

Exploratory Data Analysis and Feature Engineering: Python a great tool to use.

Pycon Nigeria 2019 workshop

While working on a particular data set many think the path to having a great model is through parameter tuning and optimizing the model. It is very common that the key is to actually take time to understand the data, perform exploratory data analysis and feature engineering to generate a dataset that is ready for the model. In this hands-on tutorial, a financial case study will be used to explain the rudiments of exploratory data analysis and feature engineering. The case study is centered around a fintech company that wants to provide its customers with a paid mobile app subscription which allows them to track their financial activities. The final aim is to access the customer's app behavior from the data, this will help the company target some users who are interested in their services. Python libraries such as numpy, pandas, matplotlib, seaborn, etc will be used to achieve the goal of this tutorial. In the end, a worthy model will be developed from a cleaned dataset. The dataset is readily available for this tutorial.

Tutorial Objectives

- Describe the data
- Clean the data
- Visualizations
- Calculate and visualize correlations
- Feature Engineering

Import Libraries

```
In [1]: import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from dateutil import parser
import warnings
warnings.filterwarnings("ignore")
```

Import dataset

```
In [2]: eda = pd.read_csv('appdata10.csv')
```

Descriptive Statistics

The first important step is to calculate some descriptive statistics for the data. Descriptive statistical analysis helps to describe basic features of a dataset and obtains a short summary of the sample and measures of the data.

```
In [3]: eda.describe()
Out[3]:
```

	user	dayofweek	age	numscreens	minigame	used_premium_feature	enrolled
count	50000.000000	50000.000000	50000.000000	50000.000000	50000.000000	50000.000000	50000.000000
mean	186889.729900	3.029860	31.72436	21.095900	0.107820	0.172020	0.621480
std	107768.520361	2.031997	10.80331	15.728812	0.310156	0.377402	0.485023
min	13.000000	0.000000	16.00000	1.000000	0.000000	0.000000	0.000000
25%	93526.750000	1.000000	24.00000	10.000000	0.000000	0.000000	0.000000
50%	187193.500000	3.000000	29.00000	18.000000	0.000000	0.000000	1.000000
75%	279984.250000	5.000000	37.00000	28.000000	0.000000	0.000000	1.000000
max	373662.000000	6.000000	101.00000	325.000000	1.000000	1.000000	1.000000

The above table gives us the summary of data. It gives the mean, the total data points, standard deviation, the quartiles and the max and min (extreme values). It gives a holistic view of the dataset. N.B: NaN values are not computed in this summary.

```
In [4]: eda.head()
Out[4]:
```

	user	first_open	dayofweek	hour	age	screen_list	numscreens	minigame
0	235136	2012-12-27 02:14:51.273	3	02:00:00	23	idscreen,joinscreen,Cycle,product_review,ScanP...	15	0
1	333588	2012-12-02 01:16:00.905	6	01:00:00	24	joinscreen,product_review,product_review2,Scan...	13	0
2	254414	2013-03-19 19:19:09.157	1	19:00:00	23	Splash,Cycle,Loan	3	0
3	234192	2013-07-05 16:08:46.354	4	16:00:00	28	product_review,Home,product_review,Loan3,Finan...	40	0
4	51549	2013-02-26 18:50:48.661	1	18:00:00	31	idscreen,joinscreen,Cycle,Credit3Container,Sca...	32	0

Data Shape.

Check the number of rows and columns N.B: Number on the left is the total rows and columns on the right: (rows, columns)

```
In [5]: eda.shape
Out[5]: (50000, 12)
```

From the above, we have 50000 rows and 12 columns. Meaning there are 12 features.

Data Profile

Check the data types of each columns.

```
In [6]: eda.info()
Out[6]:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 50000 entries, 0 to 49999
Data columns (total 12 columns):
user                50000 non-null int64
first_open          50000 non-null object
dayofweek           50000 non-null int64
hour                50000 non-null object
age                 50000 non-null int64
screen_list         50000 non-null object
numscreens          50000 non-null int64
minigame            50000 non-null int64
used_premium_feature 50000 non-null int64
enrolled            50000 non-null int64
enrolled_date       31074 non-null object
liked                50000 non-null int64
dtypes: int64(8), object(4)
memory usage: 4.6+ MB
```

We can see that the enrolled date column has some NaN values. The memory used by this dataframe is 4.6+ MB

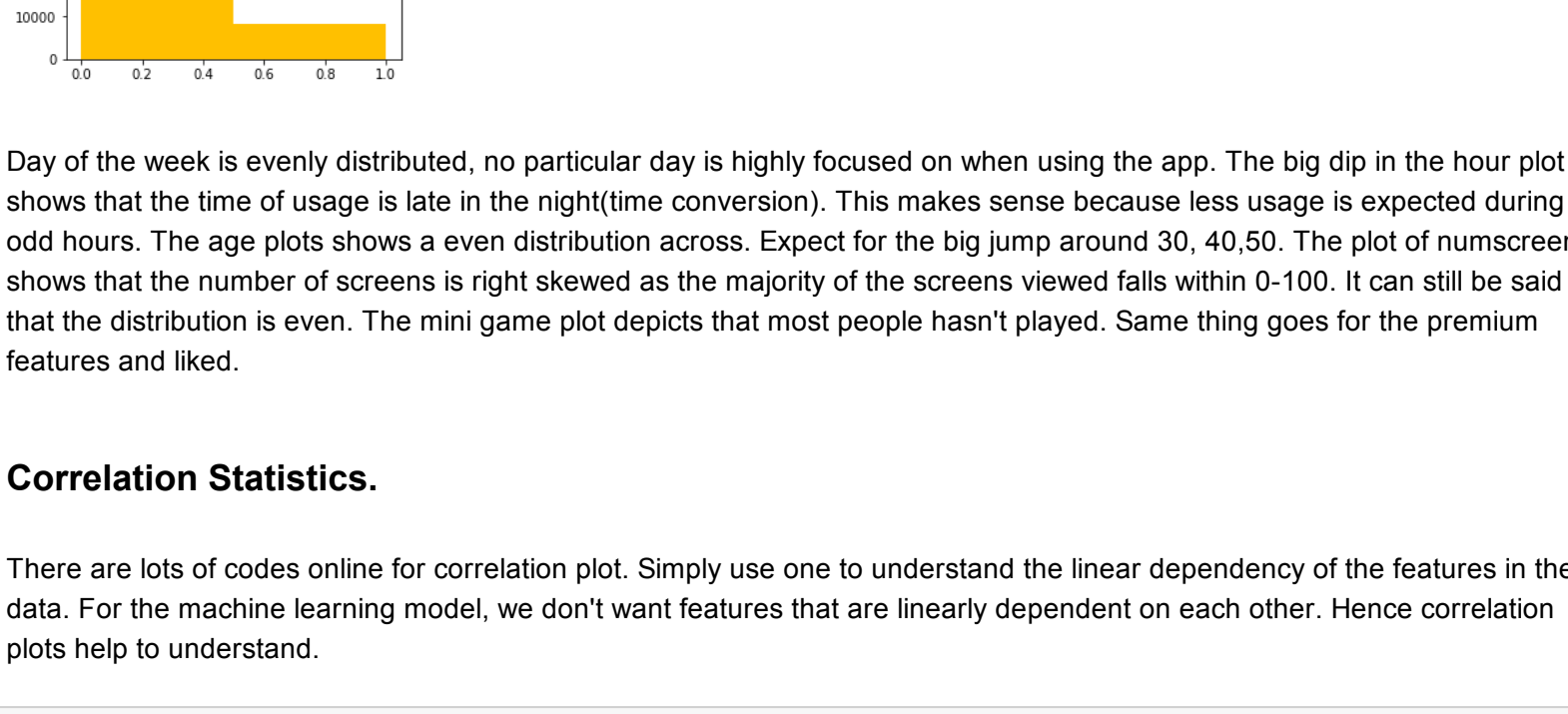
Data types

```
In [7]: eda.dtypes
Out[7]:
user                int64
first_open          object
dayofweek           int64
hour                object
age                 int64
screen_list         object
numscreens          int64
minigame            int64
used_premium_feature int64
enrolled            int64
enrolled_date       object
liked                int64
dtype: object
```

N.B: first_open, enrolled_date, hour will be our main focus in the feature engineering exercise. Simply because, we need to change the data types.

Visualizations

```
In [8]: eda['hour'] = eda.hour.str.slice(1,3).astype(int)
In [9]: visuals = eda.copy().drop(columns = ['user', 'first_open', 'screen_list', 'enrolled', 'enrolled_date'])
In [10]: for i in range(1, visuals.shape[1]+ 1):
plt.subplot(3, 3, i)
f = plt.gca()
f.set_title(visuals.columns.values[i -1])
#set Dims size
vals = np.size(visuals.iloc[:, i-1].unique())
plt.subplots_adjust(left = 0.4, bottom=0.1, right=3, top=2, wspace = 0.7, hspace= 0.4)
#the above is important to space the visuals
plt.hist(visuals.iloc[:, i-1], bins = vals, color = '#FFC000')
```



Day of the week is evenly distributed, no particular day is highly focused on when using the app. The big dip in the hour plot shows that the time of usage is late in the night(time conversion). This makes sense because less usage is expected during odd hours. The age plots shows a even distribution across. Expect for the big jump around 30, 40,50. The plot of numscreens shows that the number of screens is right skewed as the majority of the screens viewed falls within 0-100. It can still be said that the distribution is even. The mini game plot depicts that most people hasn't played. Same thing goes for the premium features and liked.

Correlation Statistics.

There are lots of codes online for correlation plot. Simply use one to understand the linear dependency of the features in the data. For the machine learning model, we don't want features that are linearly dependent on each other. Hence correlation plots help to understand.

```
In [11]: ## Correlation Matrix
sns.set(style="white", font_scale=2)

# Compute the correlation matrix
corr = visuals.corr()

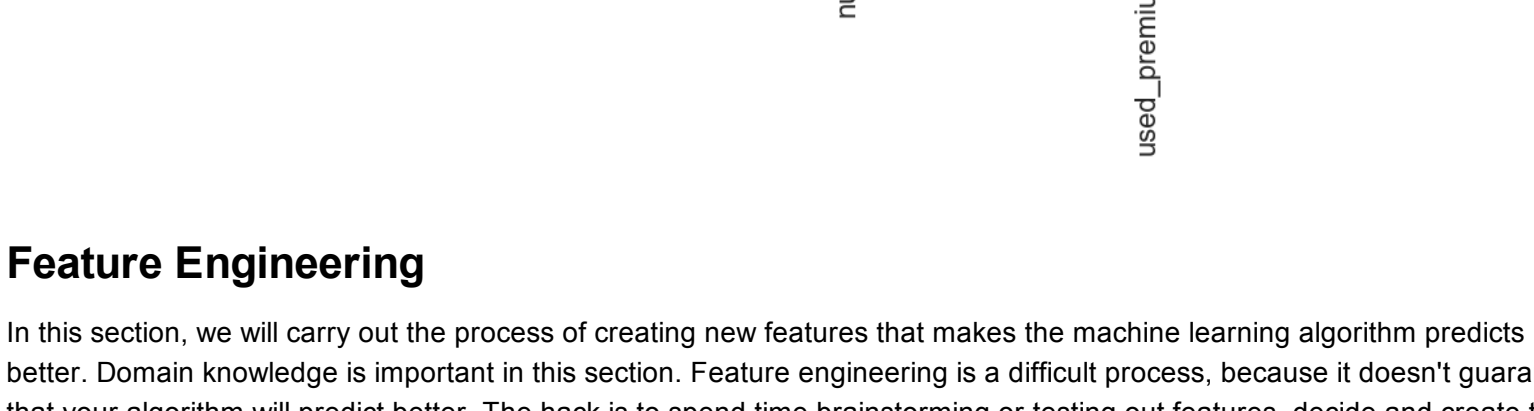
# Generate a mask for the upper triangle
mask = np.zeros_like(corr, dtype=np.bool)
mask[np.triu_indices_from(mask)] = True

# Set up the matplotlib figure
f, ax = plt.subplots(figsize=(18, 15))
f.suptitle("Correlation Matrix", fontsize = 40)

# Generate a custom diverging colormap
cmap = sns.diverging_palette(220, 10, as_cmap=True)

# Draw the heatmap with the mask and correct aspect ratio
sns.heatmap(corr, mask=mask, cmap=cmap, vmax=-3, center=0,
square=True, linewidths=.5, cbar_kws={"shrink": .5})
Out[11]: <matplotlib.axes._subplots.AxesSubplot at 0x1c902588be0>
```

Correlation Matrix



Feature Engineering

In this section, we will carry out the process of creating new features that makes the machine learning algorithm predicts better. Domain knowledge is important in this section. Feature engineering is a difficult process, because it doesn't guarantee that your algorithm will predict better. The hack is to spend time brainstorming or testing out features, decide and create the features, check how it works with the model, improve and go back to brainstorming/creating until the work is done.

