

```
In [1]: import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import os
import datetime
import warnings
from sklearn.model_selection import train_test_split, GridSearchCV

from sklearn.linear_model import LogisticRegression
from sklearn.neighbors import KNeighborsClassifier
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier, AdaBoostClassifier, GradientBoostingClassifier
from sklearn import metrics, preprocessing
warnings.filterwarnings(action='ignore')
%matplotlib inline
```

```
In [2]: df = pd.read_excel('URG_ASMA.XLSX', header='infer')
```

```
In [3]: df_asma = df[df["asma"] != 0]
```

```
In [4]: df_asma.shape
```

Out[4]: (1100, 21)

```
In [5]: # Check for the missing values
df_asma.isnull().sum(axis=0)
```

Out[5]: num_sequencial 0
sexo 0
idade 0
dicofre 0
distrito 0
concelho 0
freguesia 0
regiao 0
nacionalidade 0
data_admissao 0
colorid 1
cor 1
cod_proveniencia 0
proveniencia 0
cod_causa 0
causa 0
cod_destino 0
destino 0
diagnostico 0
classe 0
asma 0
dtype: int64

```
In [6]: df_asma['colorid'] = df_asma['colorid'].fillna(df_asma['colorid'].mode()[0])
df_asma['cor'] = df_asma['cor'].fillna(df_asma['cor'].mode()[0])
```

```
In [7]: # Check for the missing values
df_asma.isnull().sum(axis=0)
```

Out[7]: num_sequencial 0
sexo 0
idade 0
dicofre 0
distrito 0
concelho 0
freguesia 0
regiao 0
nacionalidade 0
data_admissao 0
colorid 0
cor 0
cod_proveniencia 0
proveniencia 0
cod_causa 0
causa 0
cod_destino 0
destino 0
diagnostico 0
classe 0
asma 0
dtype: int64

```
In [8]: # Type of data
df_asma.dtypes
```

Out[8]: num_sequencial int64
sexo int64
idade int64
dicofre int64
distrito object
concelho object
freguesia object
regiao object
nacionalidade object
data_admissao datetime64[ns]
colorid float64
cor object
cod_proveniencia int64
proveniencia object
cod_causa int64
causa object
cod_destino int64
destino object
diagnostico object
classe object
asma int64
dtype: object

```
In [9]: df_asma['year'] = pd.DatetimeIndex(df_asma['data_admissao']).year
```

```
In [10]: df_asma['month'] = pd.DatetimeIndex(df_asma['data_admissao']).month
```

```
In [11]: df_asma['year_month_day'] = df_asma['data_admissao'].dt.strftime('%Y-%m-%d')
```

```
In [12]: df_asma.head()
```

	num_sequencial	sexo	idade	dicofre	distrito	concelho	freguesia	regiao	nacionalidade	data_admissao	...	cod_causa	causa	cod_destino	destino	diagnostico	classe	asma	year	month	year_month_day
77949	372731	1	35	111009	LISBOA	OEIRAS	PORTO SALVO	LISBOA E VALE DO TEJO	PORTUGAL	2019-01-01 16:08:00	...	5	DOENCA	5	EXTERIOR NAO REFERENCIADO	493 - ASMA	asma	1	2019	1	2019-01-01
77950	945004	1	11	111012	LISBOA	OEIRAS	ALGÉS	LISBOA E VALE DO TEJO	BRASIL	2019-01-01 20:12:00	...	5	DOENCA	2	CONSULTA EXTERNA	493 - ASMA	asma	1	2019	1	2019-01-01
77951	1156492	1	5	111012	LISBOA	OEIRAS	ALGÉS	LISBOA E VALE DO TEJO	PORTUGAL	2019-01-01 21:44:00	...	5	DOENCA	12	ARS/CENTRO DE SAUDE	493 - ASMA	asma	1	2019	1	2019-01-01
77952	1156493	2	45	110658	LISBOA	LISBOA	BELÉM	LISBOA E VALE DO TEJO	PORTUGAL	2019-01-01 21:45:00	...	5	DOENCA	5	EXTERIOR NAO REFERENCIADO	493 - ASMA	asma	1	2019	1	2019-01-01
77953	114619	1	26	110658	LISBOA	LISBOA	BELÉM	LISBOA E VALE DO TEJO	PORTUGAL	2019-01-01 21:56:00	...	5	DOENCA	5	EXTERIOR NAO REFERENCIADO	493 - ASMA	asma	1	2019	1	2019-01-01

5 rows x 24 columns

```
In [13]: df_asma2 = df_asma.groupby(['year_month_day']).sum('asma')
print(df_asma2)
```

