IT17157674 – V.Aperame

4th Year 1st Semester, Cyber Security (CS) Batch

MLCS (Machine Learning for Cyber Security)

Assignment - Marriage Year Prediction

This project is mainly based for Indians and those (Indians) who are living outside India, like foreign countries including United States, United Kingdom, Oman etc.

This is the end to end machine learning project with the help of the Flask Deployment. The "Marriage Year Prediction" project consists of the following parts. Such as,

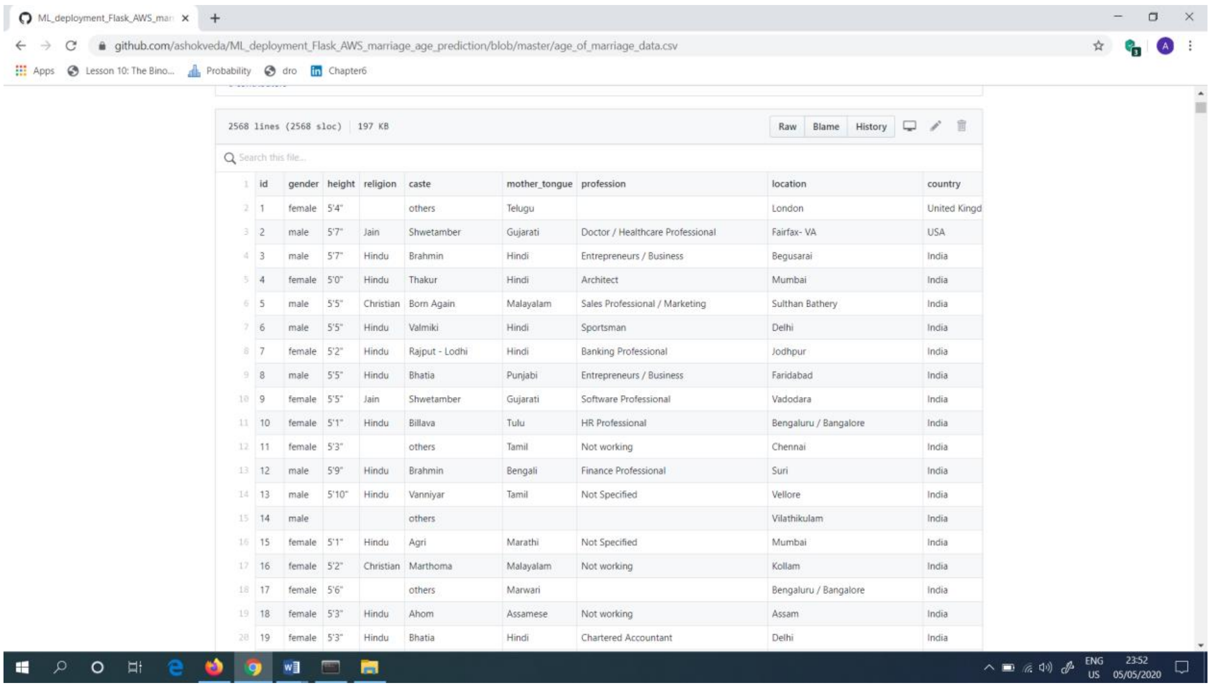
1. Preparation of Data
2. Prepare the Machine Learning Model
3. Evaluate the model
4. Export the trained model
5. Prepare LOCAL REST API with the help of Flask web-server
6. Finally, create a website for to predict the year of marriage

So, this project is mainly based on, by collecting the data sets of the people who recently got married, i.e.: from 1989 to 1992 (3 Years) from some parts of the world. And I created a predictive model which is based on certain factors like religion, caste (Because in India it is considered), location, education etc. So, by considering all these above mentioned factors, I felt very much interested in predicting the year of marriage. That's why I selected this Machine Learning Project.

The algorithm which I used was "Random Forest Algorithm", to do this Marriage Year Prediction project.

I gathered the data sets, which was needed in order to do this project was from the following GitHub link. This is as follows.
https://github.com/ashokveda/ML_deployment_Flask_AWS_marriage_age_prediction/blob/master/age_of_marriage_data.csv

There are about, 2567 rows and 10 columns were there in the gathered data set. This is shown in (Figure 1).

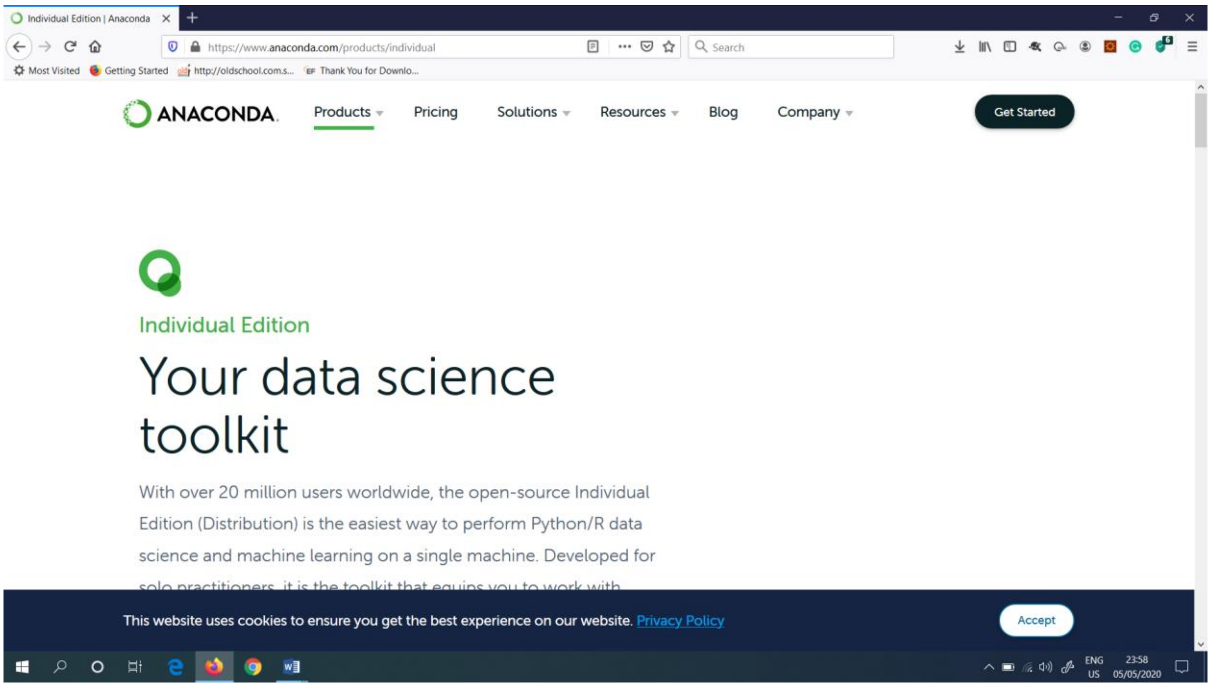

ML_deployment_Flask_AWS_marr
github.com/ashokveda/ML_deployment_Flask_AWS_marriage_age_prediction/blob/master/age_of_marriage_data.csv
Apps
Lesson 10: The Bino...
Probability
dro
Chapter6
Raw
Blame
History
Search this file...
ENG US
23:52 05/05/2020

2568 lines (2568 sloc) | 197 KB

id	gender	height	religion	caste	mother_tongue	profession	location	country
1	female	5'4"		others	Telugu		London	United Kingd
2	male	5'7"	Jain	Shwetamber	Gujarati	Doctor / Healthcare Professional	Fairfax- VA	USA
3	male	5'7"	Hindu	Brahmin	Hindi	Entrepreneurs / Business	Begusarai	India
4	female	5'0"	Hindu	Thakur	Hindi	Architect	Mumbai	India
5	male	5'5"	Christian	Born Again	Malayalam	Sales Professional / Marketing	Sulthan Bathery	India
6	male	5'5"	Hindu	Valmiki	Hindi	Sportsman	Delhi	India
7	female	5'2"	Hindu	Rajput - Lodhi	Hindi	Banking Professional	Jodhpur	India
8	male	5'5"	Hindu	Bhatia	Punjabi	Entrepreneurs / Business	Faridabad	India
9	female	5'5"	Jain	Shwetamber	Gujarati	Software Professional	Vadodara	India
10	female	5'1"	Hindu	Billava	Tulu	HR Professional	Bengaluru / Bangalore	India
11	female	5'3"		others	Tamil	Not working	Chennai	India
12	male	5'9"	Hindu	Brahmin	Bengali	Finance Professional	Suri	India
13	male	5'10"	Hindu	Vanniyar	Tamil	Not Specified	Vellore	India
14	male			others			Vilathikulam	India
15	female	5'1"	Hindu	Agri	Marathi	Not Specified	Mumbai	India
16	female	5'2"	Christian	Marthoma	Malayalam	Not working	Kollam	India
17	female	5'6"		others	Marwari		Bengaluru / Bangalore	India
18	female	5'3"	Hindu	Ahom	Assamese	Not working	Assam	India
19	female	5'3"	Hindu	Bhatia	Hindi	Chartered Accountant	Delhi	India

Figure 1: Number of rows and columns presented in the downloaded data set (.csv file)

Next, in order to do the Machine Learning Model, I installed the "Anaconda" for the PC (Personal Computer). The following (Figure 2) shows the website which I downloaded the "Anaconda3-2020.02-Windows-x86_64.exe" file, which is used to install the software named as "Anaconda".



Individual Edition

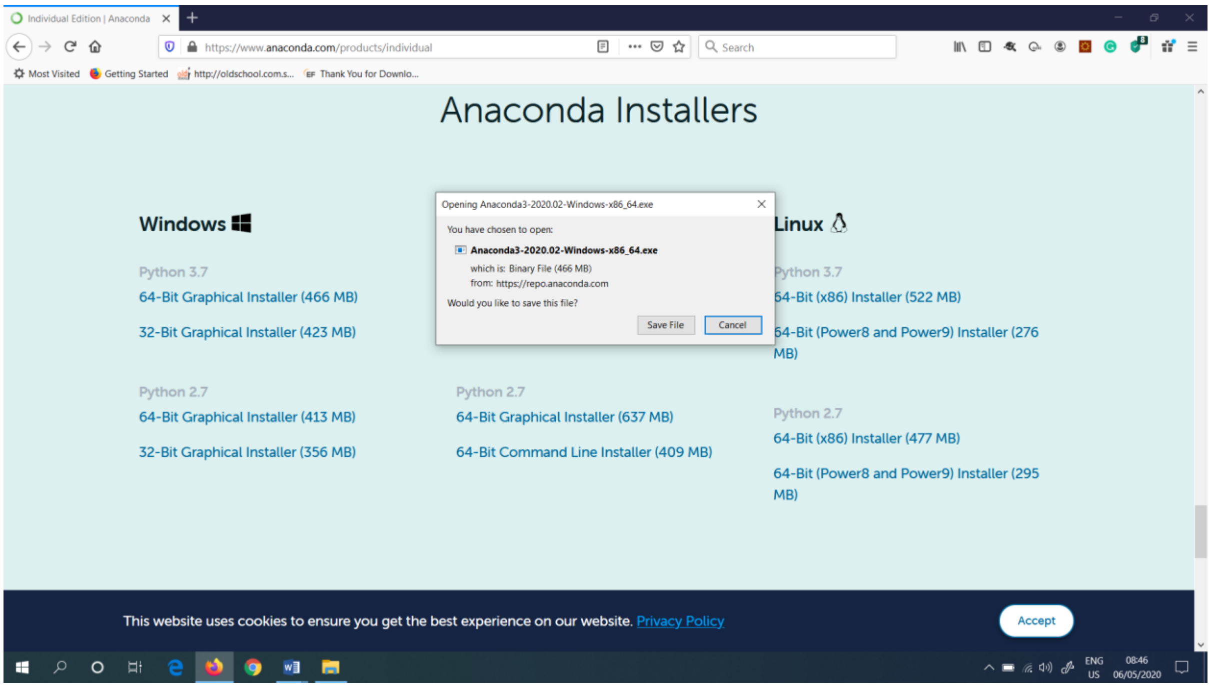
Your data science toolkit

With over 20 million users worldwide, the open-source Individual Edition (Distribution) is the easiest way to perform Python/R data science and machine learning on a single machine. Developed for solo practitioners, it is the toolkit that equips you to work with

This website uses cookies to ensure you get the best experience on our website. Privacy Policy

Figure 2: Website to download the Anaconda exe file

As my PC Operating System is Windows and as it is 64-bit PC, I selected the Windows and 64-bit option and downloaded the "Anaconda.exe file". This is shown in (Figure 3).


Anaconda Installers

Windows

Python 3.7
64-Bit Graphical Installer (466 MB)
32-Bit Graphical Installer (423 MB)

Python 2.7
64-Bit Graphical Installer (413 MB)
32-Bit Graphical Installer (356 MB)

Opening Anaconda3-2020.02-Windows-x86_64.exe
You have chosen to open:
Anaconda3-2020.02-Windows-x86_64.exe
which is: Binary File (466 MB)
from: https://repo.anaconda.com
Would you like to save this file?

Python 2.7
64-Bit Graphical Installer (637 MB)
64-Bit Command Line Installer (409 MB)

Linux

Python 3.7
64-Bit (x86) Installer (522 MB)
64-Bit (Power8 and Power9) Installer (276 MB)

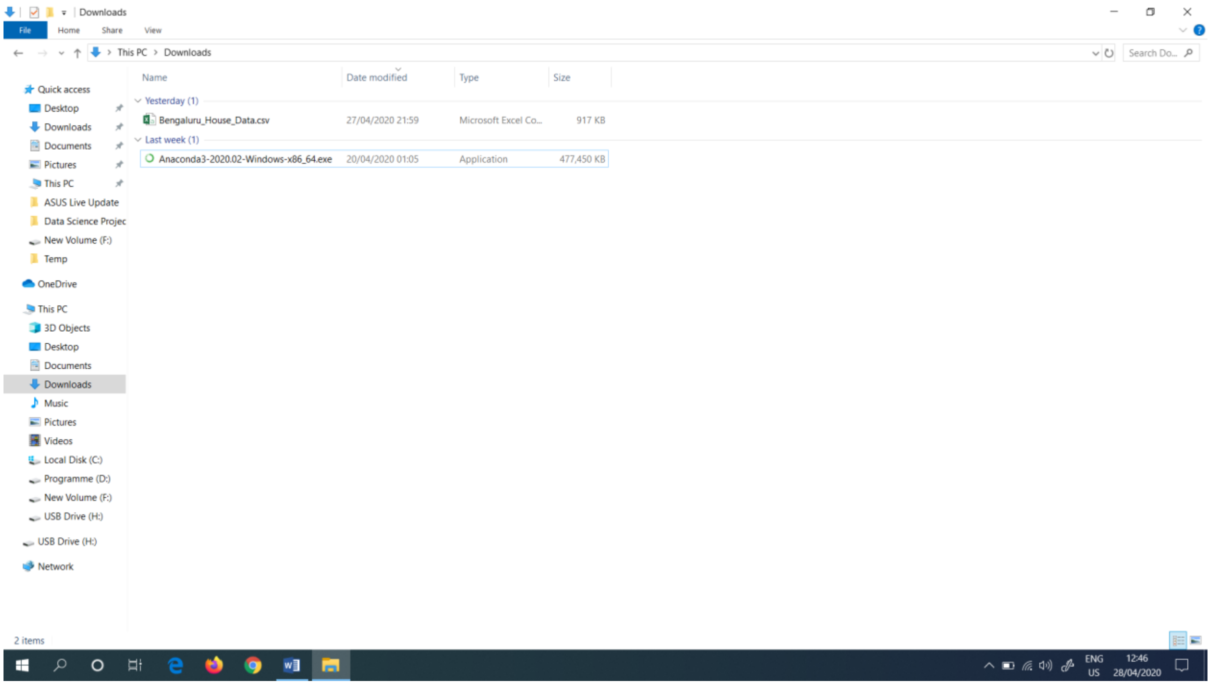
Python 2.7
64-Bit (x86) Installer (477 MB)
64-Bit (Power8 and Power9) Installer (295 MB)

This website uses cookies to ensure you get the best experience on our website. Privacy Policy

Individual Edition | Anaconda
https://www.anaconda.com/products/individual
Most Visited
Getting Started
http://oldschool.com.s...
Thank You for Downlo...
Save File
Cancel
Accept
ENG US
08:46
06/05/2020

Figure 3: Downloaded Anaconda exe file

After I downloaded the "Anaconda.exe" file, it is shown under the "Downloads" section in the PC, which is shown in (Figure 4).