FE for Date columns

```
In [12]: eda.dtypes
Out[12]:
user                int64
first_open          object
dayofweek           int64
hour                int32
age                 int64
screen_list         object
numscreens          int64
minigame            int64
used_premium_feature int64
enrolled            int64
enrolled_date       object
liked                int64
dtype: object
```

Parser module is used convert from the object data type to the date/time format. THe module offers a generic date/time string parser which is able to parse most known formats to represent date/time

```
In [13]: eda["first_open"] = [parser.parse(row_date) for row_date in eda["first_open"]]
eda["enrolled_date"] = [parser.parse(row_date) if isinstance (row_date, str) else row_date for row_date in eda["enrolled_date"]]
In [14]: eda.dtypes
Out[14]:
user                int64
first_open          datetime64[ns]
dayofweek           int64
hour                int32
age                 int64
screen_list         object
numscreens          int64
minigame            int64
used_premium_feature int64
enrolled            int64
enrolled_date       datetime64[ns]
liked                int64
dtype: object
```

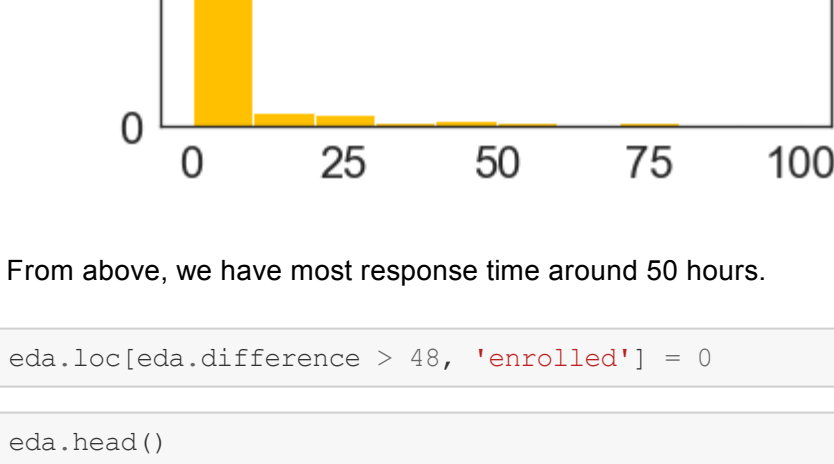
The date columns are in the proper data type formats.

Create Response Time

```
In [15]: eda["difference"] = (eda.enrolled_date - eda.first_open).astype('timedelta64[h]')
```

Displot of response time

```
In [16]: #eda['difference'].dropna().hist(bins=10, grid=True, xlabelsize=12, ylabelsize=12)
In [17]: plt.hist(eda['difference'].dropna(), color = '#FFC000', range = [0,100])
plt.title("Distribution plot for response time")
plt.show()
```



From above, we have most response time around 50 hours.

```
In [18]: eda.loc[eda.difference > 48, 'enrolled'] = 0
In [19]: eda.head()
Out[19]:
```

	user	first_open	dayofweek	hour	age	screen_list	numscreens	minigame	use
0	235136	2012-12-27 02:14:51.273	3	2	23	idscreen,joinscreen,Cycle,product_review,ScanP...	15	0	
1	333588	2012-12-02 01:16:00.905	6	1	24	joinscreen,product_review,product_review2,Scan...	13	0	
2	254414	2013-03-19 19:19:09.157	1	19	23	Splash,Cycle,Loan	3	0	
3	234192	2013-07-05 16:08:46.354	4	16	28	product_review,Home,product_review,Loan3,Finan...	40	0	
4	51549	2013-02-26 18:50:48.661	1	18	31	idscreen,joinscreen,Cycle,Credit3Container,Sca...	32	0	

In [20]: eda = eda.drop(columns = ['enrolled_date', 'difference', 'first_open'])