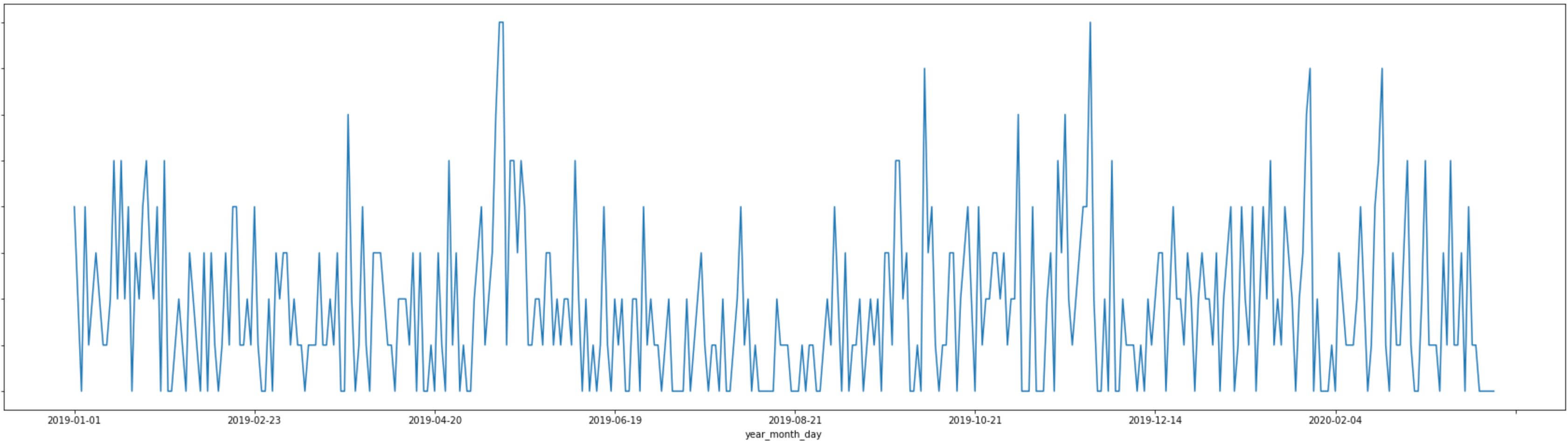
	num_sequencial	sexo	idade	dicofre	colorid	\
year_month_day						
2019-01-01	3745339	6	122	554349	13.0	
2019-01-02	2393795	4	41	332477	10.0	
2019-01-03	813432	1	7	110602	3.0	
2019-01-04	1863809	9	168	553822	15.0	
2019-01-05	663186	3	36	221634	5.0	
...	
2020-03-19	967114	2	58	111514	3.0	
2020-03-20	84631	1	28	111014	3.0	
2020-03-26	31042	1	26	110601	3.0	
2020-03-28	478131	2	17	111013	4.0	
2020-03-30	480846	1	16	111014	4.0	
	cod_proveniencia	cod_causa	cod_destino	asma	year	month
year_month_day						
2019-01-01	40	25	29	5	10095	5
2019-01-02	30	15	26	3	6057	3
2019-01-03	5	5	12	1	2019	1
2019-01-04	40	25	46	5	10095	5
2019-01-05	10	10	17	2	4038	2
...	
2020-03-19	5	5	5	1	2020	3
2020-03-20	5	5	5	1	2020	3
2020-03-26	20	5	5	1	2020	3
2020-03-28	5	5	5	1	2020	3
2020-03-30	20	5	5	1	2020	3

[395 rows x 11 columns]

```
In [14]: # Type of data
df_asma.dtypes
```

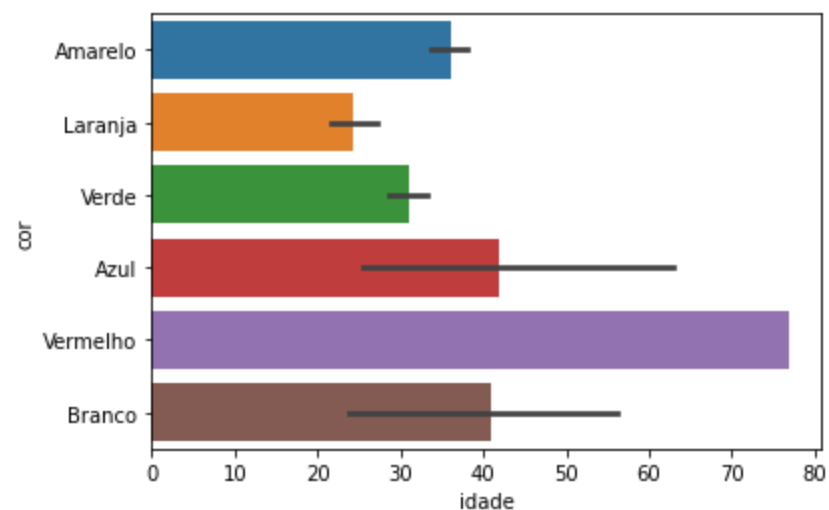
Out[14]: num_sequencial int64
sexo int64
idade int64
dicofre int64
distrito object
concelho object
freguesia object
regiao object
nacionalidade object
data_admissao datetime64[ns]
colorid float64
cor object
cod_proveniencia int64
proveniencia object
cod_causa int64
causa object
cod_destino int64
destino object
diagnostico object
classe object
asma int64
year int64
month int64
year_month_day object
dtype: object

```
In [15]: df_asma2['asma'].plot(figsize=(30, 8))
plt.show()
```



```
In [16]: sns.barplot(x='idade', y='cor', data=df_asma)
```

Out[16]: <AxesSubplot:xlabel='idade', ylabel='cor'>



```
In [ ]: sns.pairplot(df, hue = 'cor')
```

```
In [ ]: fig, ax = plt.subplots(figsize=(10, 8))
sns.distplot(df_asma.colorid)
```