: 2009-10-03 00:00:00
PG started right before: 2009-10-06 00:00:00
PG started right before: 2009-10-08 00:00:00
PG started right before: 2009-10-09 00:00:00
PG started right before: 2009-10-10 00:00:00
PG started right before: 2009-10-11 00:00:00
PG started right before: 2009-10-15 00:00:00
PG started right before: 2009-10-16 00:00:00
PG started right before: 2009-10-19 00:00:00
PG started right before: 2009-10-20 00:00:00
PG started right before: 2009-10-21 00:00:00
PG started right before: 2009-10-22 00:00:00
PG started right before: 2009-10-23 00:00:00
PG started right before: 2009-10-25 00:00:00
PG started right before: 2009-10-30 00:00:00
PG started right before: 2009-11-01 00:00:00
PG started right before: 2009-11-05 00:00:00
PG started right before: 2009-11-06 00:00:00
PG started right before: 2009-11-10 00:00:00
PG started right before: 2009-11-19 00:00:00
PG started right before: 2009-11-20 00:00:00
```

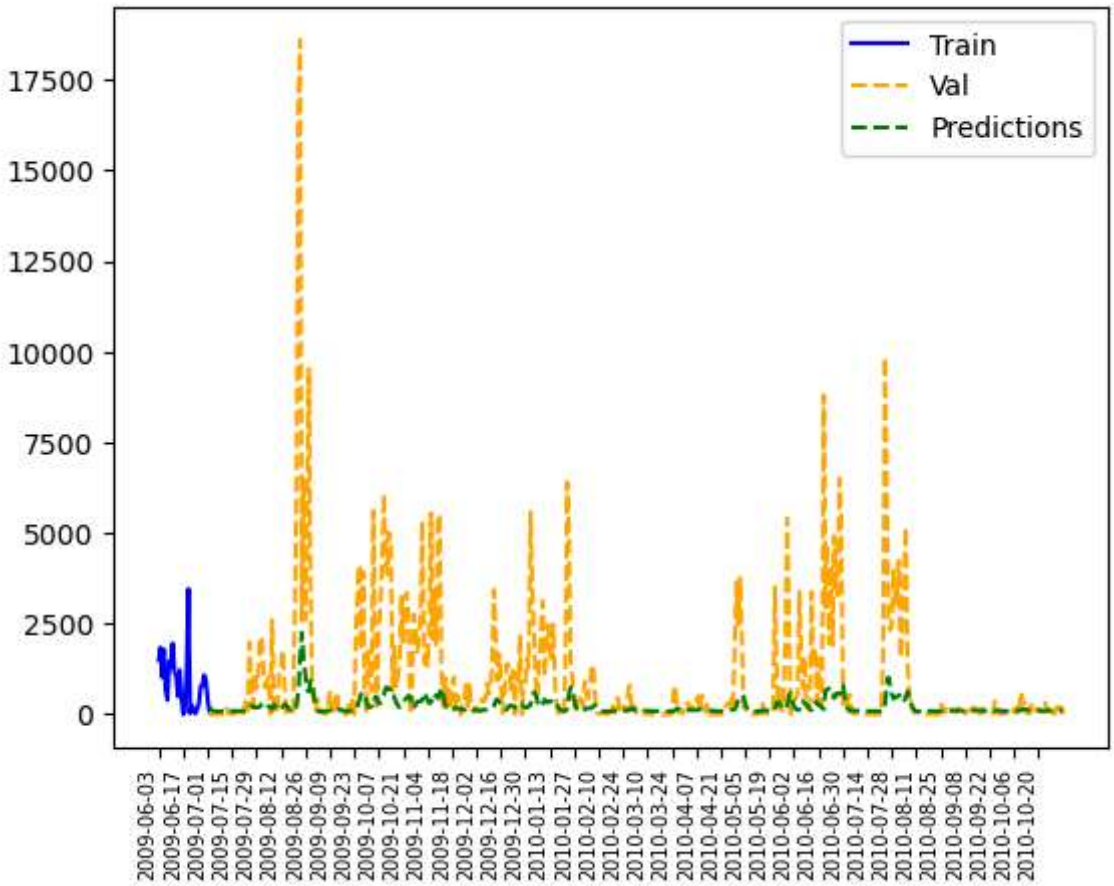
```
PG started right before: 2009-11-21 00:00:00
PG started right before: 2009-11-22 00:00:00
PG started right before: 2009-11-24 00:00:00
PG started right before: 2009-11-26 00:00:00
PG started right before: 2009-11-28 00:00:00
PG started right before: 2009-12-02 00:00:00
PG started right before: 2009-12-03 00:00:00
PG started right before: 2009-12-04 00:00:00
PG started right before: 2009-12-05 00:00:00
PG started right before: 2009-12-08 00:00:00
PG started right before: 2009-12-09 00:00:00
PG started right before: 2009-12-14 00:00:00
PG started right before: 2009-12-15 00:00:00
PG started right before: 2009-12-19 00:00:00
PG started right before: 2009-12-22 00:00:00
PG started right before: 2009-12-27 00:00:00
PG started right before: 2009-12-28 00:00:00
PG started right before: 2009-12-29 00:00:00
PG started right before: 2010-01-01 00:00:00
PG started right before: 2010-01-10 00:00:00
PG started right before: 2010-01-12 00:00:00
PG started right before: 2010-01-13 00:00:00
PG started right before: 2010-01-21 00:00:00
PG started right before: 2010-01-22 00:00:00
PG started right before: 2010-01-23 00:00:00
PG started right before: 2010-01-24 00:00:00
PG started right before: 2010-01-27 00:00:00
PG started right before: 2010-01-28 00:00:00
PG started right before: 2010-02-23 00:00:00
PG started right before: 2010-03-01 00:00:00
PG started right before: 2010-03-02 00:00:00
PG started right before: 2010-03-05 00:00:00
PG started right before: 2010-03-06 00:00:00
PG started right before: 2010-03-07 00:00:00
PG started right before: 2010-03-08 00:00:00
PG started right before: 2010-03-09 00:00:00
PG started right before: 2010-03-11 00:00:00
PG started right before: 2010-03-12 00:00:00
PG started right before: 2010-03-23 00:00:00
PG started right before: 2010-03-24 00:00:00
PG started right before: 2010-03-26 00:00:00
PG started right before: 2010-04-04 00:00:00
PG started right before: 2010-04-05 00:00:00
PG started right before: 2010-04-08 00:00:00
PG started right before: 2010-04-20 00:00:00
PG started right before: 2010-04-22 00:00:00
PG started right before: 2010-04-23 00:00:00
PG started right before: 2010-04-24 00:00:00
PG started right before: 2010-04-29 00:00:00
PG started right before: 2010-05-03 00:00:00
PG started right before: 2010-05-04 00:00:00
PG started right before: 2010-05-07 00:00:00
PG started right before: 2010-05-08 00:00:00
