Московский государственный технический университет им. Н.Э. Баумана
Кафедра «Системы обработки информации и управления»

Лабораторная работа №3
по дисциплине
«Методы машинного обучения»
на тему
«Обработка пропусков в данных, кодирование категориальных признаков, масштабирование данных»

Выполнил:
студент группы ИУ5-24М
Мельников К.

Москва — 2019 г.

In [38]:
```
import numpy as np
import pandas as pd
import seaborn as sns
from sklearn.preprocessing import StandardScaler
from sklearn.preprocessing import MinMaxScaler
```

In [14]:
```
data = pd.read_csv('survey.csv')
```

In [15]:
```
data.head()
```

Out[15]:

	Timestamp	Age	Gender	Country	state	self_employed
0	2014-08-27 11:29:31	37	Female	United States	IL	NaN
1	2014-08-27 11:29:37	44	M	United States	IN	NaN
2	2014-08-27 11:29:44	32	Male	Canada	NaN	NaN
3	2014-08-27 11:29:46	31	Male	United Kingdom	NaN	NaN
4	2014-08-27 11:30:22	31	Male	United States	TX	NaN

	family_history	treatment	work_interfere	no_employees	…	\
0	No	Yes	Often	6-25	…	
1	No	No	Rarely	More than 1000	…	
2	No	No	Rarely	6-25	…	
3	Yes	Yes	Often	26-100	…	
4	No	No	Never	100-500	…	

	leave	mental_health_consequence	phys_health_consequence
0	Somewhat easy	No	No
1	Don't know	Maybe	No
2	Somewhat difficult	No	No
3	Somewhat difficult	Yes	Yes
4	Don't know	No	No

	coworkers	supervisor	mental_health_interview	phys_health_interview
0	Some of them	Yes	No	Maybe
1	No	No	No	No
2	Yes	Yes	Yes	Yes
3	Some of them	No	Maybe	Maybe
4	Some of them	Yes	Yes	Yes

	mental_vs_physical	obs_consequence	comments
0	Yes	No	NaN
1	Don't know	No	NaN
2	No	No	NaN
3	No	Yes	NaN
4	Don't know	No	NaN

[5 rows x 27 columns]

In [16]:
```
data = data.drop(data[~((data['Gender'] == 'Male') | (data['Gender'] ==
```

In [17]:
```
data.head()
```

Out[17]:

	Timestamp	Age	Gender	Country	state	self_employed
0	2014-08-27 11:29:31	37	Female	United States	IL	NaN
2	2014-08-27 11:29:44	32	Male	Canada	NaN	NaN
3	2014-08-27 11:29:46	31	Male	United Kingdom	NaN	NaN
4	2014-08-27 11:30:22	31	Male	United States	TX	NaN
5	2014-08-27 11:31:22	33	Male	United States	TN	NaN

	family_history	treatment	work_interfere	no_employees	…	\
0	No	Yes	Often	6-25	…	
2	No	No	Rarely	6-25	…	
3	Yes	Yes	Often	26-100	…	
4	No	No	Never	100-500	…	
5	Yes	No	Sometimes	6-25	…	

	leave	mental_health_consequence	phys_health_consequence
0	Somewhat easy	No	No
2	Somewhat difficult	No	No
3	Somewhat difficult	Yes	Yes
4	Don't know	No	No
5	Don't know	No	No

	coworkers	supervisor	mental_health_interview	phys_health_interview
0	Some of them	Yes	No	Maybe
2	Yes	Yes	Yes	Yes
3	Some of them	No	Maybe	Maybe
4	Some of them	Yes	Yes	Yes
5	Yes	Yes	No	Maybe

	mental_vs_physical	obs_consequence	comments
0	Yes	No	NaN
2	No	No	NaN
3	No	Yes	NaN
4	Don't know	No	NaN
5	Don't know	No	NaN

[5 rows x 27 columns]

In [18]:
```
data['Gender'] = data['Gender'].astype('category')
```

In [19]:
```
data.dtypes
```

Out[19]:
```
Timestamp                    object
Age                           int64
Gender                     category
Country                      object
state                        object
self_employed                object
family_history               object
treatment                    object
work_interfere               object
no_employees                 object
```

```
remote_work                  object
tech_company                 object
benefits                     object
care_options                 object
wellness_program             object
seek_help                    object
anonymity                    object
leave                        object
mental_health_consequence    object
phys_health_consequence      object
coworkers                    object
supervisor                   object
mental_health_interview      object
phys_health_interview        object
mental_vs_physical           object
obs_consequence              object
comments                     object
dtype: object
```

In [26]:
```
total = data.shape[0]
```

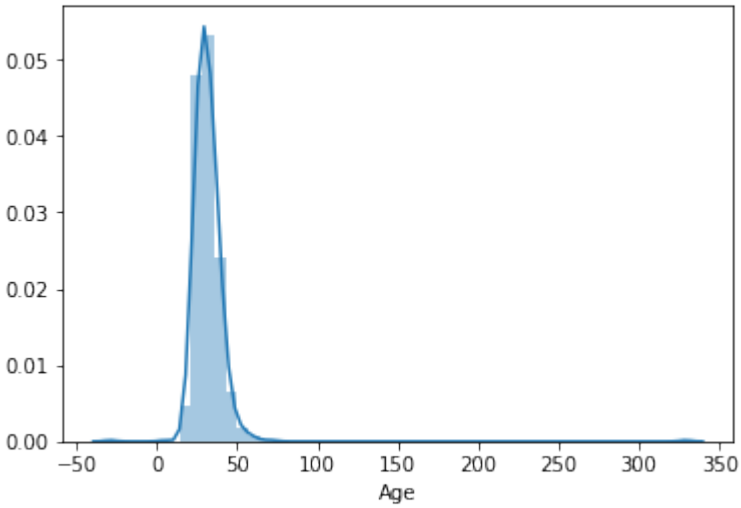
In [27]:
```
cat_cols = []
for col in data.columns:
    # Количество пустых значений

    null_count = data[data[col].isnull()].shape[0]
    dt = str(data[col].dtype)
    if null_count>0 and (dt=='object'):
        cat_cols.append(col)
        print(col + ' ' + str(null_count/total))
```

```
state 0.42527173913043476
self_employed 0.017663043478260868
work_interfere 0.20244565217391305
comments 0.8790760869565217
```