This PC > Downloads

Name	Date modified	Type	Size
Yesterday (1)			
Bengaluru_House_Data.csv	27/04/2020 21:59	Microsoft Excel Co...	917 KB
Last week (1)			
Anaconda3-2020.02-Windows-x86_64.exe	20/04/2020 01:05	Application	477,450 KB

2 items

Figure 4: Downloaded Anaconda.exe file which is presented under the "Downloads"

Then I opened the "Anaconda Prompt" in the search bar (Figure 5).

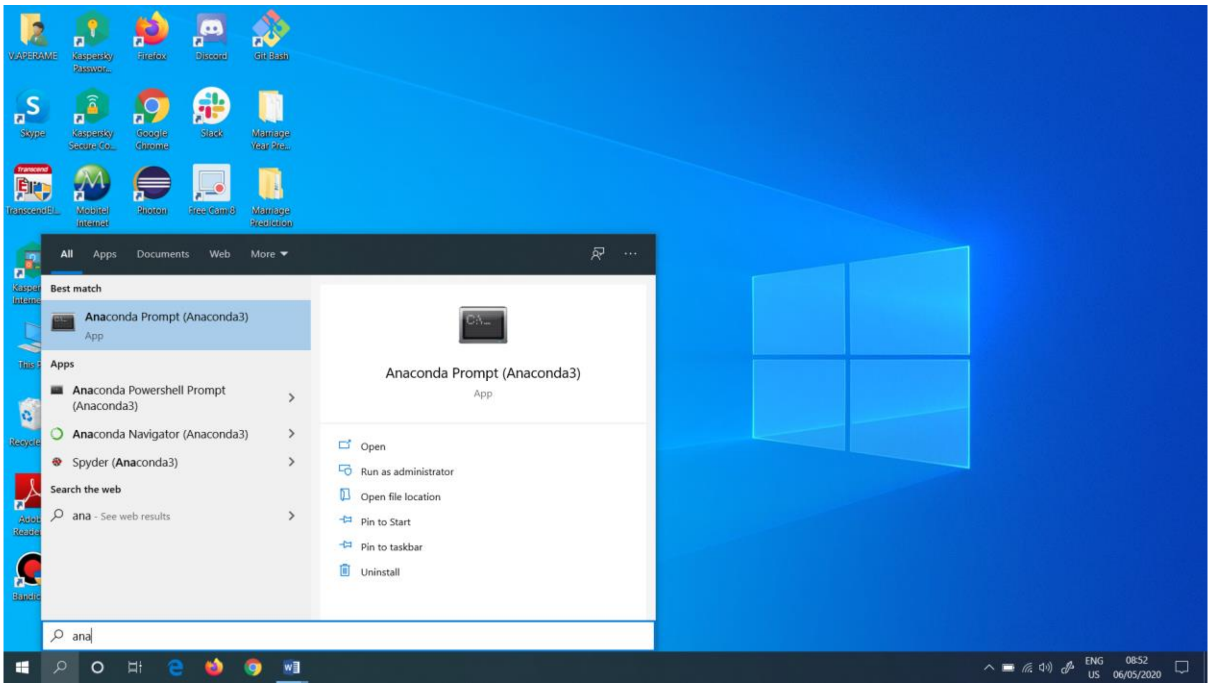
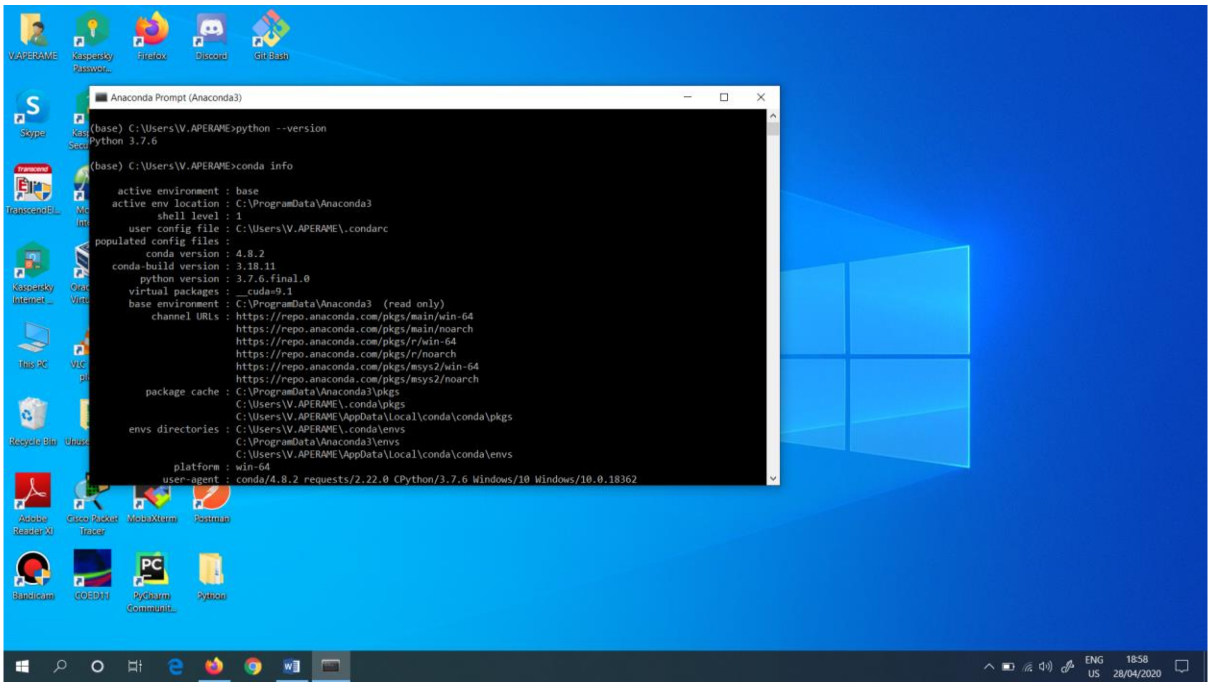


Figure 5: Opened the Anaconda Prompt

Next I checked the Python version, in the Anaconda prompt. So I typed as “python - - version”, I have also seen the version of conda. So I typed as “conda info” to get the version of conda. These are shown in (Figure 6).

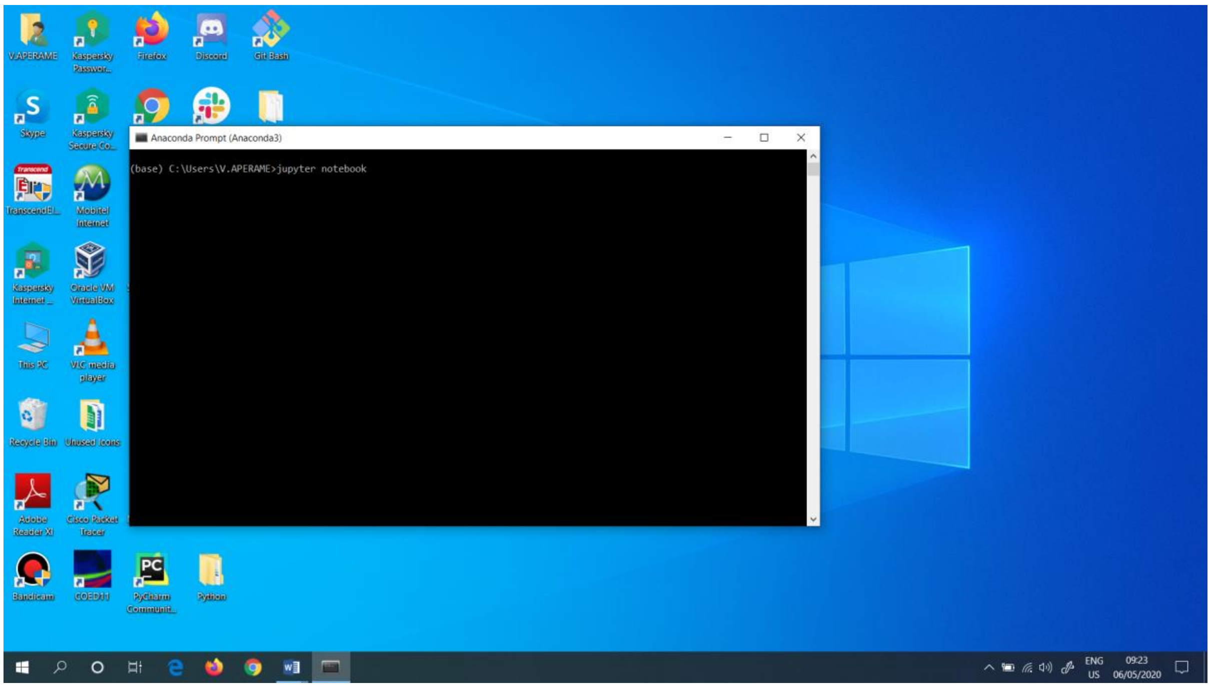


```
(base) C:\Users\V.APERAME>python --version
Python 3.7.6

(base) C:\Users\V.APERAME>conda info

     active environment : base
    active env location : C:\ProgramData\Anaconda3
            shell level : 1
       user config file : C:\Users\V.APERAME\.condarc
 populated config files :
          conda version : 4.8.2
    conda-build version : 3.18.11
         python version : 3.7.6.final.0
       virtual packages : __cuda=9.1
       base environment : C:\ProgramData\Anaconda3  (read only)
           channel URLs : https://repo.anaconda.com/pkgs/main/win-64
                          https://repo.anaconda.com/pkgs/main/noarch
                          https://repo.anaconda.com/pkgs/r/win-64
                          https://repo.anaconda.com/pkgs/r/noarch
                          https://repo.anaconda.com/pkgs/msys2/win-64
                          https://repo.anaconda.com/pkgs/msys2/noarch
          package cache : C:\ProgramData\Anaconda3\pkgs
                          C:\Users\V.APERAME\.conda\pkgs
                          C:\Users\V.APERAME\AppData\Local\conda\conda\pkgs
       envs directories : C:\Users\V.APERAME\.conda\envs
                          C:\ProgramData\Anaconda3\envs
                          C:\Users\V.APERAME\AppData\Local\conda\conda\envs
               platform : win-64
             user-agent : conda/4.8.2 requests/2.22.0 CPython/3.7.6 Windows/10 Windows/10.0.18362
```

Figure 6: Checked the Python and conda version in Anaconda prompt

“Conda” is an open source package management system and environment management system that runs on Windows, macOS and Linux. Conda quickly installs, runs and updates packages and their dependencies.

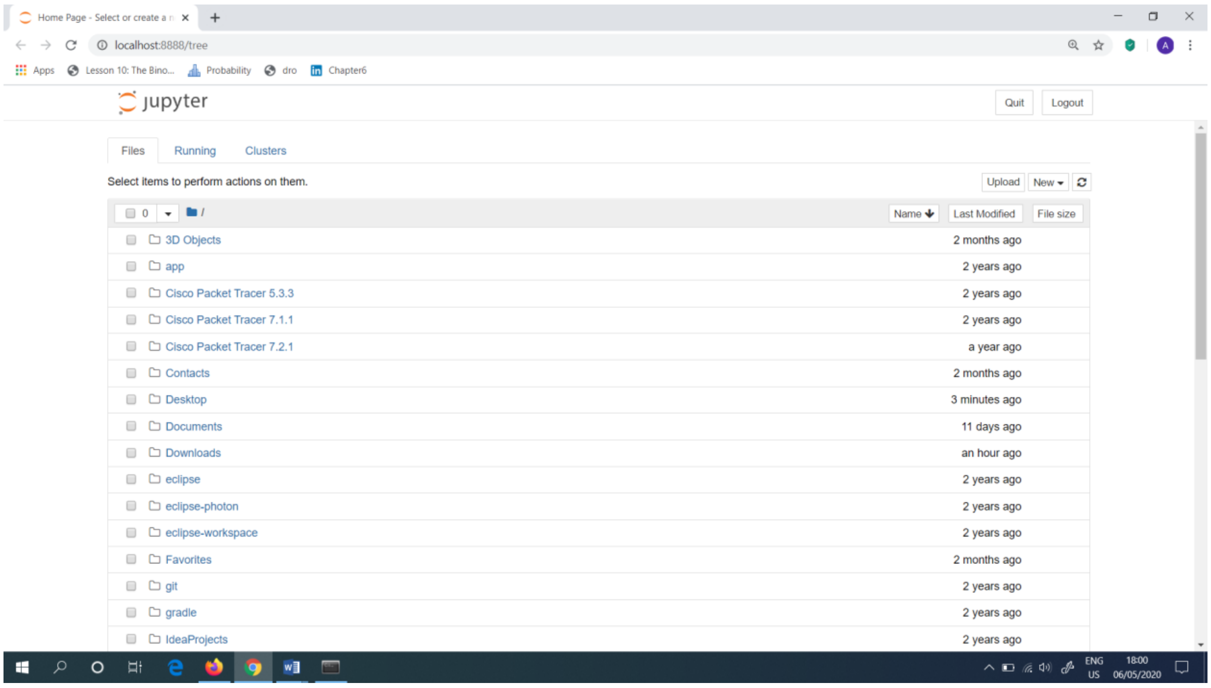
Then I typed as “Jupyter Notebook” in the Anaconda Prompt (Figure 7) and opened the Jupyter Notebook in the local host in any browser.



```
(base) C:\Users\V.APERAME>jupyter notebook
```

Figure 7: Opened the Jupyter Notebook

So, I opened the Jupyter Notebook in Google Chrome. This is shown in (Figure 8).



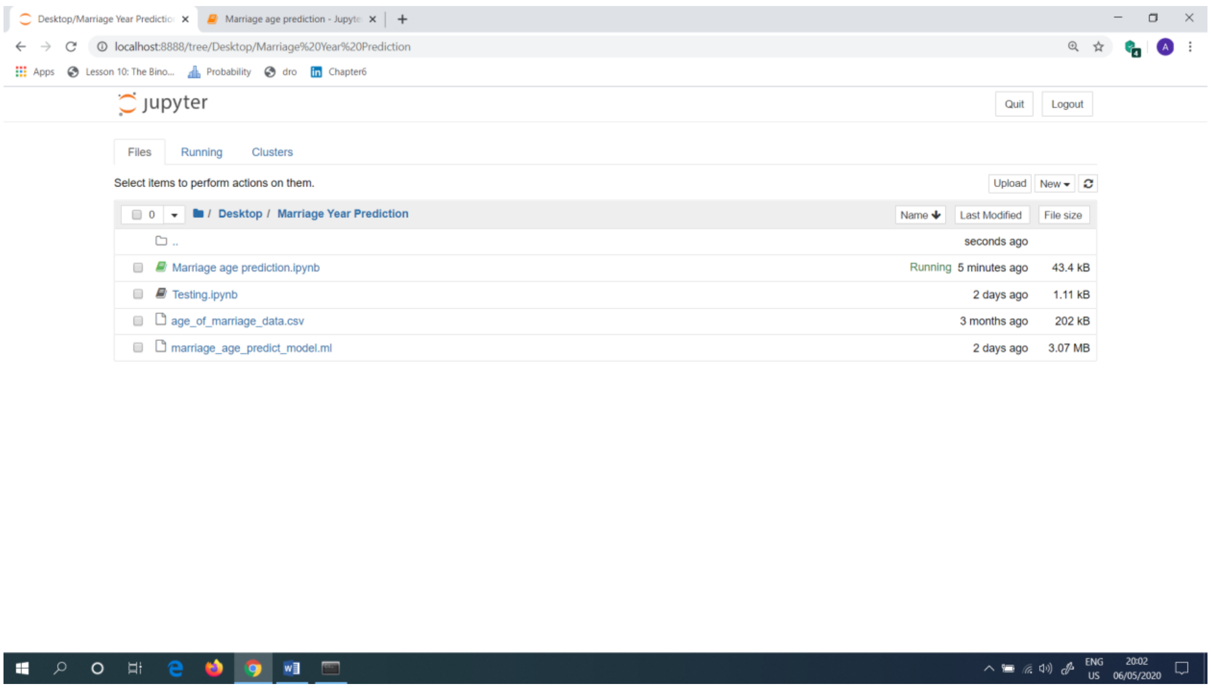
localhost:8888/tree

Select items to perform actions on them.

Name	Last Modified
3D Objects	2 months ago
app	2 years ago
Cisco Packet Tracer 5.3.3	2 years ago
Cisco Packet Tracer 7.1.1	2 years ago
Cisco Packet Tracer 7.2.1	a year ago
Contacts	2 months ago
Desktop	3 minutes ago
Documents	11 days ago
Downloads	an hour ago
eclipse	2 years ago
eclipse-photon	2 years ago
eclipse-workspace	2 years ago
Favorites	2 months ago
git	2 years ago
gradle	2 years ago
IdeaProjects	2 years ago

Figure 8: Opened the Jupyter Notebook in Google Chrome

Then I selected the “Desktop” folder, and selected the folder named as “Marriage Year Prediction”. This folder consists of all the related files which are needed for this project. To open the Jupyter Notebook file, I selected the “Marriage age prediction.ipynb”. This is shown in (Figure 9).



localhost:8888/tree/Desktop/Marriage%20Year%20Prediction

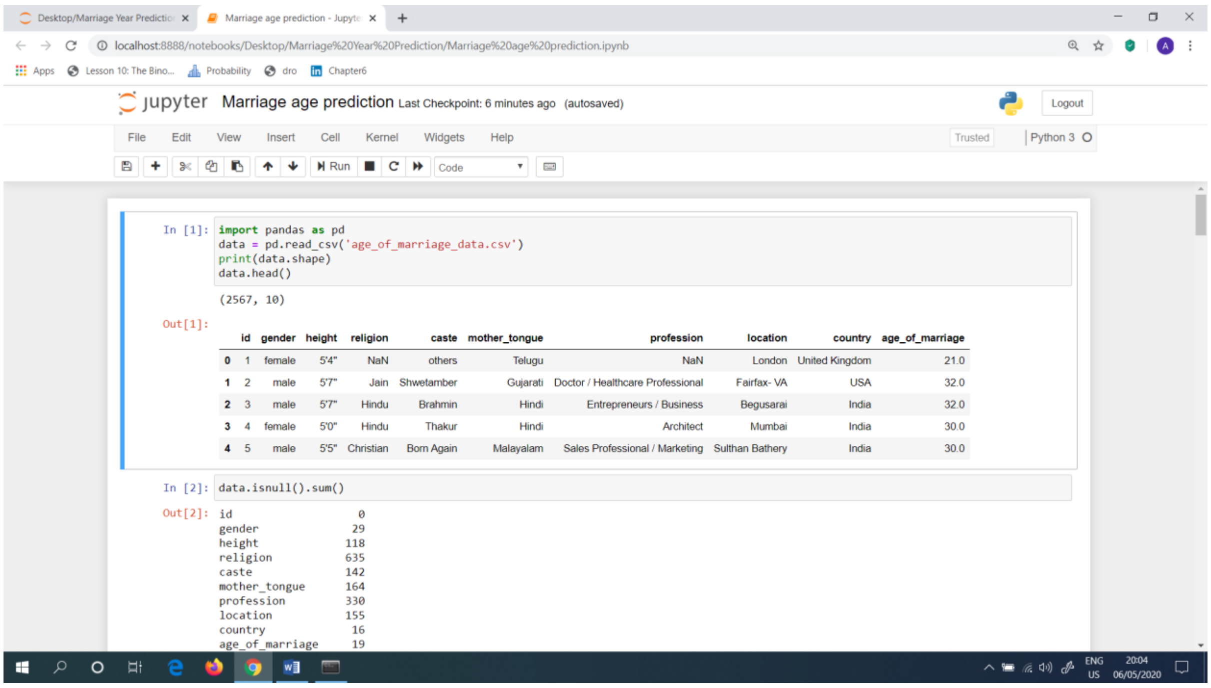
Select items to perform actions on them.

Desktop / Marriage Year Prediction

Name		Last Modified	File size
..		seconds ago	
Marriage age prediction.ipynb	Running	5 minutes ago	43.4 kB
Testing.ipynb		2 days ago	1.11 kB
age_of_marriage_data.csv		3 months ago	202 kB
marriage_age_predict_model.ml		2 days ago	3.07 MB

Figure 9: Opened the Marriage Year Prediction Project Folder

After I clicked the "Marriage age prediction.ipynb" file, it prompts a python interface, Jupyter Notebook which is shown in (Figure 10).



localhost:8888/notebooks/Desktop/Marriage%20Year%20Prediction/Marriage%20age%20prediction.ipynb

Marriage age prediction Last Checkpoint: 6 minutes ago (autosaved)

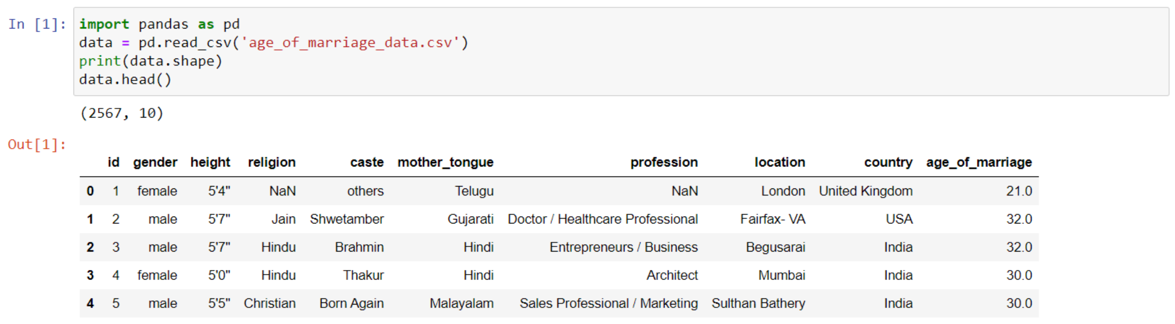
```
In [1]: import pandas as pd
        data = pd.read_csv('age_of_marriage_data.csv')
        print(data.shape)
        data.head()

        (2567, 10)

Out[1]:
```

	id	gender	height	religion	caste	mother_tongue	profession	location	country	age_of_marriage
0	1	female	5'4"	NaN	others	Telugu	NaN	London	United Kingdom	21.0
1	2	male	5'7"	Jain	Shwetamber	Gujarati	Doctor / Healthcare Professional	Fairfax- VA	USA	32.0
2	3	male	5'7"	Hindu	Brahmin	Hindi	Entrepreneurs / Business	Begusarai	India	32.0
3	4	female	5'0"	Hindu	Thakur	Hindi	Architect	Mumbai	India	30.0
4	5	male	5'5"	Christian	Born Again	Malayalam	Sales Professional / Marketing	Sulthan Bathery	India	30.0

```
In [2]: data.isnull().sum()

Out[2]: id                  0
        gender             29
        height            118
        religion          635
        caste             142
        mother_tongue     164
        profession        330
        location          155
        country            16
        age_of_marriage    19
```

Figure 10: Interface for the Marriage age prediction.ipynb

Next, first I imported the packages as shown in (Figure 11). Then I imported the .csv file, which I already created in the jupyter notebook. Again, I typed the command as "print(data.shape)" in order to check how many rows and columns available in the imported .csv file. And I came to know that 2567 rows and 10 columns available. Next, by typing the "data.head()" function, which helped to see the overview picture of the table.



```
In [1]: import pandas as pd
        data = pd.read_csv('age_of_marriage_data.csv')
        print(data.shape)
        data.head()

        (2567, 10)

Out[1]:
```

	id	gender	height	religion	caste	mother_tongue	profession	location	country	age_of_marriage
0	1	female	5'4"	NaN	others	Telugu	NaN	London	United Kingdom	21.0
1	2	male	5'7"	Jain	Shwetamber	Gujarati	Doctor / Healthcare Professional	Fairfax- VA	USA	32.0
2	3	male	5'7"	Hindu	Brahmin	Hindi	Entrepreneurs / Business	Begusarai	India	32.0
3	4	female	5'0"	Hindu	Thakur	Hindi	Architect	Mumbai	India	30.0
4	5	male	5'5"	Christian	Born Again	Malayalam	Sales Professional / Marketing	Sulthan Bathery	India	30.0

Figure 11: Imported the .csv file and came to know the count of rows and columns available in the data set

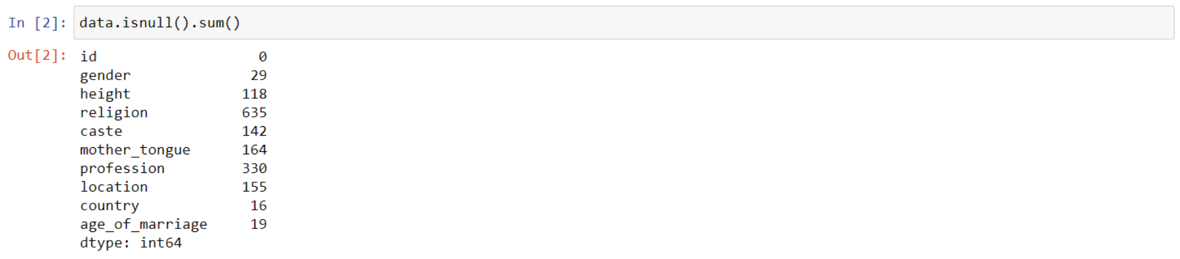
The table which I got as the output consists of "gender", "height", "religion", "caste", "mother_tongue", "profession", "location", "country" and "age_of_marriage".

And in the "religion" column I identified the "NaN" value. Which means, "Not a Number". The definition for "NaN" which is used to describe a non-numeric value. It also helps to indicates that the monitoring system is not receiving any numeric data. So the value of "NaN" need to be treated.

Mostly this data set which I used to do this project are mainly based for Indians. Indians, those who lives in abroad also included in this data set.

But, in (Figure 11) the first record seems to be an outlier. Because the "age_of_marriage" is 21 years, as it is too young age.

Next, I have planned to do a bit of analysis, that how many duplicates are presented. This is shown in (Figure 12).

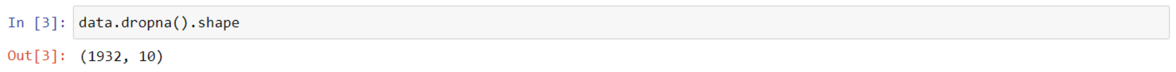


```
In [2]: data.isnull().sum()

Out[2]: 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
```

Figure 12: Checked the duplicates

In this above gained output, I came to know that the "profession" have 330 records. And also the "religion" is quite high. I.e.: There are 635 records.

Next, I dropped some records and saw the final number of records. This is shown in (Figure 13).

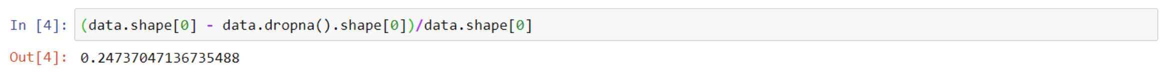


```
In [3]: data.dropna().shape

Out[3]: (1932, 10)
```

Figure 13: Dropped some records

From the above output, I came to know that there are 1932 rows and 10 columns available.

Next, I checked the percentage calculation, for the above mentioned same records, in order to get the percentage (Figure 14).

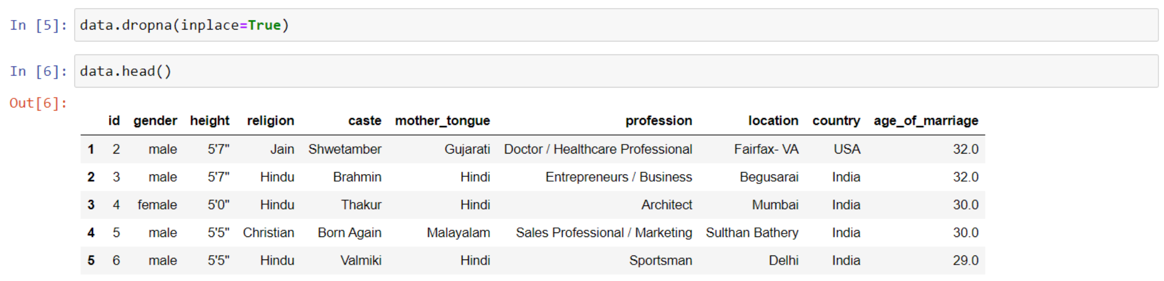


```
In [4]: (data.shape[0] - data.dropna().shape[0])/data.shape[0]
Out[4]: 0.24737047136735488
```

Figure 14: Calculated the Percentage

From the above output, 24% records had missing values. Ideally, I thought to treat the missing values. This process is known as “Imputation”.

Next, I dropped all the records. And then again, I looked at the data. This is shown in (Figure 15).

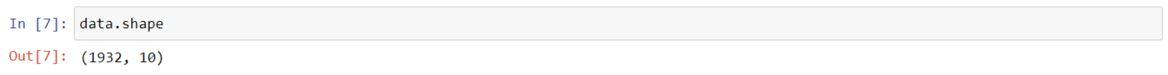


```
In [5]: data.dropna(inplace=True)

In [6]: data.head()
Out[6]:
```

	id	gender	height	religion	caste	mother_tongue	profession	location	country	age_of_marriage
1	2	male	5'7"	Jain	Shwetamber	Gujarati	Doctor / Healthcare Professional	Fairfax- VA	USA	32.0
2	3	male	5'7"	Hindu	Brahmin	Hindi	Entrepreneurs / Business	Begusarai	India	32.0
3	4	female	5'0"	Hindu	Thakur	Hindi	Architect	Mumbai	India	30.0
4	5	male	5'5"	Christian	Born Again	Malayalam	Sales Professional / Marketing	Sulthan Bathery	India	30.0
5	6	male	5'5"	Hindu	Valmiki	Hindi	Sportsman	Delhi	India	29.0

Figure 15: Checked the data again

To identify the rows and columns, I again typed the command as “data.shape”. This is shown in (Figure 16).

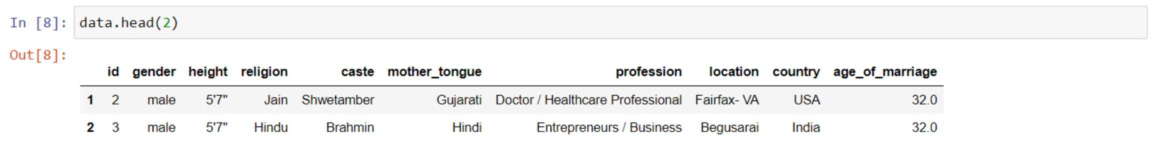


```
In [7]: data.shape
Out[7]: (1932, 10)
```

Figure 16: Checked the number of rows and columns

From the above output, there are 1932 rows and 10 columns available.

Next, I have done the feature engineering. So, I only picked up certain variables, which will be very useful to the model. In this case, the “gender” is definitely needed. Because, normally in India, the Male marry little bit later, compared to Females. “height” is also needed. Because when a certain male/ female marry, the height of both parties also considered for a marriage. The marriage is depended on the “religion” and “caste”. “mother_tongue” and “location” are same. Because, if a certain person is from a certain location, then the mother_tongue of that particular person will also be same. “country” is also matters. Because, as shown in (Figure 17) there are chances to marry a bit later. Finally, the age_of_marriage is the target variable. Because, first I planned to calculate the age of marriage. Then only, with the help of the predicted age, I developed a system which calculated the marriage of the year.

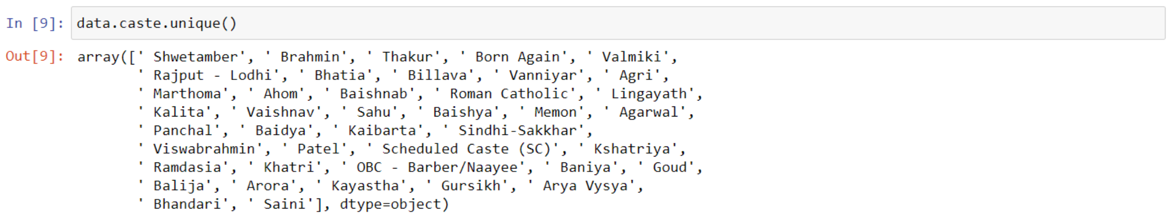


```
In [8]: data.head(2)
Out[8]:
```

	id	gender	height	religion	caste	mother_tongue	profession	location	country	age_of_marriage
1	2	male	5'7"	Jain	Shwetamber	Gujarati	Doctor / Healthcare Professional	Fairfax- VA	USA	32.0
2	3	male	5'7"	Hindu	Brahmin	Hindi	Entrepreneurs / Business	Begusarai	India	32.0

Figure 17: Checked the features needed to do feature engineering

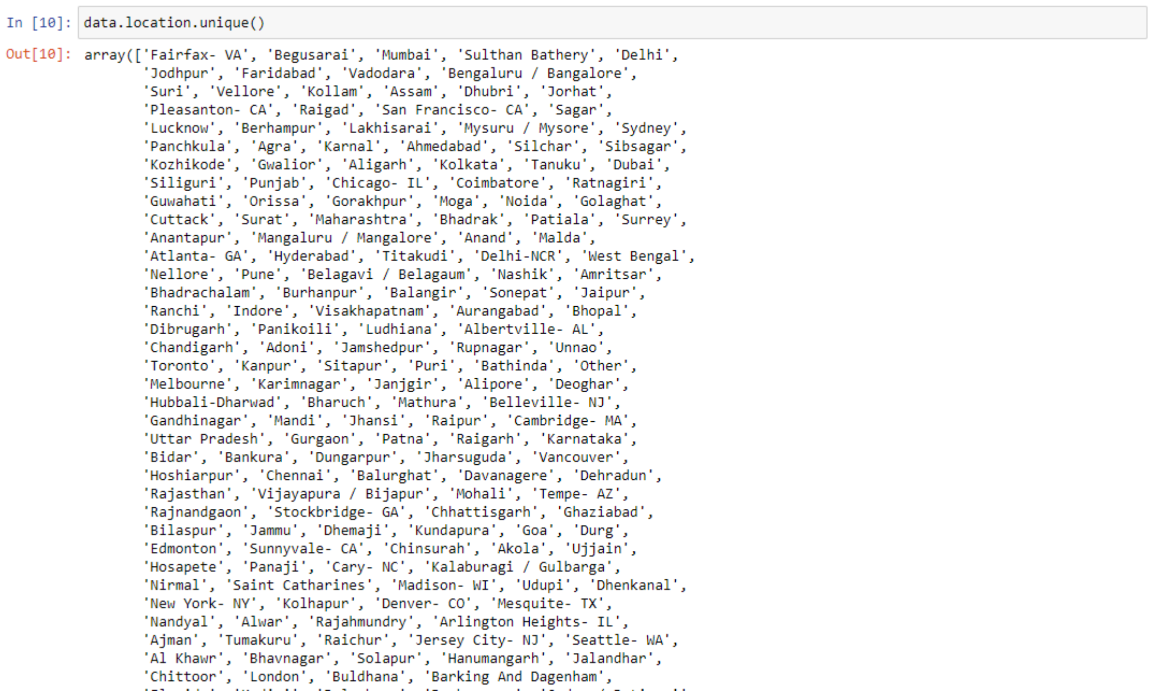
Then, to know how many unique values the “caste” consists of, I typed the command as shown in (Figure 18). In this, as the data are simple, it was easy to manage.



```
In [9]:  data.caste.unique()

Out[9]:  array([' Shwetamber', ' Brahmin', ' Thakur', ' Born Again', ' Valmiki',
                ' Rajput - Lodhi', ' Bhatia', ' Billava', ' Vanniyar', ' Agri',
                ' Marthoma', ' Ahom', ' Baishnab', ' Roman Catholic', ' Lingayath',
                ' Kalita', ' Vaishnav', ' Sahu', ' Baishya', ' Memon', ' Agarwal',
                ' Panchal', ' Baidya', ' Kaibarta', ' Sindhi-Sakkhar',
                ' Viswabrahmin', ' Patel', ' Scheduled Caste (SC)', ' Kshatriya',
                ' Ramdasia', ' Khatri', ' OBC - Barber/Naayee', ' Baniya', ' Goud',
                ' Balija', ' Arora', ' Kayastha', ' Gursikh', ' Arya Vysya',
                ' Bhandari', ' Saini'], dtype=object)
```

Figure 18: Unique values for “caste”

Next, I also saw the unique values for “location” (Figure 19).

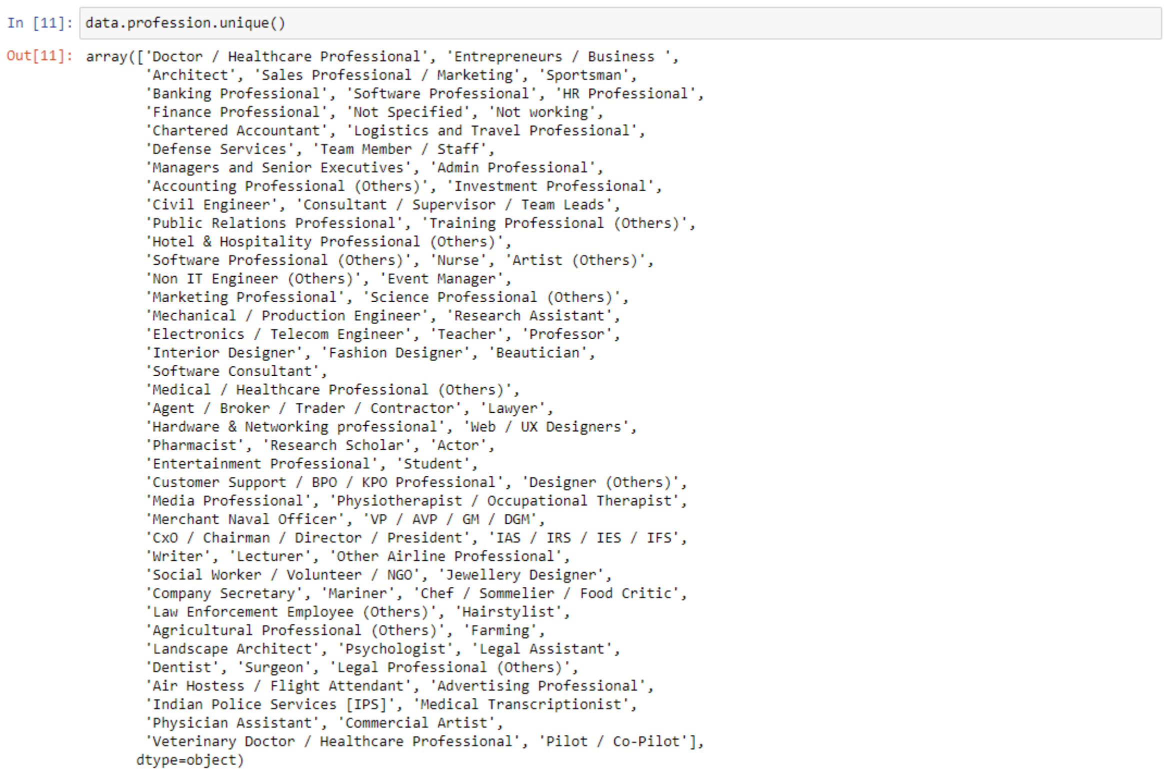


```
In [10]:  data.location.unique()

Out[10]:  array(['Fairfax- VA', 'Begusarai', 'Mumbai', 'Sulthan Bathery', 'Delhi',
                 'Jodhpur', 'Faridabad', 'Vadodara', 'Bengaluru / Bangalore',
                 'Suri', 'Vellore', 'Kollam', 'Assam', 'Dhubri', 'Jorhat',
                 'Pleasanton- CA', 'Raigad', 'San Francisco- CA', 'Sagar',
                 'Lucknow', 'Berhampur', 'Lakhisarai', 'Mysuru / Mysore', 'Sydney',
                 'Panchkula', 'Agra', 'Karnal', 'Ahmedabad', 'Silchar', 'Sibsagar',
                 'Kozhikode', 'Gwalior', 'Aligarh', 'Kolkata', 'Tanuku', 'Dubai',
                 'Siliguri', 'Punjab', 'Chicago- IL', 'Coimbatore', 'Ratnagiri',
                 'Guwahati', 'Orissa', 'Gorakhpur', 'Moga', 'Noida', 'Golaghat',
                 'Cuttack', 'Surat', 'Maharashtra', 'Bhadrak', 'Patiala', 'Surrey',
                 'Anantapur', 'Mangaluru / Mangalore', 'Anand', 'Malda',
                 'Atlanta- GA', 'Hyderabad', 'Titakudi', 'Delhi-NCR', 'West Bengal',
                 'Nellore', 'Pune', 'Belagavi / Belagaum', 'Nashik', 'Amritsar',
                 'Bhadrachalam', 'Burhanpur', 'Balangir', 'Sonepat', 'Jaipur',
                 'Ranchi', 'Indore', 'Visakhapatnam', 'Aurangabad', 'Bhopal',
                 'Dibrugarh', 'Panikoili', 'Ludhiana', 'Albertville- AL',
                 'Chandigarh', 'Adoni', 'Jamshedpur', 'Rupnagar', 'Unnao',
                 'Toronto', 'Kanpur', 'Sitapur', 'Puri', 'Bathinda', 'Other',
                 'Melbourne', 'Karimnagar', 'Janjgir', 'Alipore', 'Deoghar',
                 'Hubbali-Dharwad', 'Bharuch', 'Mathura', 'Belleville- NJ',
                 'Gandhinagar', 'Mandi', 'Jhansi', 'Raipur', 'Cambridge- MA',
                 'Uttar Pradesh', 'Gurgaon', 'Patna', 'Raigarh', 'Karnataka',
                 'Bidar', 'Bankura', 'Dungarpur', 'Jharsuguda', 'Vancouver',
                 'Hoshiarpur', 'Chennai', 'Balurghat', 'Davanagere', 'Dehradun',
                 'Rajasthan', 'Vijayapura / Bijapur', 'Mohali', 'Tempe- AZ',
                 'Rajnandgaon', 'Stockbridge- GA', 'Chhattisgarh', 'Ghaziabad',
                 'Bilaspur', 'Jammu', 'Dhemaji', 'Kundapura', 'Goa', 'Durg',
                 'Edmonton', 'Sunnyvale- CA', 'Chinsurah', 'Akola', 'Ujjain',
                 'Hosapete', 'Panaji', 'Cary- NC', 'Kalaburagi / Gulbarga',
                 'Nirmal', 'Saint Catharines', 'Madison- WI', 'Udupi', 'Dhenkanal',
                 'New York- NY', 'Kolhapur', 'Denver- CO', 'Mesquite- TX',
                 'Nandyal', 'Alwar', 'Rajahmundry', 'Arlington Heights- IL',
                 'Ajman', 'Tumakuru', 'Raichur', 'Jersey City- NJ', 'Seattle- WA',
                 'Al Khawr', 'Bhavnagar', 'Solapur', 'Hanumangarh', 'Jalandhar',
                 'Chittoor', 'London', 'Buldhana', 'Barking And Dagenham',
```