PG started right before: 2010-05-19 00:00:00
PG started right before: 2010-05-20 00:00:00
PG started right before: 2010-05-21 00:00:00
PG started right before: 2010-05-26 00:00:00
PG started right before: 2010-05-27 00:00:00
PG started right before: 2010-05-28 00:00:00
PG started right before: 2010-05-29 00:00:00
```

```
PG started right before: 2010-06-01 00:00:00
PG started right before: 2010-06-10 00:00:00
PG started right before: 2010-06-11 00:00:00
PG started right before: 2010-06-12 00:00:00
PG started right before: 2010-06-20 00:00:00
PG started right before: 2010-06-21 00:00:00
PG started right before: 2010-06-22 00:00:00
PG started right before: 2010-06-23 00:00:00
PG started right before: 2010-06-24 00:00:00
PG started right before: 2010-06-26 00:00:00
PG started right before: 2010-06-30 00:00:00
PG started right before: 2010-07-10 00:00:00
PG started right before: 2010-07-11 00:00:00
PG started right before: 2010-07-24 00:00:00
PG started right before: 2010-07-27 00:00:00
PG started right before: 2010-07-28 00:00:00
PG started right before: 2010-08-01 00:00:00
PG started right before: 2010-08-02 00:00:00
PG started right before: 2010-08-03 00:00:00
PG started right before: 2010-08-20 00:00:00
PG started right before: 2010-08-21 00:00:00
PG started right before: 2010-08-22 00:00:00
PG started right before: 2010-08-23 00:00:00
PG started right before: 2010-08-24 00:00:00
PG started right before: 2010-08-26 00:00:00
PG started right before: 2010-09-01 00:00:00
PG started right before: 2010-09-02 00:00:00
PG started right before: 2010-09-03 00:00:00
PG started right before: 2010-09-04 00:00:00
PG started right before: 2010-09-09 00:00:00
PG started right before: 2010-09-10 00:00:00
PG started right before: 2010-09-22 00:00:00
PG started right before: 2010-09-24 00:00:00
PG started right before: 2010-09-30 00:00:00
PG started right before: 2010-10-01 00:00:00
PG started right before: 2010-10-02 00:00:00
PG started right before: 2010-10-05 00:00:00
PG started right before: 2010-10-07 00:00:00
PG started right before: 2010-10-08 00:00:00
PG started right before: 2010-10-09 00:00:00
PG started right before: 2010-10-13 00:00:00
PG started right before: 2010-10-19 00:00:00
PG started right before: 2010-10-20 00:00:00
PG started right before: 2010-10-23 00:00:00
PG started right before: 2010-10-24 00:00:00
PG started right before: 2010-10-29 00:00:00
PG started right before: 2010-10-30 00:00:00
```




```
Mean: 238.77552505147563
Standard Deviation: 980.4562732027541
Variance: 961294.5036626337
1934/1934 [==============================] - 261s 133ms/step - loss: 0.0010
16/16 [==============================] - 3s 139ms/step
Test RMSE: 2064.926045109493
Test MAE: 878.5353481787856
```



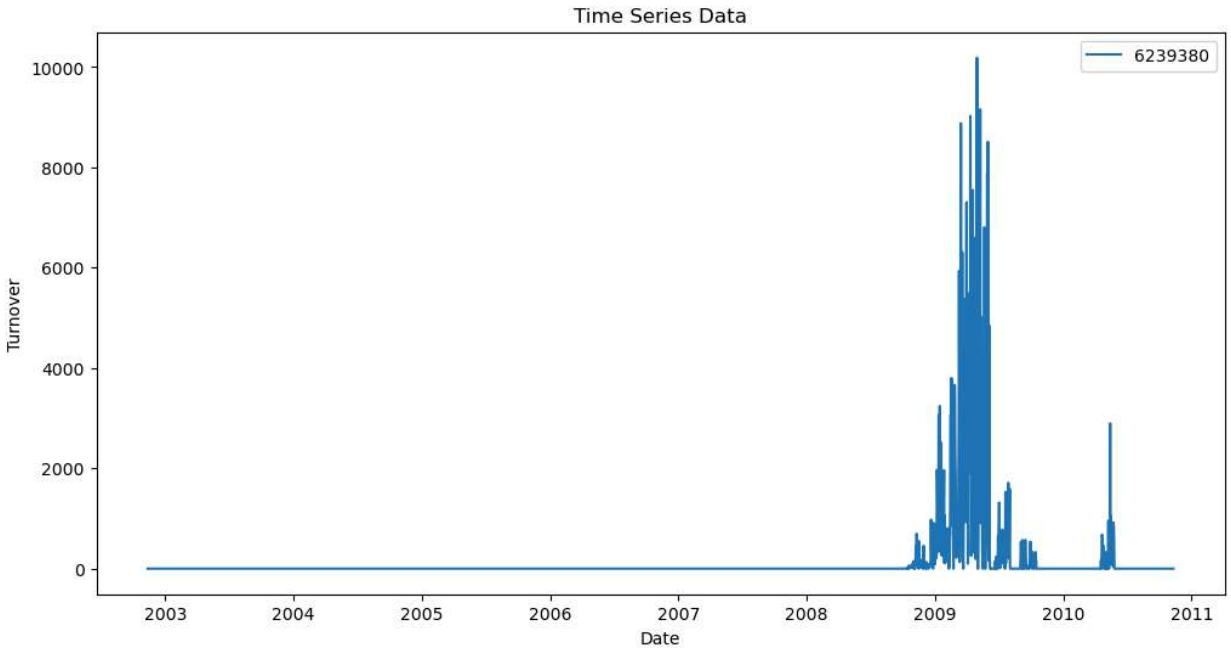

Train
Val
Predictions
17500
15000
12500
10000
7500
5000
2500
0
2009-06-03
2009-06-17
2009-07-01
2009-07-15
2009-07-29
2009-08-12
2009-08-26
2009-09-09
2009-09-23
2009-10-07
2009-10-21
2009-11-04
2009-11-18
2009-12-02
2009-12-16
2009-12-30
2010-01-13
2010-01-27
2010-02-10
2010-02-24
2010-03-10
2010-03-24
2010-04-07
2010-04-21
2010-05-05
2010-05-19
2010-06-02
2010-06-16
2010-06-30
2010-07-14
2010-07-28
2010-08-11
2010-08-25
2010-09-08
2010-09-22
2010-10-06
2010-10-20

```
PG started right before: 2009-07-06 00:00:00
PG started right before: 2009-07-07 00:00:00
PG started right before: 2009-07-08 00:00:00
PG started right before: 2009-07-09 00:00:00
PG started right before: 2009-07-10 00:00:00
PG started right before: 2009-07-11 00:00:00
PG started right before: 2009-07-12 00:00:00
PG started right before: 2009-07-18 00:00:00
PG started right before: 2009-07-22 00:00:00
PG started right before: 2009-07-23 00:00:00
PG started right before: 2009-07-25 00:00:00
PG started right before: 2009-07-26 00:00:00
PG started right before: 2009-07-28 00:00:00
PG started right before: 2009-07-30 00:00:00
PG started right before: 2009-07-31 00:00:00
PG started right before: 2009-08-01 00:00:00
PG started right before: 2009-08-07 00:00:00
PG started right before: 2009-08-11 00:00:00
PG started right before: 2009-08-12 00:00:00
PG started right before: 2009-08-13 00:00:00
PG started right before: 2009-08-20 00:00:00
PG started right before: 2009-08-21 00:00:00
PG started right before: 2009-08-22 00:00:00
PG started right before: 2009-08-23 00:00:00
PG started right before: 2009-08-28 00:00:00
PG started right before: 2009-09-04 00:00:00
PG started right before: 2009-09-05 00:00:00
PG started right before: 2009-09-06 00:00:00
PG started right before: 2009-09-08 00:00:00
PG started right before: 2009-09-09 00:00:00
PG started right before: 2009-09-10 00:00:00
PG started right before: 2009-09-13 00:00:00
PG started right before: 2009-09-20 00:00:00
PG started right before: 2009-09-21 00:00:00
PG started right before: 2009-09-22 00:00:00
PG started right before: 2009-09-24 00:00:00
PG started right before: 2009-09-25 00:00:00
PG started right before: 2009-09-26 00:00:00
PG started right before: 2009-09-28 00:00:00
PG started right before: 2009-10-01 00:00:00
PG started right before: 2009-10-02 00:00:00
PG started right before: 2009-10-04 00:00:00
PG started right before: 2009-10-08 00:00:00
PG started right before: 2009-10-09 00:00:00
PG started right before: 2009-10-10 00:00:00
PG started right before: 2009-10-11 00:00:00
PG started right before: 2009-10-13 00:00:00
PG started right before: 2009-10-19 00:00:00
PG started right before: 2009-10-20 00:00:00
PG started right before: 2009-10-21 00:00:00
PG started right before: 2009-10-22 00:00:00
PG started right before: 2009-10-23 00:00:00
PG started right before: 2009-10-27 00:00:00
PG started right before: 2009-10-28 00:00:00
PG started right before: 2009-10-29 00:00:00
PG started right before: 2009-10-30 00:00:00
PG started right before: 2009-10-31 00:00:00
PG started right before: 2009-11-01 00:00:00
PG started right before: 2009-11-05 00:00:00
PG started right before: 2009-11-06 00:00:00
```

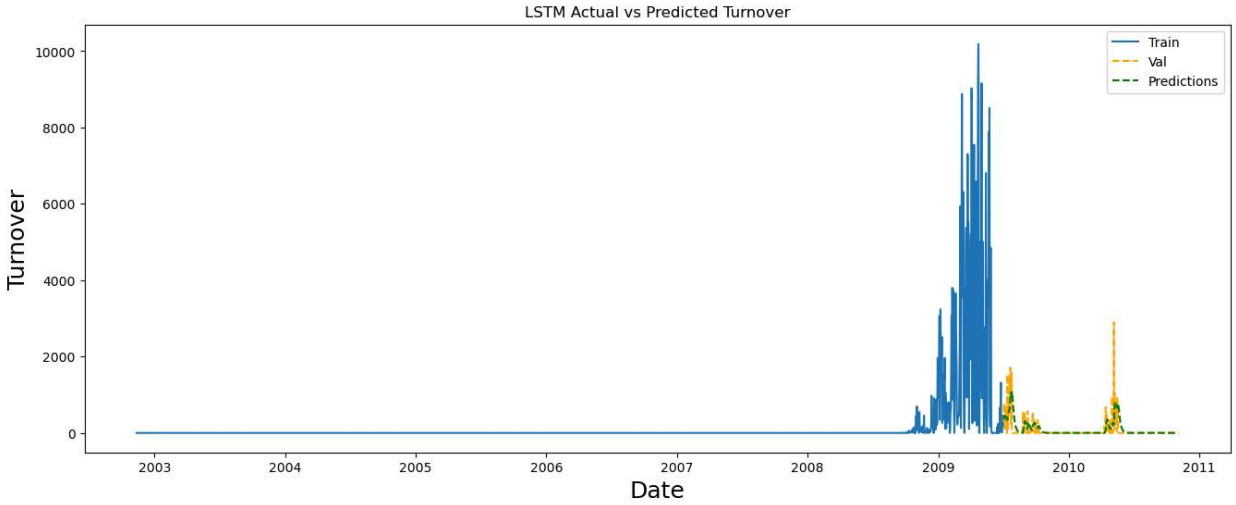
```
PG started right before: 2009-11-08 00:00:00
PG started right before: 2009-11-10 00:00:00
PG started right before: 2009-11-11 00:00:00
PG started right before: 2009-11-16 00:00:00
PG started right before: 2009-11-18 00:00:00
PG started right before: 2009-11-19 00:00:00
PG started right before: 2009-11-24 00:00:00
PG started right before: 2009-11-25 00:00:00
PG started right before: 2009-11-26 00:00:00
PG started right before: 2009-11-28 00:00:00
PG started right before: 2009-12-03 00:00:00
PG started right before: 2009-12-05 00:00:00
PG started right before: 2009-12-06 00:00:00
PG started right before: 2009-12-07 00:00:00
PG started right before: 2009-12-08 00:00:00
PG started right before: 2009-12-09 00:00:00
PG started right before: 2009-12-10 00:00:00
PG started right before: 2009-12-11 00:00:00
PG started right before: 2009-12-12 00:00:00
PG started right before: 2009-12-13 00:00:00
PG started right before: 2009-12-19 00:00:00
PG started right before: 2009-12-20 00:00:00
PG started right before: 2009-12-21 00:00:00
PG started right before: 2009-12-23 00:00:00
PG started right before: 2009-12-25 00:00:00
PG started right before: 2009-12-26 00:00:00
PG started right before: 2009-12-27 00:00:00
PG started right before: 2009-12-31 00:00:00
PG started right before: 2010-01-01 00:00:00
PG started right before: 2010-01-02 00:00:00
PG started right before: 2010-01-03 00:00:00
PG started right before: 2010-01-08 00:00:00
PG started right before: 2010-01-09 00:00:00
PG started right before: 2010-01-12 00:00:00
PG started right before: 2010-01-15 00:00:00
PG started right before: 2010-01-21 00:00:00
PG started right before: 2010-01-22 00:00:00
PG started right before: 2010-01-23 00:00:00
PG started right before: 2010-01-24 00:00:00
PG started right before: 2010-01-30 00:00:00
PG started right before: 2010-01-31 00:00:00
PG started right before: 2010-02-02 00:00:00
PG started right before: 2010-02-03 00:00:00
PG started right before: 2010-02-05 00:00:00
PG started right before: 2010-02-06 00:00:00
PG started right before: 2010-02-07 00:00:00
PG started right before: 2010-02-13 00:00:00
PG started right before: 2010-02-14 00:00:00
PG started right before: 2010-02-15 00:00:00
PG started right before: 2010-02-16 00:00:00
PG started right before: 2010-02-17 00:00:00
PG started right before: 2010-02-18 00:00:00
PG started right before: 2010-02-20 00:00:00
PG started right before: 2010-02-21 00:00:00
PG started right before: 2010-02-24 00:00:00
PG started right before: 2010-02-27 00:00:00
PG started right before: 2010-02-28 00:00:00
PG started right before: 2010-03-06 00:00:00
PG started right before: 2010-03-07 00:00:00
PG started right before: 2010-03-25 00:00:00
```

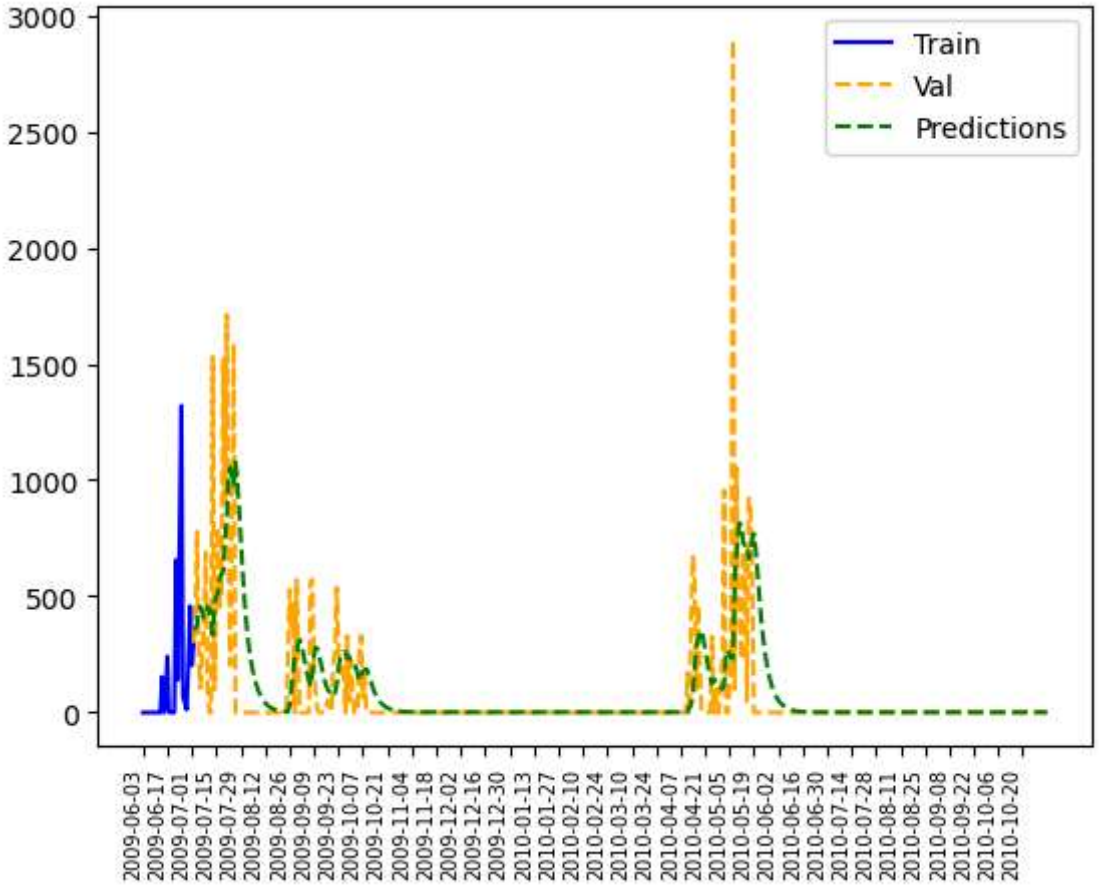
```
PG started right before: 2010-03-26 00:00:00
PG started right before: 2010-03-31 00:00:00
PG started right before: 2010-04-01 00:00:00
PG started right before: 2010-04-02 00:00:00
PG started right before: 2010-04-07 00:00:00
PG started right before: 2010-04-08 00:00:00
PG started right before: 2010-04-10 00:00:00
PG started right before: 2010-04-13 00:00:00
PG started right before: 2010-04-23 00:00:00
PG started right before: 2010-04-24 00:00:00
PG started right before: 2010-04-25 00:00:00
PG started right before: 2010-04-26 00:00:00
PG started right before: 2010-04-27 00:00:00
PG started right before: 2010-04-29 00:00:00
PG started right before: 2010-04-30 00:00:00
PG started right before: 2010-05-01 00:00:00
PG started right before: 2010-05-02 00:00:00
PG started right before: 2010-05-09 00:00:00
PG started right before: 2010-05-10 00:00:00
PG started right before: 2010-05-11 00:00:00
PG started right before: 2010-05-12 00:00:00
PG started right before: 2010-05-13 00:00:00
PG started right before: 2010-05-15 00:00:00
PG started right before: 2010-05-21 00:00:00
PG started right before: 2010-05-22 00:00:00
PG started right before: 2010-05-23 00:00:00
PG started right before: 2010-05-28 00:00:00
PG started right before: 2010-05-29 00:00:00
PG started right before: 2010-05-30 00:00:00
PG started right before: 2010-06-04 00:00:00
PG started right before: 2010-06-05 00:00:00
PG started right before: 2010-06-06 00:00:00
PG started right before: 2010-06-12 00:00:00
PG started right before: 2010-06-13 00:00:00
PG started right before: 2010-06-14 00:00:00
PG started right before: 2010-06-19 00:00:00
PG started right before: 2010-06-20 00:00:00
PG started right before: 2010-06-21 00:00:00
PG started right before: 2010-06-25 00:00:00
PG started right before: 2010-06-26 00:00:00
PG started right before: 2010-06-28 00:00:00
PG started right before: 2010-06-29 00:00:00
PG started right before: 2010-07-04 00:00:00
PG started right before: 2010-07-22 00:00:00
PG started right before: 2010-07-23 00:00:00
PG started right before: 2010-07-24 00:00:00
PG started right before: 2010-07-25 00:00:00
PG started right before: 2010-07-29 00:00:00
PG started right before: 2010-07-30 00:00:00
PG started right before: 2010-07-31 00:00:00
PG started right before: 2010-08-01 00:00:00
PG started right before: 2010-08-04 00:00:00
PG started right before: 2010-08-05 00:00:00
PG started right before: 2010-08-12 00:00:00
PG started right before: 2010-08-13 00:00:00
PG started right before: 2010-08-14 00:00:00
PG started right before: 2010-08-16 00:00:00
PG started right before: 2010-08-25 00:00:00
PG started right before: 2010-08-26 00:00:00
PG started right before: 2010-08-29 00:00:00
```

```
PG started right before: 2010-09-04 00:00:00
PG started right before: 2010-09-09 00:00:00
PG started right before: 2010-09-10 00:00:00
PG started right before: 2010-09-11 00:00:00
PG started right before: 2010-09-12 00:00:00
PG started right before: 2010-09-13 00:00:00
PG started right before: 2010-09-14 00:00:00
PG started right before: 2010-09-15 00:00:00
PG started right before: 2010-09-18 00:00:00
PG started right before: 2010-09-23 00:00:00
PG started right before: 2010-09-24 00:00:00
PG started right before: 2010-09-25 00:00:00
PG started right before: 2010-10-02 00:00:00
PG started right before: 2010-10-03 00:00:00
PG started right before: 2010-10-05 00:00:00
PG started right before: 2010-10-06 00:00:00
PG started right before: 2010-10-08 00:00:00
PG started right before: 2010-10-10 00:00:00
PG started right before: 2010-10-11 00:00:00
PG started right before: 2010-10-16 00:00:00
PG started right before: 2010-10-17 00:00:00
PG started right before: 2010-10-20 00:00:00
PG started right before: 2010-10-24 00:00:00
PG started right before: 2010-10-27 00:00:00
PG started right before: 2010-10-30 00:00:00
PG started right before: 2010-10-31 00:00:00
PG started right before: 2010-11-02 00:00:00
```

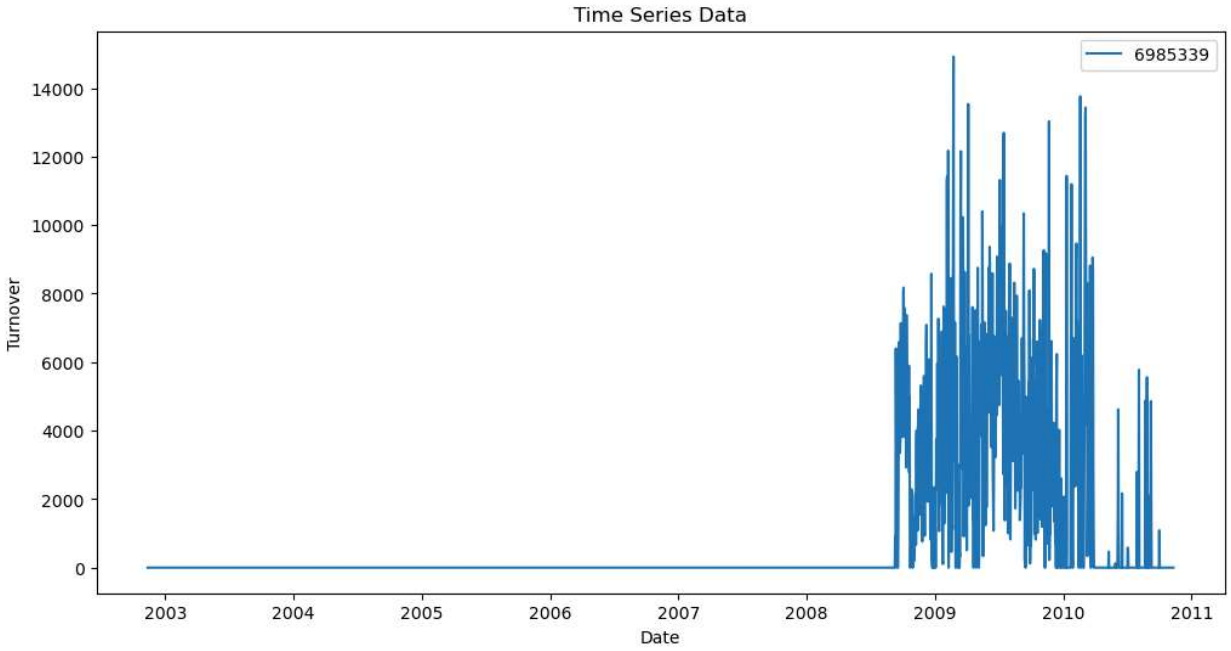



```
Mean: 147.61094715168153
Standard Deviation: 780.2908435877481
Variance: 608853.8005868795
1934/1934 [==============================] - 310s 158ms/step - loss: 0.0058
16/16 [==============================] - 3s 134ms/step
Test RMSE: 322.38036411034545
Test MAE: 86.72363813992908
```

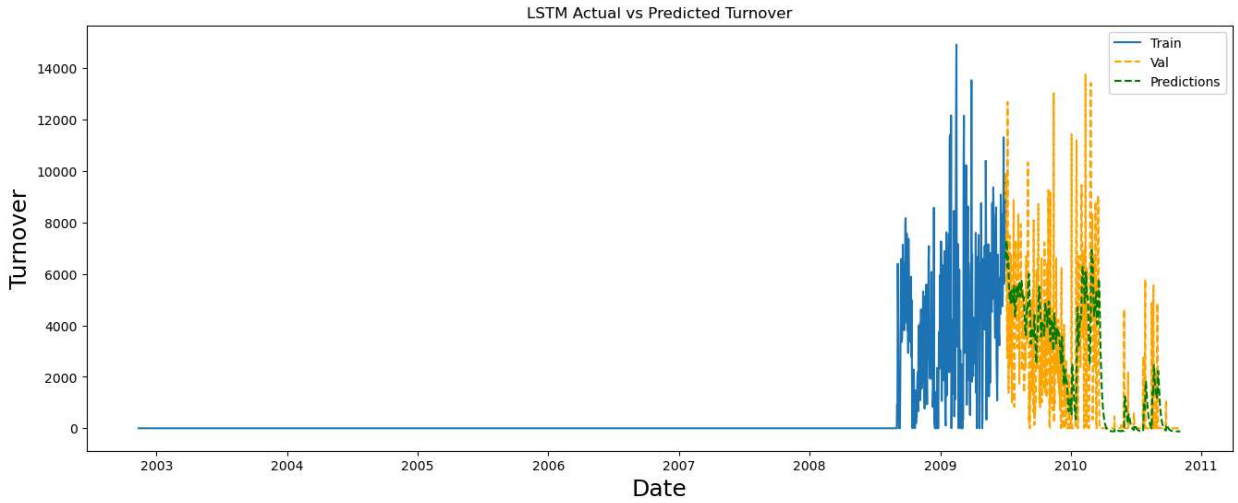

LSTM Actual vs Predicted Turnover
Train
Val
Predictions
Turnover
Date
0
2000
4000
6000
8000
10000
2003
2004
2005
2006
2007
2008
2009
2010
2011

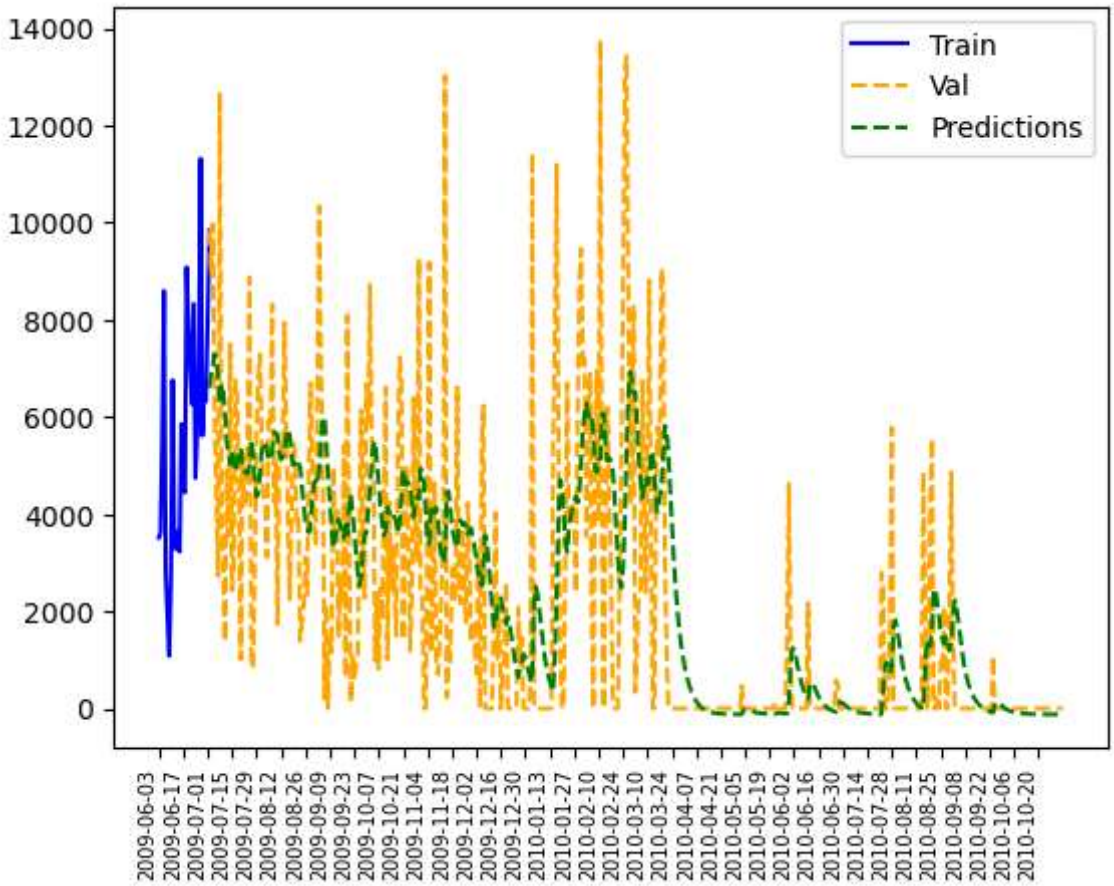

Train
Val
Predictions
0
500
1000
1500
2000
2500
3000
2009-06-03
2009-06-17
2009-07-01
2009-07-15
2009-07-29
2009-08-12
2009-08-26
2009-09-09
2009-09-23
2009-10-07