In [29]:
```
sns.distplot(data['Age'])
```

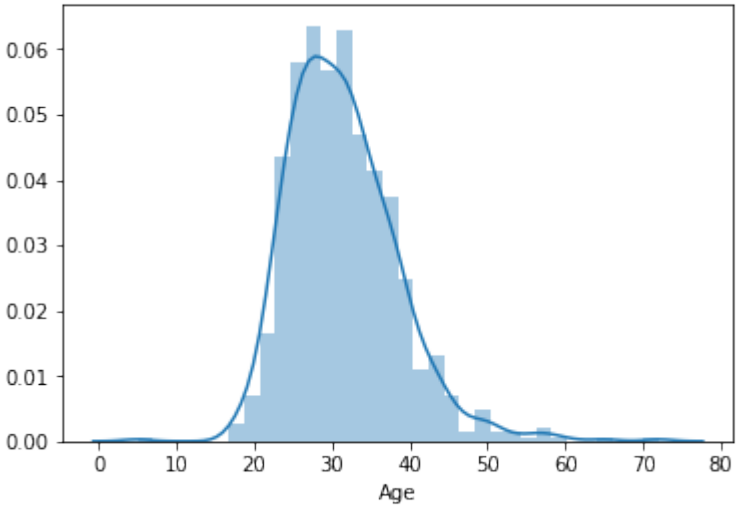
Out[29]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f34754c9e10>
```




```
In [32]: data = data.drop(data[(data['Age']>100) | (data['Age'] < 0)].index)
```

```
In [33]: sns.distplot(data['Age'])
```

```
Out[33]: <matplotlib.axes._subplots.AxesSubplot at 0x7f3473394ef0>
```




```
In [37]: stand = StandardScaler()
```

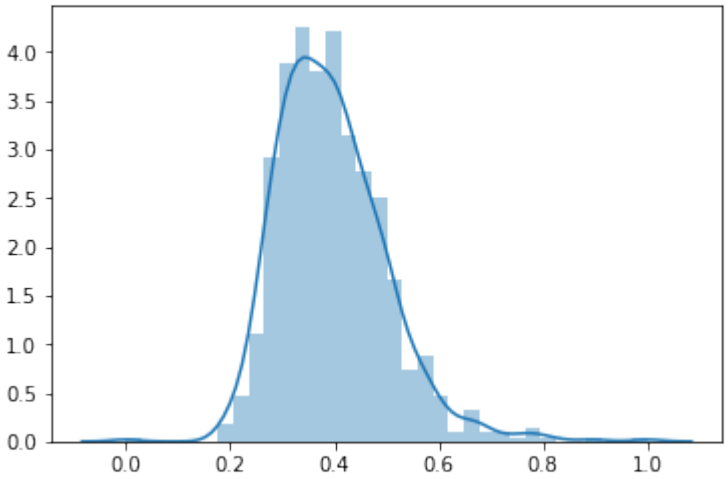
```
In [39]: minmax = MinMaxScaler()
```

```
In [ ]: stand.
```

```
In [47]: sns.distplot(minmax.fit_transform(data[['Age']]))
```

```
/home/hexagramg/tmo/venv/lib/python3.6/site-packages/sklearn/preprocessing/data.py
  return self.partial_fit(X, y)
```

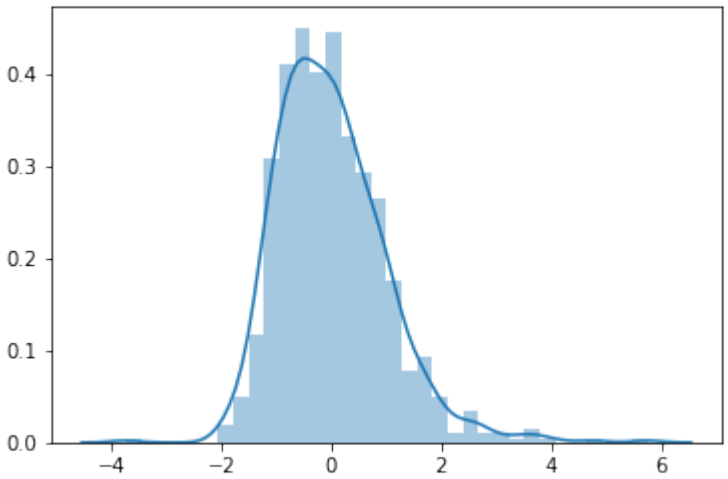
Out[47]: <matplotlib.axes._subplots.AxesSubplot at 0x7f3470aa04e0>



In [49]: sns.distplot(stand.fit_transform(data[['Age']]))

```
/home/hexagramg/tmo/venv/lib/python3.6/site-packages/sklearn/preprocessing/data.py
  return self.partial_fit(X, y)
/home/hexagramg/tmo/venv/lib/python3.6/site-packages/sklearn/base.py:464: DataConv
  return self.fit(X, **fit_params).transform(X)
```

Out[49]: <matplotlib.axes._subplots.AxesSubplot at 0x7f3470a4e748>



In [ ]: