Machine Learning in Real World

- # Name: Ajinkya Ghadge
- # Email: aghadge@umass.edu
- # SPIRE ID: 32285077

Introduction

This class projects works towards gaining insights from the data that has been taken from a hospice. These insights can help the patients as well as the doctors who work at hospice.

- Hospice care is a type of health care that focuses on the palliation of a terminally ill patient's pain.
- Additionally, symptoms as well as their emotional and spiritual needs at the end of life are also addressed.
- Hospice care prioritizes comfort and quality of life by reducing pain and suffering.
- Hospice care provides an alternative to therapies focused on life-prolonging measures that may be arduous
- These are likely to cause more symptoms, and may not be aligned with a person's final life goals.
- The goal here is to see what is the best course of action to reduce the pain and suffering of people who are treated in the hospice in their last days.

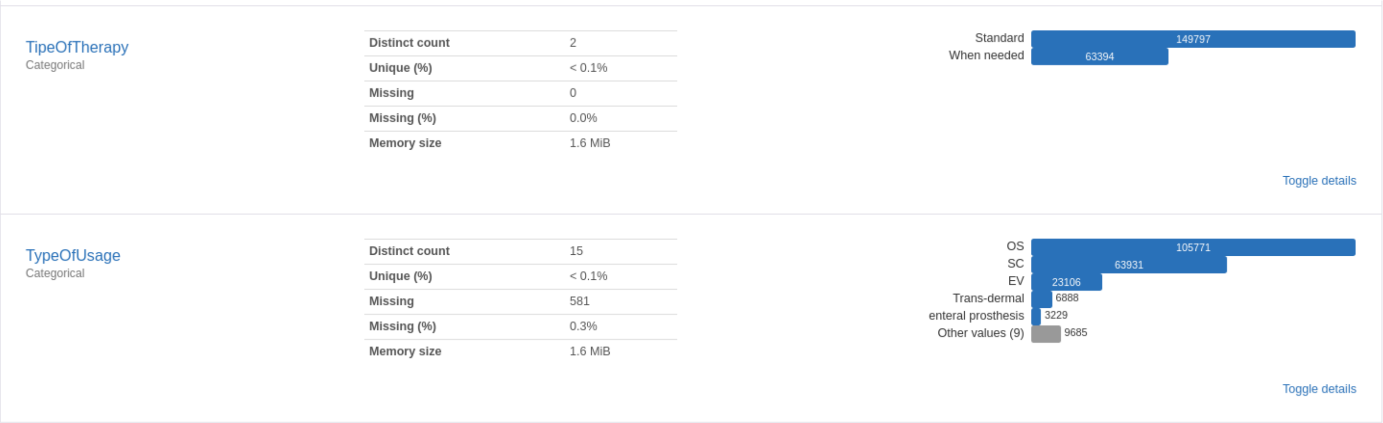
We were provided with 6 files which are as follows:

1. care_translated.xlsx
2. hospice_therapy_translated.xlsx
3. input_table_trarnslated.xlsx
4. obs_table_translated.xlsx
5. observation_table.xlsx
6. outcome_translated.xlsx

Analysis of the dataset

One of the best learnings was the introduction of the pandas profiler. The pandas profiler provides information in the form of missing values, unique values in a column.

It generates profile reports from a pandas DataFrame. The pandas df.describe() function is great but a little basic for serious exploratory data analysis. pandas_profiling extends the pandas DataFrame with df.profile_report() for quick data analysis.



TipeOfTherapy
Categorical

Distinct count	2
Unique (%)	< 0.1%
Missing	0
Missing (%)	0.0%
Memory size	1.6 MiB

TypeOfUsage
Categorical

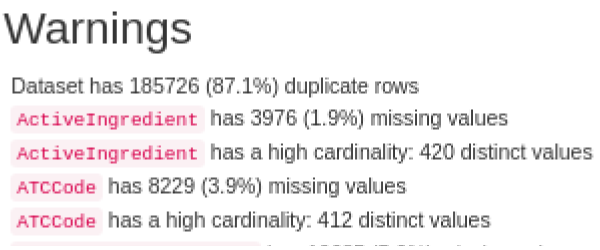
Distinct count	15
Unique (%)	< 0.1%
Missing	581
Missing (%)	0.3%
Memory size	1.6 MiB

I tried a few ideas to get the data consistent. One of the ideas was to do some NLP model that could read a column and make similar data points exactly same. But this idea was too complex, it was simpler to eliminate noise rather than attempt some sort of smoothing function.