2009-10-21
2009-11-04
2009-11-18
2009-12-02
2009-12-16
2009-12-30
2010-01-13
2010-01-27
2010-02-10
2010-02-24
2010-03-10
2010-03-24
2010-04-07
2010-04-21
2010-05-05
2010-05-19
2010-06-02
2010-06-16
2010-06-30
2010-07-14
2010-07-28
2010-08-11
2010-08-25
2010-09-08
2010-09-22
2010-10-06
2010-10-20

```
PG started right before: 2009-07-03 00:00:00
PG started right before: 2009-07-04 00:00:00
PG started right before: 2009-07-05 00:00:00
PG started right before: 2009-07-08 00:00:00
PG started right before: 2009-07-09 00:00:00
PG started right before: 2009-07-13 00:00:00
PG started right before: 2009-07-14 00:00:00
PG started right before: 2009-07-15 00:00:00
PG started right before: 2009-07-16 00:00:00
PG started right before: 2009-07-17 00:00:00
PG started right before: 2009-07-18 00:00:00
PG started right before: 2009-07-19 00:00:00
PG started right before: 2009-07-20 00:00:00
PG started right before: 2009-07-21 00:00:00
PG started right before: 2009-07-22 00:00:00
PG started right before: 2009-07-25 00:00:00
PG started right before: 2009-08-25 00:00:00
PG started right before: 2009-08-26 00:00:00
PG started right before: 2009-08-27 00:00:00
PG started right before: 2009-08-28 00:00:00
PG started right before: 2009-08-29 00:00:00
PG started right before: 2009-08-30 00:00:00
PG started right before: 2009-08-31 00:00:00
PG started right before: 2009-09-07 00:00:00
PG started right before: 2009-09-08 00:00:00
PG started right before: 2009-09-09 00:00:00
PG started right before: 2009-09-20 00:00:00
PG started right before: 2009-09-21 00:00:00
PG started right before: 2009-09-22 00:00:00
PG started right before: 2009-09-23 00:00:00
PG started right before: 2009-09-24 00:00:00
PG started right before: 2009-09-25 00:00:00
PG started right before: 2009-09-28 00:00:00
PG started right before: 2009-10-05 00:00:00
PG started right before: 2009-10-06 00:00:00
PG started right before: 2009-10-07 00:00:00
PG started right before: 2009-10-08 00:00:00
PG started right before: 2010-04-11 00:00:00
PG started right before: 2010-04-12 00:00:00
PG started right before: 2010-04-13 00:00:00
PG started right before: 2010-04-14 00:00:00
PG started right before: 2010-04-15 00:00:00
PG started right before: 2010-04-16 00:00:00
PG started right before: 2010-04-17 00:00:00
PG started right before: 2010-04-18 00:00:00
PG started right before: 2010-04-25 00:00:00
PG started right before: 2010-05-01 00:00:00
PG started right before: 2010-05-02 00:00:00
PG started right before: 2010-05-03 00:00:00
PG started right before: 2010-05-07 00:00:00
PG started right before: 2010-05-08 00:00:00
PG started right before: 2010-05-09 00:00:00
PG started right before: 2010-05-10 00:00:00
PG started right before: 2010-05-14 00:00:00
PG started right before: 2010-05-16 00:00:00
PG started right before: 2010-05-17 00:00:00
PG started right before: 2010-05-18 00:00:00
```




```
Mean: 761.8822115991763
Standard Deviation: 2004.6482107288575
Variance: 4018614.44877841
1934/1934 [==============================] - 253s 129ms/step - loss: 0.0061
16/16 [==============================] - 3s 119ms/step
Test RMSE: 3639.499538276778
Test MAE: 1516.2712461583662
```




Train
Val
Predictions
14000
12000
10000
8000
6000
4000
2000
0
2009-06-03
2009-06-17
2009-07-01
2009-07-15
2009-07-29
2009-08-12
2009-08-26
2009-09-09
2009-09-23
2009-10-07
2009-10-21
2009-11-04
2009-11-18
2009-12-02
2009-12-16
2009-12-30
2010-01-13
2010-01-27
2010-02-10
2010-02-24
2010-03-10
2010-03-24
2010-04-07
2010-04-21
2010-05-05
2010-05-19
2010-06-02
2010-06-16
2010-06-30
2010-07-14
2010-07-28
2010-08-11
2010-08-25
2010-09-08
2010-09-22
2010-10-06
2010-10-20

```
PG started right before: 2009-07-03 00:00:00
PG started right before: 2009-07-04 00:00:00
PG started right before: 2009-07-08 00:00:00
PG started right before: 2009-07-14 00:00:00
PG started right before: 2009-07-17 00:00:00
PG started right before: 2009-07-18 00:00:00
PG started right before: 2009-07-19 00:00:00
PG started right before: 2009-07-24 00:00:00
PG started right before: 2009-07-25 00:00:00
PG started right before: 2009-07-29 00:00:00
PG started right before: 2009-07-30 00:00:00
PG started right before: 2009-07-31 00:00:00
PG started right before: 2009-08-01 00:00:00
PG started right before: 2009-08-05 00:00:00
PG started right before: 2009-08-06 00:00:00
PG started right before: 2009-08-07 00:00:00
PG started right before: 2009-08-12 00:00:00
PG started right before: 2009-08-13 00:00:00
PG started right before: 2009-08-14 00:00:00
PG started right before: 2009-08-15 00:00:00
PG started right before: 2009-08-19 00:00:00
PG started right before: 2009-08-20 00:00:00
PG started right before: 2009-08-28 00:00:00
PG started right before: 2009-08-29 00:00:00
PG started right before: 2009-08-30 00:00:00
PG started right before: 2009-08-31 00:00:00
PG started right before: 2009-09-02 00:00:00
PG started right before: 2009-09-03 00:00:00
PG started right before: 2009-09-04 00:00:00
PG started right before: 2009-09-05 00:00:00
PG started right before: 2009-09-11 00:00:00
PG started right before: 2009-09-12 00:00:00
PG started right before: 2009-09-13 00:00:00
PG started right before: 2009-09-17 00:00:00
PG started right before: 2009-09-19 00:00:00
PG started right before: 2009-09-20 00:00:00
PG started right before: 2009-09-26 00:00:00
PG started right before: 2009-09-27 00:00:00
PG started right before: 2009-09-28 00:00:00
PG started right before: 2009-09-29 00:00:00
PG started right before: 2009-09-30 00:00:00
PG started right before: 2009-10-01 00:00:00
PG started right before: 2009-10-02 00:00:00
PG started right before: 2009-10-03 00:00:00
PG started right before: 2009-10-04 00:00:00
PG started right before: 2009-10-10 00:00:00
PG started right before: 2009-10-11 00:00:00
PG started right before: 2009-10-13 00:00:00
PG started right before: 2009-10-18 00:00:00
PG started right before: 2009-10-19 00:00:00
PG started right before: 2009-10-20 00:00:00
PG started right before: 2009-10-23 00:00:00
PG started right before: 2009-10-27 00:00:00
PG started right before: 2009-10-28 00:00:00
PG started right before: 2009-10-30 00:00:00
PG started right before: 2009-11-05 00:00:00
PG started right before: 2009-11-08 00:00:00
PG started right before: 2009-11-13 00:00:00
PG started right before: 2009-11-14 00:00:00
PG started right before: 2009-11-15 00:00:00
```

```
PG started right before: 2009-11-21 00:00:00
PG started right before: 2009-11-22 00:00:00
PG started right before: 2009-11-24 00:00:00
PG started right before: 2009-11-27 00:00:00
PG started right before: 2009-11-28 00:00:00
PG started right before: 2009-12-03 00:00:00
PG started right before: 2009-12-05 00:00:00
PG started right before: 2009-12-06 00:00:00
