In [1]:

```
import numpy as np
import pandas as pd
```

In [4]:

```
data=pd.read_csv("C:/Users/wit12/Downloads/skin cancer.csv")
```

In [5]:

```
data
```

Out[5]:

	lesion_id	image_id	dx	dx_type	age	sex	localization
0	HAM_0000118	ISIC_0027419	bkl	histo	80.0	male	scalp
1	HAM_0000118	ISIC_0025030	bkl	histo	80.0	male	scalp
2	HAM_0002730	ISIC_0026769	bkl	histo	80.0	male	scalp
3	HAM_0002730	ISIC_0025661	bkl	histo	80.0	male	scalp
4	HAM_0001466	ISIC_0031633	bkl	histo	75.0	male	ear
...	...	...	...	...	...	...	...
10010	HAM_0002867	ISIC_0033084	akiec	histo	40.0	male	abdomen
10011	HAM_0002867	ISIC_0033550	akiec	histo	40.0	male	abdomen
10012	HAM_0002867	ISIC_0033536	akiec	histo	40.0	male	abdomen
10013	HAM_0000239	ISIC_0032854	akiec	histo	80.0	male	face
10014	HAM_0003521	ISIC_0032258	mel	histo	70.0	female	back

10015 rows × 7 columns

In [6]:

```
# check if there is any null values in the data
```

In [9]:

```
data.isnull().sum()
```

Out[9]:

```
lesion_id        0
image_id         0
dx               0
dx_type          0
age             57
sex              0
localization     0
dtype: int64
```

In [11]:

```
data.size
```

Out[11]:

```
70105
```

In [14]:

```
data.dropna()
```

Out[14]:

	lesion_id	image_id	dx	dx_type	age	sex	localization
0	HAM_0000118	ISIC_0027419	bkl	histo	80.0	male	scalp
1	HAM_0000118	ISIC_0025030	bkl	histo	80.0	male	scalp
2	HAM_0002730	ISIC_0026769	bkl	histo	80.0	male	scalp
3	HAM_0002730	ISIC_0025661	bkl	histo	80.0	male	scalp
4	HAM_0001466	ISIC_0031633	bkl	histo	75.0	male	ear
...	...	...	...	...	...	...	...
10010	HAM_0002867	ISIC_0033084	akiec	histo	40.0	male	abdomen
10011	HAM_0002867	ISIC_0033550	akiec	histo	40.0	male	abdomen
10012	HAM_0002867	ISIC_0033536	akiec	histo	40.0	male	abdomen
10013	HAM_0000239	ISIC_0032854	akiec	histo	80.0	male	face
10014	HAM_0003521	ISIC_0032258	mel	histo	70.0	female	back

9958 rows × 7 columns

In [29]:

```
val=int(data['age'].mean())
data['age'].fillna(val,inplace=True)
```

In [30]:

```
data.isnull().sum()
```

Out[30]:

```
lesion_id       0
image_id        0
dx              0
dx_type         0
age             0
sex             0
localization    0
dtype: int64
```

In [31]:

```
data.shape
```

Out[31]:

```
(10015, 7)
```

In [32]:

```
data.columns
```

Out[32]:

```
Index(['lesion_id', 'image_id', 'dx', 'dx_type', 'age', 'sex', 'localizatio
n'], dtype='object')
```

In [36]:

```
# count no of male cancers and female cancers
cntmale=0;
cntfemale=0;
for i in data['sex']:
    if i== "male":
        cntmale+=1
    elif i== 'female':
        cntfemale+=1
print("No.of male cancers are :",cntmale,"No.of female cancers are:",cntfemale)
```

```
No.of male cancers are : 5406 No.of female cancers are: 4552
```

In [39]:

```
print(len(data[data['sex']=='male']))
print(len(data[data['sex']=='female']))
```

```
5406
4552
```

In [40]:

```
data['dx']
```

Out[40]:

```
0          bkl
1          bkl
2          bkl
3          bkl
4          bkl
         ...
10010    akiec
10011    akiec
10012    akiec
10013    akiec
10014      mel
Name: dx, Length: 10015, dtype: object
```

In [38]:

```
dx = data['dx'].value_counts().sort_index()
print(dx)
```

```
akiec     327
bcc       514
bkl      1099
df        115
mel      1113
nv       6705
vasc      142
Name: dx, dtype: int64
```

In [41]:

```
data.head()
```

Out[41]:

	lesion_id	image_id	dx	dx_type	age	sex	localization
0	HAM_0000118	ISIC_0027419	bkl	histo	80.0	male	scalp
1	HAM_0000118	ISIC_0025030	bkl	histo	80.0	male	scalp
2	HAM_0002730	ISIC_0026769	bkl	histo	80.0	male	scalp
3	HAM_0002730	ISIC_0025661	bkl	histo	80.0	male	scalp
4	HAM_0001466	ISIC_0031633	bkl	histo	75.0	male	ear

In [47]:

```
data['dx_type'].unique()
```

Out[47]:

```
array(['histo', 'consensus', 'confocal', 'follow_up'], dtype=object)
```

In [49]:

```
categories = dx.index.values
counts = dx.values
print(categories)
print(counts)
```

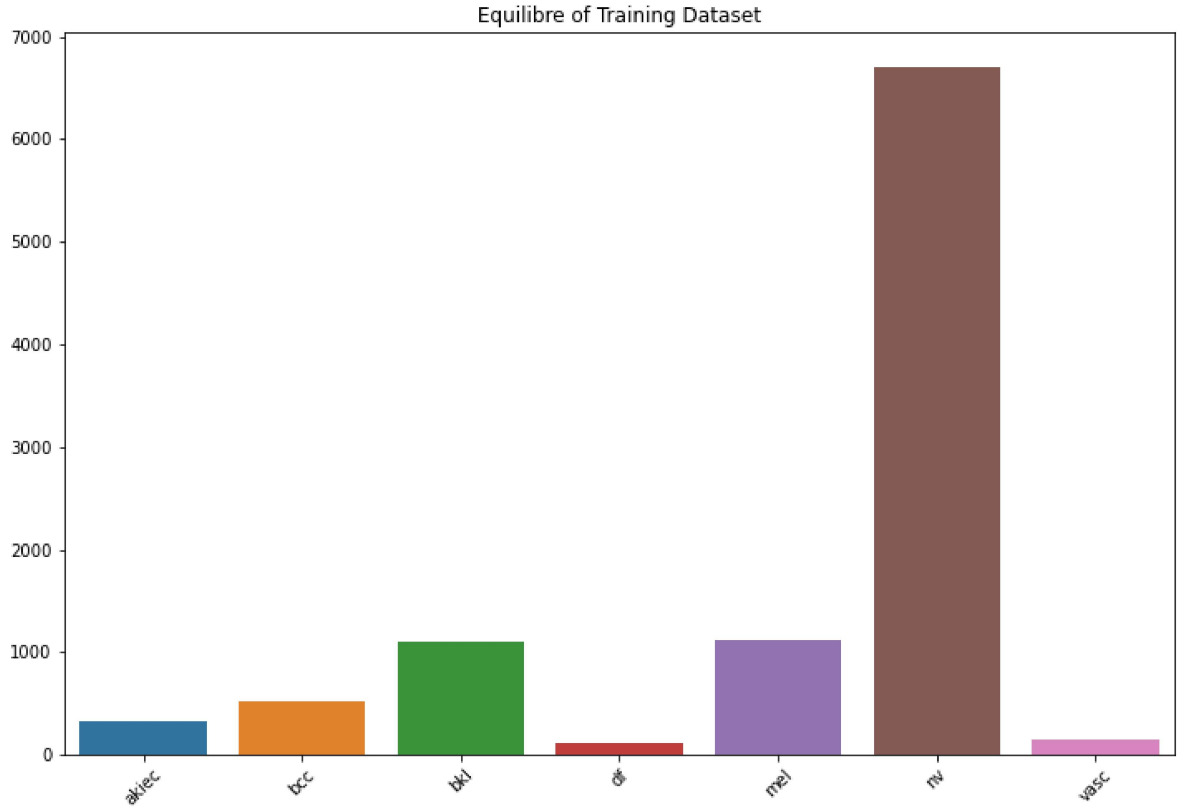
```
['akiec' 'bcc' 'bkl' 'df' 'mel' 'nv' 'vasc']
[ 327  514 1099  115 1113 6705  142]
```

In [50]:

```
import matplotlib.pyplot as plt
import seaborn as sns
#sns.set_style("whitegrid")

def plot_equilibre(categories, counts):

    plt.figure(figsize=(12, 8))

    sns_bar = sns.barplot(x=categories, y=counts)
    sns_bar.set_xticklabels(categories, rotation=45)
    plt.title('Equilibre of Training Dataset')
    plt.show()
```

In [51]:

```
plot_equilibre(categories, counts)
```



In [53]:

```
# training the model
from sklearn.model_selection import train_test_split
df_train, df_tmp = train_test_split(data, test_size=0.2, random_state=101, stratify=data['d
df_val, df_test = train_test_split(df_tmp, test_size=0.5, random_state=101)
```

In [ ]:

```

```