Dataset info

Number of variables	22
Number of observations	213191
Missing cells	408591 (8.7%)
Duplicate rows	185726 (87.1%)
Total size in memory	178.7 MiB
Average record size in memory	879.0 B

At first glance, I also felt like new data frames could be created that could be useful for mining information present across the various files, but the challenge here was the scale of the data. Hence I took up only one file as explained in the next part.



Warnings

Dataset has 185726 (87.1%) duplicate rows

`ActiveIngredient` has 3976 (1.9%) missing values

`ActiveIngredient` has a high cardinality: 420 distinct values

`ATCCode` has 8229 (3.9%) missing values

`ATCCode` has a high cardinality: 412 distinct values

Vidas Hospice Data Exploration

We have five tables -

1. Care
2. Hospice Therapy
3. Input Table
4. Observation Table
5. Outcome Table

I worked on analysis of the hospice therapy sheet. Here are some of the graphs that obtained.

In [0]:
```
df = pd.read_csv('data/hospice_therapy_translated.csv', index_col=0)
df.shape
```

```
/home/aghadge/anaconda3/envs/appliedai/lib/python3.7/site-package
s/IPython/core/interactiveshell.py:3057: DtypeWarning: Columns (1) ha
ve mixed types. Specify dtype option on import or set low_memory=Fals
e.
  interactivity=interactivity, compiler=compiler, result=result)
```

Out[0]:
```
(213191, 21)
```

In [0]:

```
df_value_counts(df, top=10)
```

Out[0]:

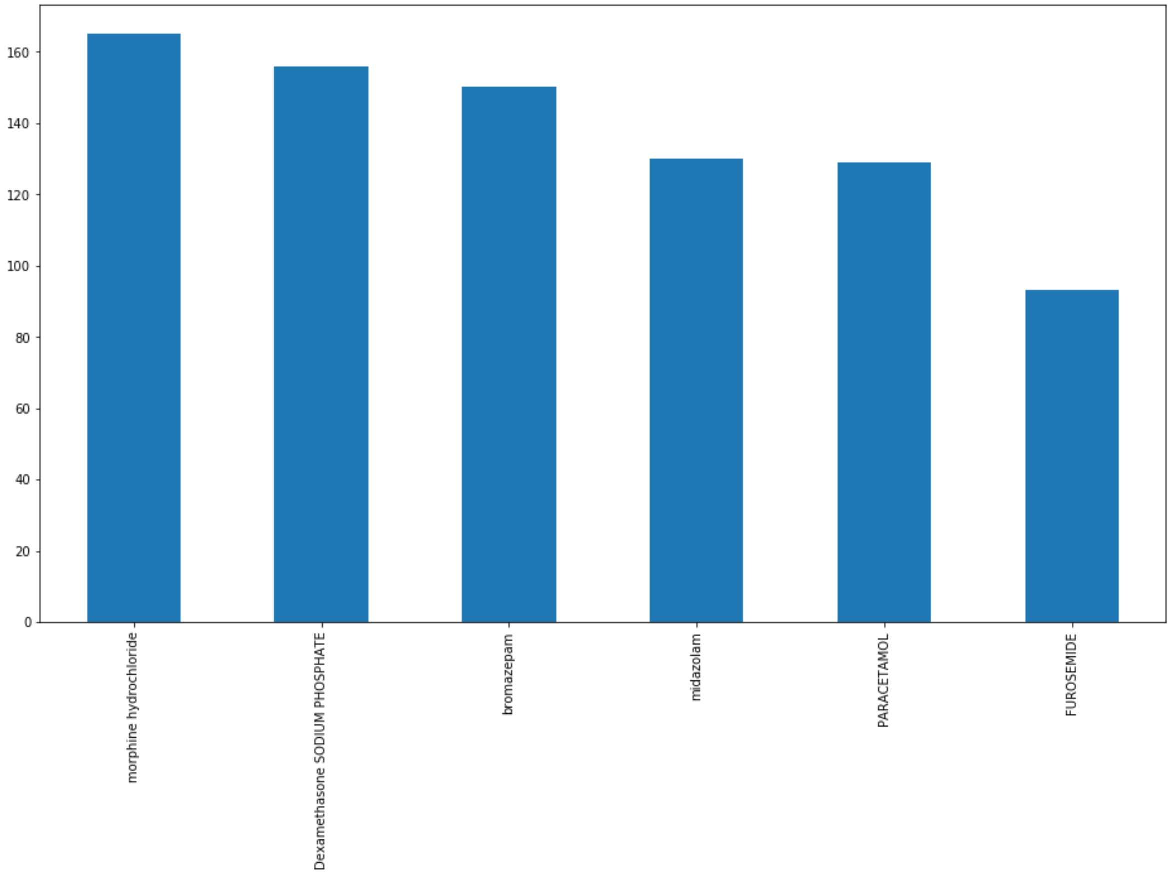
	IDBac	EHRID	Age	Sex	StartOfHospitalization	EndOfHospitalization	Diagnosis	T
0	0 - np.NaN	0 - np.NaN	1 - np.NaN	0 - np.NaN	550 - np.NaN	4085 - np.NaN	2804 - np.NaN	
1	1769 - Unique	1860 - Unique	82 - Unique	3 - Unique	831 - Unique	772 - Unique	25 - Unique	
2	1986 - 2490	1854 - 2146	10178 - 78.0	106999 - M	1940 - 2016-11-29	4388 - 2017-04-03	38527 - Lung	
3	1868 - 2609	1630 - 2158	9193 - 84.0	106191 - F	1630 - 2016-12-06	1981 - 2017-03-13	23513 - Colorectal	
4	1854 - 2893	1411 - 2131	8134 - 82.0	1 - OR	1613 - 2016-12-13	1949 - 2017-09-13	16295 - Pancreas	
5	1664 - 2228	1384 - 2499	8001 - 88.0	NaN	1464 - 2017-10-23	1538 - 2018-01-17	16023 - Breast	
6	1411 - 2827	1212 - 2664	7984 - 83.0	NaN	1384 - 2017-07-10	1537 - 2017-10-11	15414 - Diagnosis is not cancer	
7	1384 - 3976	1167 - 2495	7816 - 89.0	NaN	1368 - 2015-04-15	1523 - 2017-09-01	15151 - Brain-snc	
8	1237 - 1529	1142 - 1079	7755 - 85.0	NaN	1365 - 2015-12-10	1511 - 2015-05-07	12450 - Liver-biliary	
9	1212 - 3144	1131 - 2686	7197 - 87.0	NaN	1304 - 2016-10-04	1464 - 2015-09-19	9674 - Head-neck	
10	1167 - 3944	1130 - 2380	6581 - 81.0	NaN	1243 - 2017-07-13	1370 - 2016-09-12	9253 - Stomach	
11	1143 - 66	1113 - 2058	6446 - 73.0	NaN	1185 - 2015-08-19	1366 - 2017-07-05	8166 - Prostate	

Domain knowledge

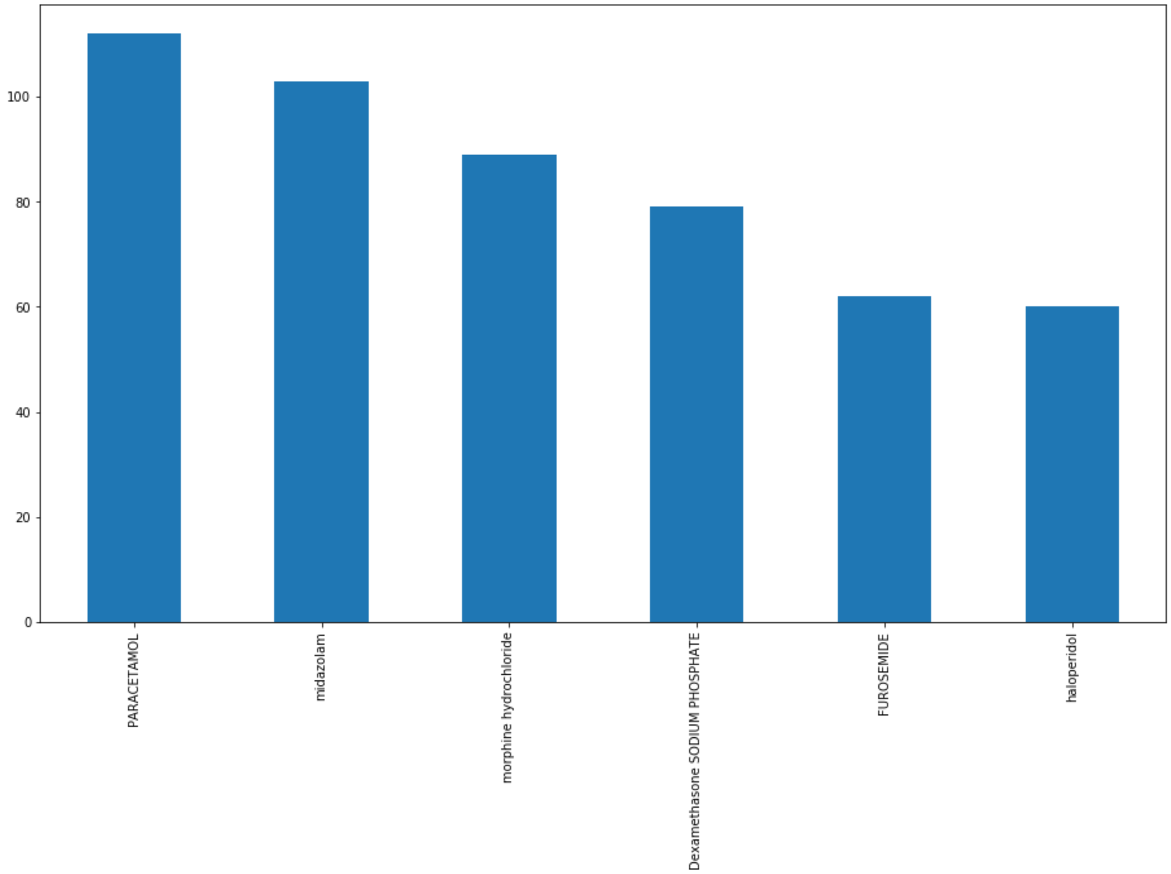
This data frame provided a solid base for exploration of various cancers. I went ahead with doing some frequency based analysis. But through out the talks, I realized that domain knowledge is often more valuable, especially while mining big datasets. The domain knowledge allows one to narrow down on what is meanigful in the data.

Initially it was felt that this was useful analysis. But it only confirmed the commonly known facts in the medical world.

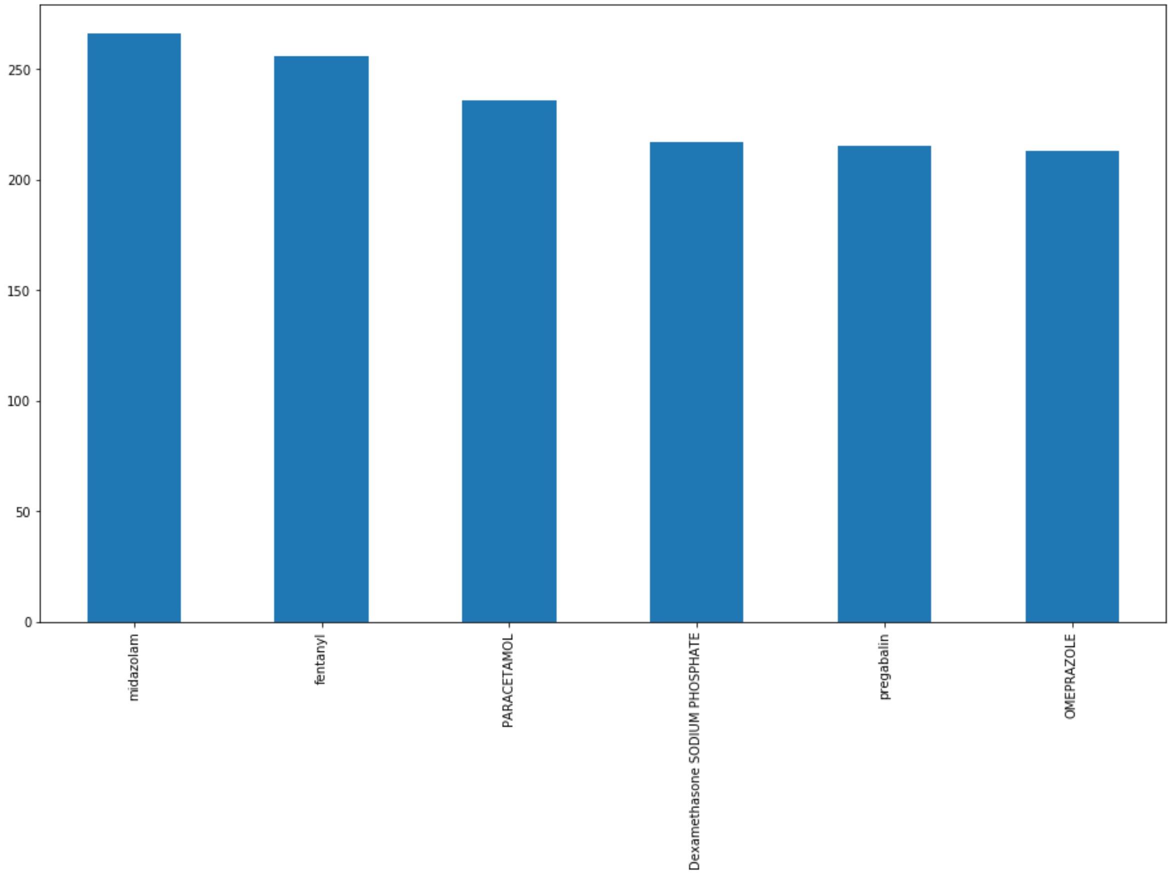
Diagnosis - Bone-soft tissue



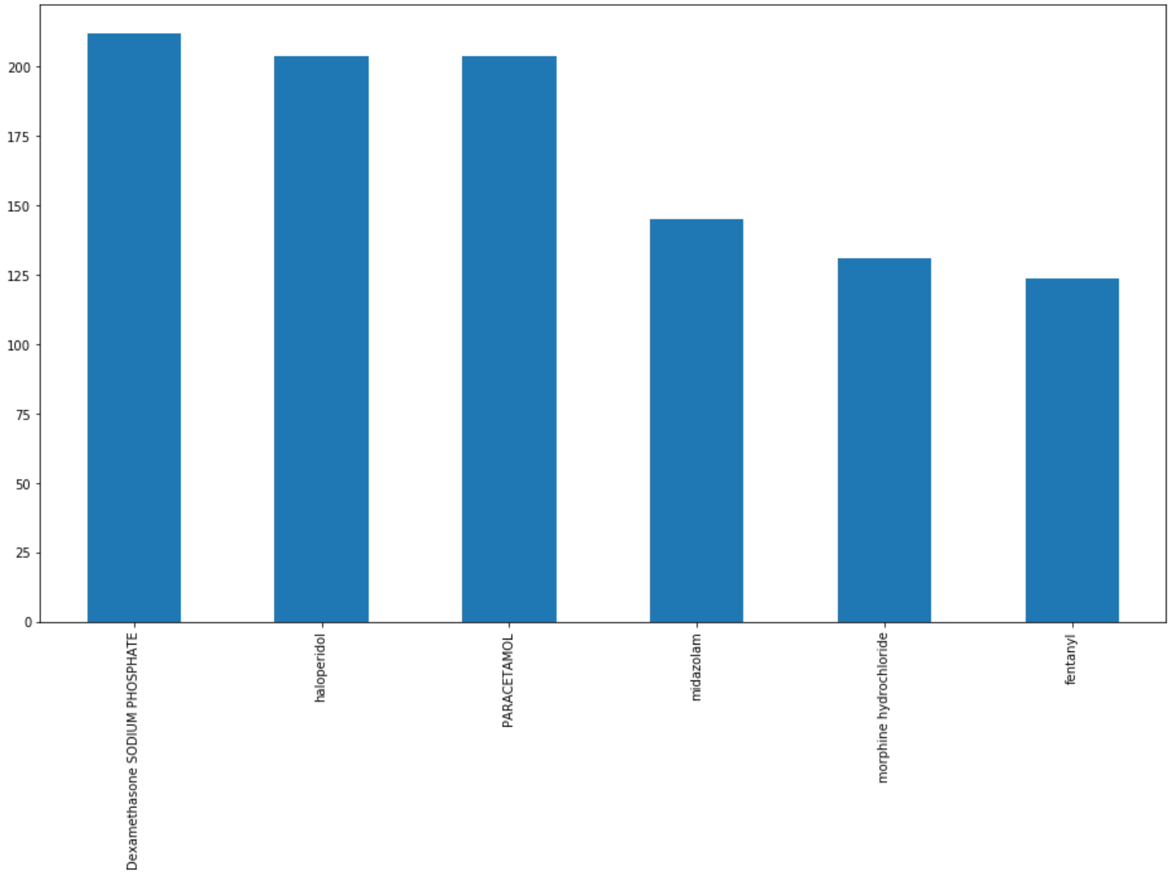
Diagnosis - Myelomas-other myeloproliferative diseases



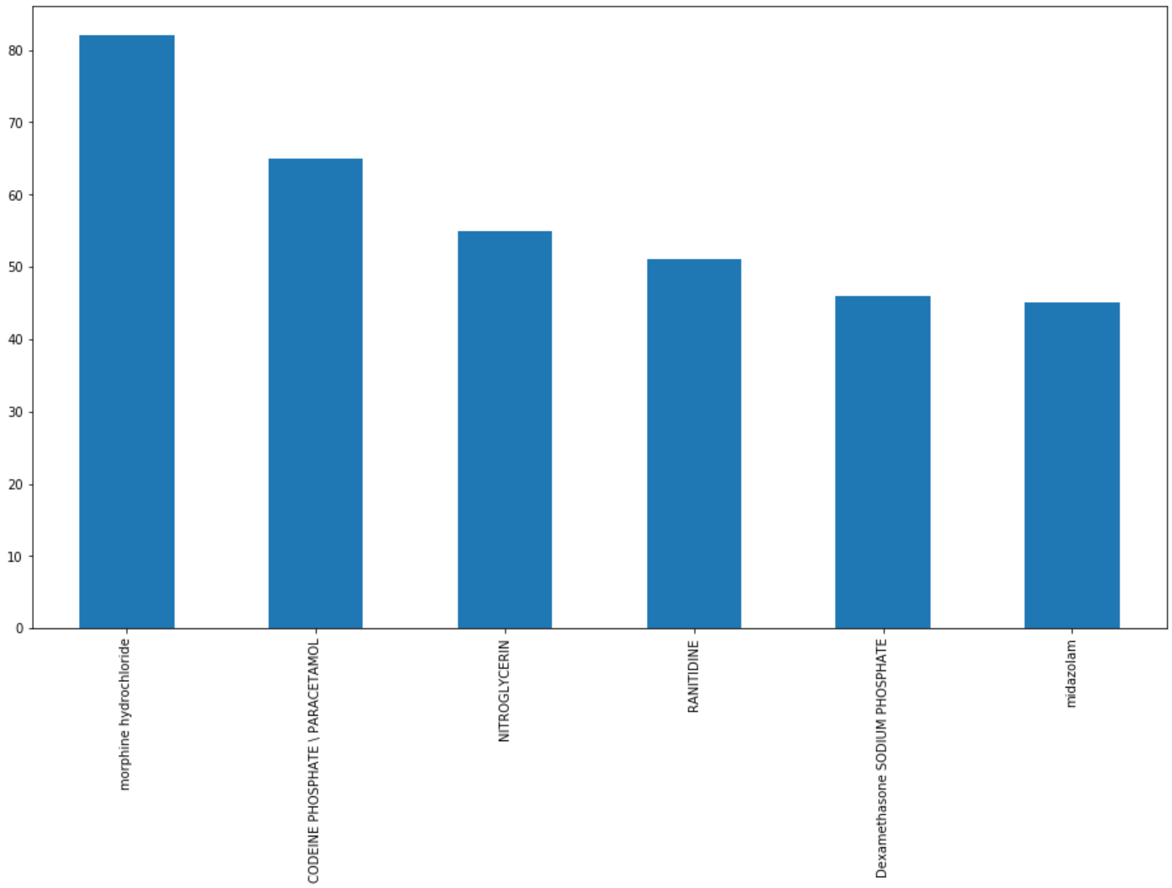
Diagnosis - Other



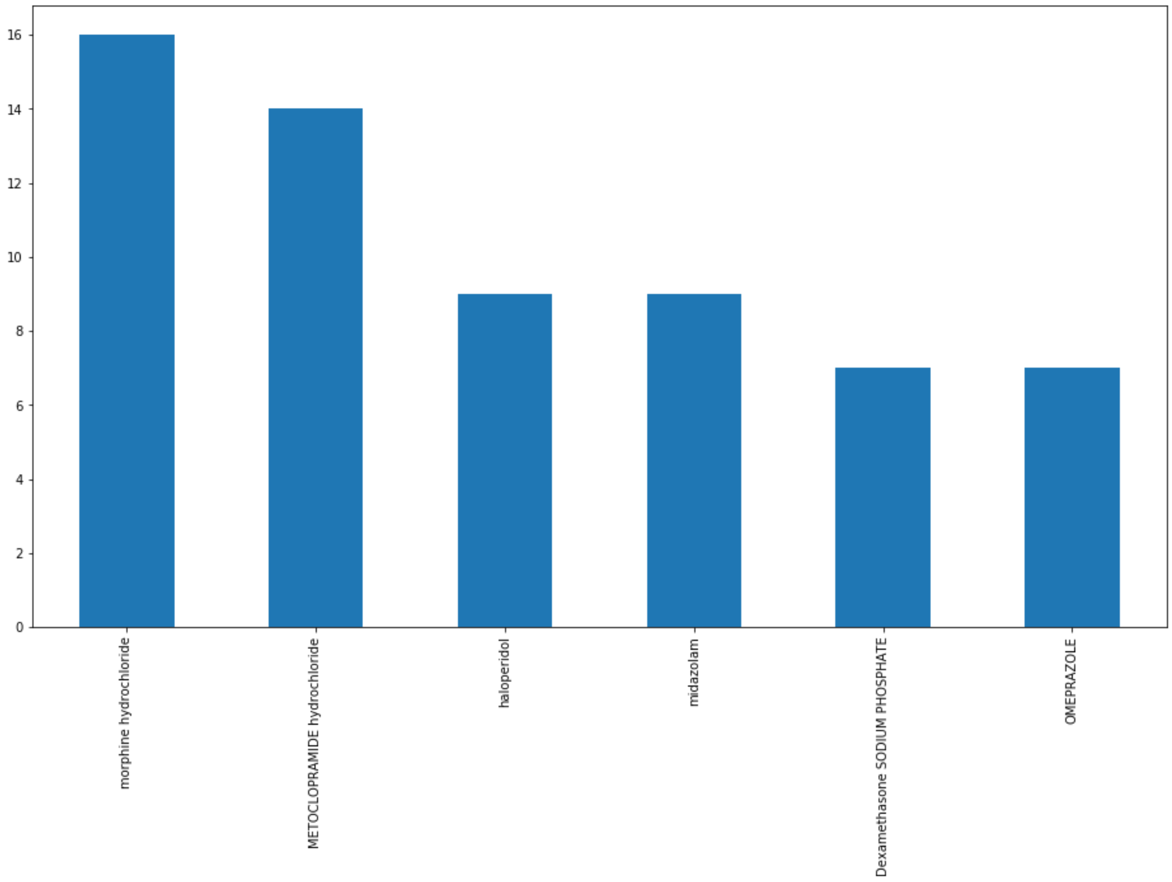
Diagnosis - not known



Diagnosis - endocrine glands



Diagnosis - psychological



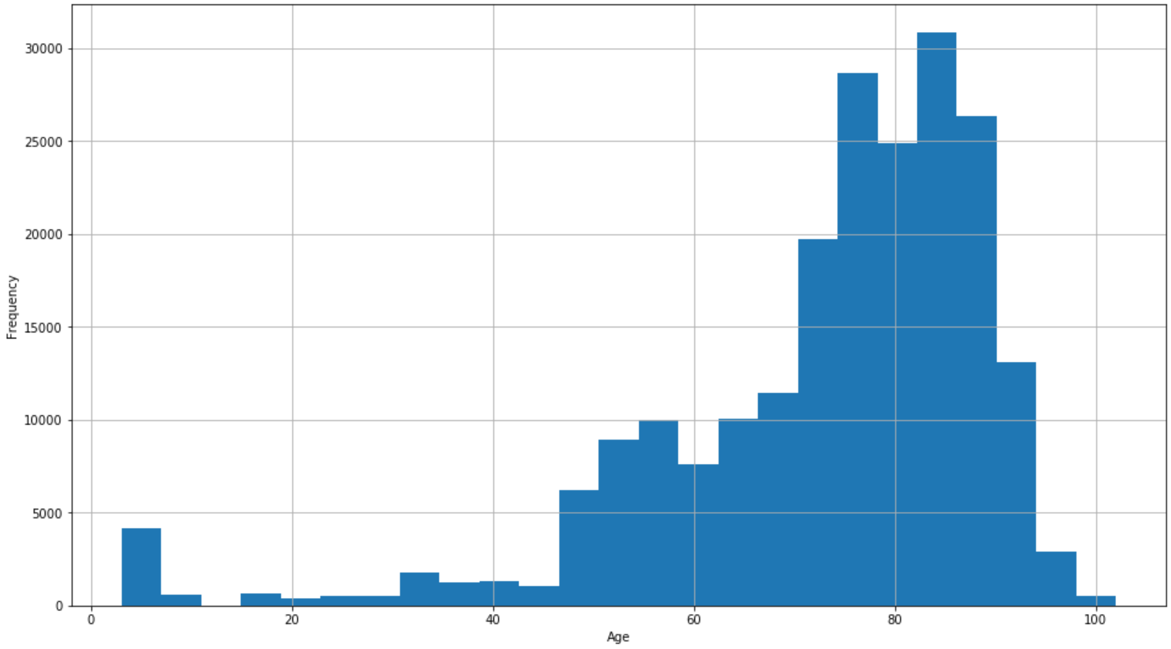
Some more frequency based information

In [0]:

```
df.Age.hist(bins=25, figsize=(16,9))
plt.xlabel('Age')
plt.ylabel('Frequency')
```

Out[0]:

```
Text(0, 0.5, 'Frequency')
```



Conclusion

The analysis paved way for further mining for each type of cancer. The main learning from this analysis was that one can often find meanigful information from a dataset. But meanigful information might not always correspond to relevant or useful information. In a very high-dimmensional and noisy dataset, as is found in the real world, it is important to have domain expertise. Interaction with the domain experts, was really helpful in taking the analysis forward.

One way of looking at requirement for domain experts was to understand that, correlations need to understood and many correlations need to be eliminated to improve the generalization of the model.

When we have a very high dimmensional dataset, domain experts, allow us to quickly identify correlated features. This is something that I think is at the heart of machine learning in real world.