marriage-age

September 30, 2023

```
[1]: import numpy as np
     import pandas as pd
```

```
[2]: data = pd.read_csv("marriage.csv")
     data.head()
```

```
[2]:    id  gender height   religion        caste mother_tongue  \
     0   1  female   5'4"        NaN       others        Telugu
     1   2    male   5'7"       Jain   Shwetamber      Gujarati
     2   3    male   5'7"      Hindu      Brahmin         Hindi
     3   4  female   5'0"      Hindu       Thakur         Hindi
     4   5    male   5'5"  Christian  Born Again     Malayalam

                              profession         location         country  \
     0                               NaN           London  United Kingdom
     1  Doctor / Healthcare Professional      Fairfax- VA             USA
     2         Entrepreneurs / Business         Begusarai           India
     3                         Architect           Mumbai           India
     4    Sales Professional / Marketing  Sulthan Bathery           India

        age_of_marriage
     0             21.0
     1             32.0
     2             32.0
     3             30.0
     4             30.0
```

```
[3]: data.isnull().sum()
```

```
[3]: id                   0
     gender              29
     height             118
     religion           635
     caste              142
     mother_tongue      164
     profession         330
     location           155
```

```
country             16
age_of_marriage     19
dtype: int64
```

[5]:
```
data.shape
```

[5]:
```
(2567, 10)
```

[7]:
```
data.dropna(inplace=True)
```

[8]:
```
data.shape
```

[8]:
```
(1932, 10)
```

[10]:
```
x = data.loc[:, ['gender', 'religion', 'caste', 'mother_tongue', 'country',␣
 ↪'height']]
y=data.age_of_marriage
```

[11]:
```
x.head()
```

[11]:
```
   gender   religion        caste mother_tongue country height
1    male       Jain   Shwetamber      Gujarati     USA   5'7"
2    male      Hindu      Brahmin         Hindi   India   5'7"
3  female      Hindu       Thakur         Hindi   India   5'0"
4    male  Christian   Born Again     Malayalam   India   5'5"
5    male      Hindu      Valmiki         Hindi   India   5'5"
```

[12]:
```
from sklearn.preprocessing import LabelEncoder
enc = LabelEncoder()
```

[14]:
```
x.loc[:,['gender', 'religion', 'caste', 'mother_tongue', 'country']] = \
x.loc[:, ['gender', 'religion', 'caste', 'mother_tongue', 'country']].apply(enc.
 ↪fit_transform)
x.head()
```

[14]:
```
  gender religion caste mother_tongue country height
1      1        2    34             6      19   5'7"
2      1        1    14             8       5   5'7"
3      0        1    36             8       5   5'0"
4      1        0    13            13       5   5'5"
5      1        1    38             8       5   5'5"
```

[15]:
```
def h_cms(h):
    return int(h[0])*30.48 + int(h[2])*2.54

x.height = x.height.apply(h_cms)
```

[16]:

```
x.head()
```

[16]:

```
  gender religion caste mother_tongue country  height
1      1        2    34             6      19  170.18
2      1        1    14             8       5  170.18
3      0        1    36             8       5  152.40
4      1        0    13            13       5  165.10
5      1        1    38             8       5  165.10
```

[17]:

```
from sklearn.model_selection import train_test_split

x_train, x_test, y_train, y_test = train_test_split(x,y, test_size=0.2,␣
 ↪random_state=42)
```

[37]:

```
from sklearn.ensemble import RandomForestRegressor
from sklearn.tree import DecisionTreeRegressor
from sklearn.svm import SVR
```

[39]:

```
rf= RandomForestRegressor(n_estimators=80)
rf.fit(x_train, y_train)
y_pred = rf.predict(x_test)
```

Evaluation

[40]:

```
from sklearn.metrics import mean_absolute_error, r2_score
print(mean_absolute_error(y_test, y_pred))
print(r2_score(y_test, y_pred))
```

```
1.090895819328617
0.6786114852293732
```

[41]:

```
dt = DecisionTreeRegressor()
dt.fit(x_train, y_train)
y_pred = dt.predict(x_test)
```

[42]:

```
print(mean_absolute_error(y_test, y_pred))
print(r2_score(y_test, y_pred))
```

```
1.1781797301177146
0.5965608709473644
```

[43]:

```
# create SVR model
svr = SVR()
svr.fit(x_train, y_train)
y_pred = svr.predict(x_test)

print(mean_absolute_error(y_test, y_pred))
print(r2_score(y_test, y_pred))
```

```
1.8474070323401295
0.041590953887349570
```

[44]:

```
# create ensemble model
from sklearn.ensemble import VotingRegressor

vr = VotingRegressor([('rf', rf), ('dt', dt), ('svr', svr)])
vr.fit(x_train, y_train)
y_pred = vr.predict(x_test)

print(mean_absolute_error(y_test, y_pred))
print(r2_score(y_test, y_pred))
```

```
1.170410447882755
0.6200396100676283
```

[48]:

```
import joblib
joblib.dump(vr, 'marriage_age_predictor.ml')
```

[48]:

```
['marriage_age_predictor.ml']
```

[ ]:

```

```