Creating screen groupings

```
In [21]: eda_screens = pd.read_csv('top_screens.csv').top_screens.values
In [22]: eda['screen_list'] = eda.screen_list.astype(str) + ','
In [23]: for sc in top_screens:
eda[sc] = eda.screen_list.str.contains(sc).astype(int)
eda['screen_list'] = eda.screen_list.str.replace(sc,"", "")
eda['other'] = eda.screen_list.str.count(",")
eda = eda.drop(columns = ['screen_list'])
In [24]: eda.head()
Out[24]:
```

	user	dayofweek	hour	age	numscreens	minigame	used_premium_feature	enrolled	liked	Loan2	Loan3	Loan4	Loan5	Loan6	Loan7	Loan8	Loan9	Loan10	Loan11	Loan12	Loan13	Loan14	Loan15	Loan16	Loan17	Loan18	Loan19	Loan20	Loan21	Loan22	Loan23	Loan24	Loan25	Loan26	Loan27	Loan28	Loan29	Loan30	Loan31	Loan32	Loan33	Loan34	Loan35	Loan36	Loan37	Loan38	Loan39	Loan40	Loan41	Loan42	Loan43	Loan44	Loan45	Loan46	Loan47	Loan48	Loan49	Loan50	Loan51	Loan52	Loan53	Loan54	Loan55	Loan56	Loan57	Loan58	Loan59	Loan60	Loan61	Loan62	Loan63	Loan64	Loan65	Loan66	Loan67	Loan68	Loan69	Loan70	Loan71	Loan72	Loan73	Loan74	Loan75	Loan76	Loan77	Loan78	Loan79	Loan80	Loan81	Loan82	Loan83	Loan84	Loan85	Loan86	Loan87	Loan88	Loan89	Loan90	Loan91	Loan92	Loan93	Loan94	Loan95	Loan96	Loan97	Loan98	Loan99	Loan100	Loan101	Loan102	Loan103	Loan104	Loan105	Loan106	Loan107	Loan108	Loan109	Loan110	Loan111	Loan112	Loan113	Loan114	Loan115	Loan116	Loan117	Loan118	Loan119	Loan120	Loan121	Loan122	Loan123	Loan124	Loan125	Loan126	Loan127	Loan128	Loan129	Loan130	Loan131	Loan132	Loan133	Loan134	Loan135	Loan136	Loan137	Loan138	Loan139	Loan140	Loan141	Loan142	Loan143	Loan144	Loan145	Loan146	Loan147	Loan148	Loan149	Loan150	Loan151	Loan152	Loan153	Loan154	Loan155	Loan156	Loan157	Loan158	Loan159	Loan160	Loan161	Loan162	Loan163	Loan164	Loan165	Loan166	Loan167	Loan168	Loan169	Loan170	Loan171	Loan172	Loan173	Loan174	Loan175	Loan176	Loan177	Loan178	Loan179	Loan180	Loan181	Loan182	Loan183	Loan184	Loan185	Loan186	Loan187	Loan188	Loan189	Loan190	Loan191	Loan192	Loan193	Loan194	Loan195	Loan196	Loan197	Loan198	Loan199	Loan200	Loan201	Loan202	Loan203	Loan204	Loan205	Loan206	Loan207	Loan208	Loan209	Loan210	Loan211	Loan212	Loan213	Loan214	Loan215	Loan216	Loan217	Loan218	Loan219	Loan220	Loan221	Loan222	Loan223	Loan224	Loan225	Loan226	Loan227	Loan228	Loan229	Loan230	Loan231	Loan232	Loan233	Loan234	Loan235	Loan236	Loan237	Loan238	Loan239	Loan240	Loan241	Loan242	Loan243	Loan244	Loan245	Loan246	Loan247	Loan248	Loan249	Loan250	Loan251	Loan252	Loan253	Loan254	Loan255	Loan256	Loan257	Loan258	Loan259	Loan260	Loan261	Loan262	Loan263	Loan264	Loan265	Loan266	Loan267	Loan268	Loan269	Loan270	Loan271	Loan272	Loan273	Loan274	Loan275	Loan276	Loan277	Loan278	Loan279	Loan280	Loan281	Loan282	Loan283	Loan284	Loan285	Loan286	Loan287	Loan288	Loan289	Loan290	Loan291	Loan292	Loan293	Loan294	Loan295	Loan296	Loan297	Loan298	Loan299	Loan300	Loan301	Loan302	Loan303	Loan304	Loan305	Loan306	Loan307	Loan308	Loan309	Loan310	Loan311	Loan312	Loan313	Loan314	Loan315	Loan316	Loan317	Loan318	Loan319	Loan320	Loan321	Loan322	Loan323	Loan324	Loan325	Loan326	Loan327	Loan328	Loan329	Loan330	Loan331	Loan332	Loan333	Loan334	Loan335	Loan336	Loan337	Loan338	Loan339	Loan340	Loan341	Loan342	Loan343	Loan344	Loan345	Loan346	Loan347	Loan348	Loan349	Loan350	Loan351	Loan352	Loan353	Loan354	Loan355	Loan356	Loan357	Loan358	Loan359	Loan360	Loan361	Loan362	Loan363	Loan364	Loan365	Loan366	Loan367	Loan368	Loan369	Loan370	Loan371	Loan372	Loan373	Loan374	Loan375	Loan376	Loan377	Loan378	Loan379	Loan380	Loan381	Loan382	Loan383	Loan384	Loan385	Loan386	Loan387	Loan388	Loan389	Loan390	Loan391	Loan392	Loan393	Loan394	Loan395	Loan396	Loan397	Loan398	Loan399	Loan400	Loan401	Loan402	Loan403	Loan404	Loan405	Loan406	Loan407	Loan408	Loan409	Loan410	Loan411	Loan412	Loan413	Loan414	Loan415	Loan416	Loan417	Loan418	Loan419	Loan420	Loan421	Loan422	Loan423	Loan424	Loan425	Loan426	Loan427	Loan428	Loan429	Loan430	Loan431	Loan432	Loan433	Loan434	Loan435	Loan436	Loan437	Loan438	Loan439	Loan440	Loan441	Loan442	Loan443	Loan444	Loan445	Loan446	Loan447	Loan448	Loan449	Loan450	Loan451	Loan452	Loan453	Loan454	Loan455	Loan456	Loan457	Loan458	Loan459	Loan460	Loan461	Loan462	Loan463	Loan464	Loan465	Loan466	Loan467	Loan468	Loan469	Loan470	Loan471	Loan472	Loan473	Loan474	Loan475	Loan476	Loan477	Loan478	Loan479	Loan480	Loan481	Loan482	Loan483	Loan484	Loan485	Loan486	Loan487	Loan488	Loan489	Loan490	Loan491	Loan492	Loan493	Loan494	Loan495	Loan496	Loan497	Loan498	Loan499	Loan500	Loan501	Loan502	Loan503	Loan504	Loan505	Loan506	Loan507	Loan508	Loan509	Loan510	Loan511	Loan512	Loan513	Loan514	Loan515	Loan516	Loan517	Loan518	Loan519	Loan520	Loan521	Loan522	Loan523	Loan524	Loan525	Loan526	Loan527	Loan528	Loan529	Loan530	Loan531	Loan532	Loan533	Loan534	Loan535	Loan536	Loan537	Loan538	Loan539	Loan540	Loan541	Loan542	Loan543	Loan544	Loan545	Loan546	Loan547	Loan548	Loan549	Loan550	Loan551	Loan552	Loan553	Loan554	Loan555	Loan556	Loan557	Loan558	Loan559	Loan560	Loan561	Loan562	Loan563	Loan564	Loan565	Loan566	Loan567	Loan568	Loan569	Loan570	Loan571	Loan572	Loan573	Loan574	Loan575	Loan576	Loan577	Loan578	Loan579	Loan580	Loan581	Loan582	Loan583	Loan584	Loan585	Loan586	Loan587	Loan588	Loan589	Loan590	Loan591	Loan592	Loan593	Loan594	Loan595	Loan596	Loan597	Loan598	Loan599	Loan600	Loan601	Loan602	Loan603	Loan604	Loan605	Loan606	Loan607	Loan608	Loan609	Loan610	Loan611	Loan612	Loan613	Loan614	Loan615	Loan616	Loan617	Loan618	Loan619	Loan620	Loan621	Loan622	Loan623	Loan624	Loan625	Loan626	Loan627	Loan628	Loan629	Loan630	Loan631	Loan632	Loan633	Loan634	Loan635	Loan636	Loan637	Loan638	Loan639	Loan640	Loan641	Loan642	Loan643	Loan644	Loan645	Loan646
--	------	-----------	------	-----	------------	----------	----------------------	----------	-------	-------	-------	-------	-------	-------	-------	-------	-------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------