PG started right before: 2009-12-13 00:00:00
PG started right before: 2009-12-14 00:00:00
PG started right before: 2009-12-15 00:00:00
PG started right before: 2009-12-19 00:00:00
PG started right before: 2009-12-26 00:00:00
PG started right before: 2009-12-27 00:00:00
PG started right before: 2009-12-28 00:00:00
PG started right before: 2010-01-03 00:00:00
PG started right before: 2010-01-04 00:00:00
PG started right before: 2010-01-15 00:00:00
PG started right before: 2010-01-16 00:00:00
PG started right before: 2010-01-17 00:00:00
PG started right before: 2010-01-18 00:00:00
PG started right before: 2010-01-23 00:00:00
PG started right before: 2010-01-24 00:00:00
PG started right before: 2010-01-25 00:00:00
PG started right before: 2010-01-26 00:00:00
PG started right before: 2010-01-27 00:00:00
PG started right before: 2010-01-29 00:00:00
PG started right before: 2010-01-30 00:00:00
PG started right before: 2010-01-31 00:00:00
PG started right before: 2010-02-01 00:00:00
PG started right before: 2010-02-02 00:00:00
PG started right before: 2010-02-05 00:00:00
PG started right before: 2010-02-09 00:00:00
PG started right before: 2010-02-10 00:00:00
PG started right before: 2010-02-11 00:00:00
PG started right before: 2010-02-12 00:00:00
PG started right before: 2010-02-15 00:00:00
PG started right before: 2010-02-23 00:00:00
PG started right before: 2010-02-24 00:00:00
PG started right before: 2010-02-25 00:00:00
PG started right before: 2010-02-26 00:00:00
PG started right before: 2010-02-27 00:00:00
PG started right before: 2010-03-02 00:00:00
PG started right before: 2010-03-07 00:00:00
PG started right before: 2010-03-08 00:00:00
PG started right before: 2010-03-10 00:00:00
PG started right before: 2010-03-11 00:00:00
PG started right before: 2010-03-12 00:00:00
PG started right before: 2010-03-16 00:00:00
PG started right before: 2010-03-17 00:00:00
PG started right before: 2010-03-18 00:00:00
PG started right before: 2010-03-19 00:00:00
PG started right before: 2010-05-03 00:00:00
PG started right before: 2010-05-04 00:00:00
PG started right before: 2010-05-22 00:00:00
PG started right before: 2010-05-23 00:00:00
PG started right before: 2010-05-29 00:00:00
PG started right before: 2010-05-30 00:00:00
PG started right before: 2010-05-31 00:00:00
PG started right before: 2010-06-10 00:00:00
```

```
PG started right before: 2010-06-11 00:00:00
PG started right before: 2010-06-26 00:00:00
PG started right before: 2010-06-27 00:00:00
PG started right before: 2010-06-28 00:00:00
PG started right before: 2010-07-22 00:00:00
PG started right before: 2010-07-23 00:00:00
PG started right before: 2010-07-24 00:00:00
PG started right before: 2010-07-27 00:00:00
PG started right before: 2010-07-28 00:00:00
PG started right before: 2010-07-29 00:00:00
PG started right before: 2010-08-15 00:00:00
PG started right before: 2010-08-16 00:00:00
PG started right before: 2010-08-17 00:00:00
PG started right before: 2010-08-19 00:00:00
PG started right before: 2010-08-20 00:00:00
PG started right before: 2010-08-21 00:00:00
PG started right before: 2010-08-27 00:00:00
PG started right before: 2010-08-30 00:00:00
PG started right before: 2010-08-31 00:00:00
PG started right before: 2010-09-01 00:00:00
PG started right before: 2010-09-24 00:00:00
PG started right before: 2010-09-25 00:00:00
```