Figure 19: Unique values for “location”

In (Figure 19) it consists of many data.

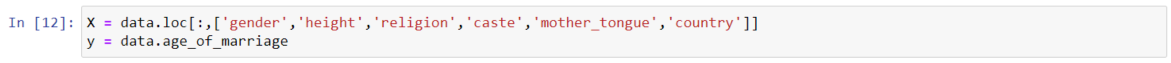
Next, I saw the unique values for “profession” (Figure 20).



```
In [11]: data.profession.unique()

Out[11]: array(['Doctor / Healthcare Professional', 'Entrepreneurs / Business ',
                'Architect', 'Sales Professional / Marketing', 'Sportsman',
                'Banking Professional', 'Software Professional', 'HR Professional',
                'Finance Professional', 'Not Specified', 'Not working',
                'Chartered Accountant', 'Logistics and Travel Professional',
                'Defense Services', 'Team Member / Staff',
                'Managers and Senior Executives', 'Admin Professional',
                'Accounting Professional (Others)', 'Investment Professional',
                'Civil Engineer', 'Consultant / Supervisor / Team Leads',
                'Public Relations Professional', 'Training Professional (Others)',
                'Hotel & Hospitality Professional (Others)',
                'Software Professional (Others)', 'Nurse', 'Artist (Others)',
                'Non IT Engineer (Others)', 'Event Manager',
                'Marketing Professional', 'Science Professional (Others)',
                'Mechanical / Production Engineer', 'Research Assistant',
                'Electronics / Telecom Engineer', 'Teacher', 'Professor',
                'Interior Designer', 'Fashion Designer', 'Beautician',
                'Software Consultant',
                'Medical / Healthcare Professional (Others)',
                'Agent / Broker / Trader / Contractor', 'Lawyer',
                'Hardware & Networking professional', 'Web / UX Designers',
                'Pharmacist', 'Research Scholar', 'Actor',
                'Entertainment Professional', 'Student',
                'Customer Support / BPO / KPO Professional', 'Designer (Others)',
                'Media Professional', 'Physiotherapist / Occupational Therapist',
                'Merchant Naval Officer', 'VP / AVP / GM / DGM',
                'CxO / Chairman / Director / President', 'IAS / IRS / IES / IFS',
                'Writer', 'Lecturer', 'Other Airline Professional',
                'Social Worker / Volunteer / NGO', 'Jewellery Designer',
                'Company Secretary', 'Mariner', 'Chef / Sommelier / Food Critic',
                'Law Enforcement Employee (Others)', 'Hairstylist',
                'Agricultural Professional (Others)', 'Farming',
                'Landscape Architect', 'Psychologist', 'Legal Assistant',
                'Dentist', 'Surgeon', 'Legal Professional (Others)',
                'Air Hostess / Flight Attendant', 'Advertising Professional',
                'Indian Police Services [IPS]', 'Medical Transcriptionist',
                'Physician Assistant', 'Commercial Artist',
                'Veterinary Doctor / Healthcare Professional', 'Pilot / Co-Pilot'],
               dtype=object)
```

Figure 19: Unique values for "profession"

Next, I took "X" and "Y", as shown in (Figure 20).

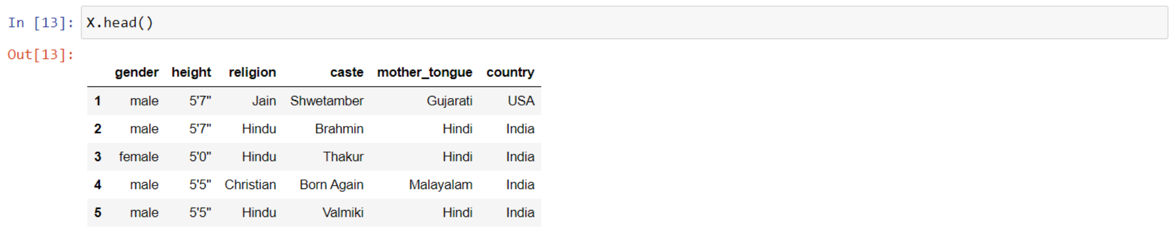


```
In [12]: X = data.loc[:,['gender','height','religion','caste','mother_tongue','country']]
         y = data.age_of_marriage
```

Figure 20: Typed "X" and "Y"

The algorithm which I used was "Random Forest Algorithm".

Next, I have done the "Label Encoding". Because, as shown in (Figure 21), they all are in Strings. So, I need to convert them into numbers.

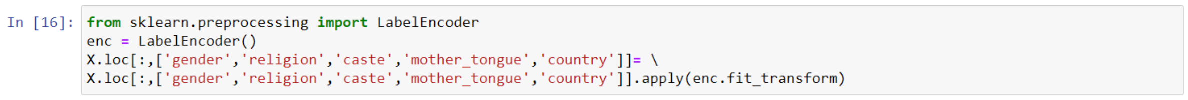


```
In [13]: X.head()

Out[13]:
```

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

Figure 21: The values are in "Strings"

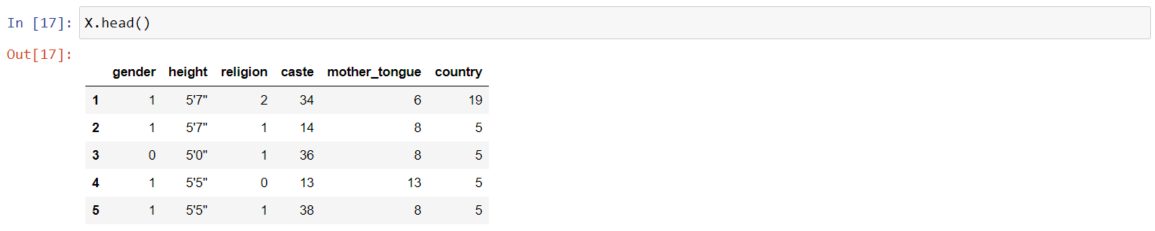
Then, I have done the label encoding as shown in (Figure 22). The "\" is used in order to go to the next line. I have done the encoding for "X".



```
In [16]: from sklearn.preprocessing import LabelEncoder
         enc = LabelEncoder()
         X.loc[:,['gender','religion','caste','mother_tongue','country']]= \
         X.loc[:,['gender','religion','caste','mother_tongue','country']].apply(enc.fit_transform)
```

Figure 22: Label Encoding

Next, I again done the “X.head()”, which produced the output as shown in (Figure 23). This output was gained after done the label encoding.



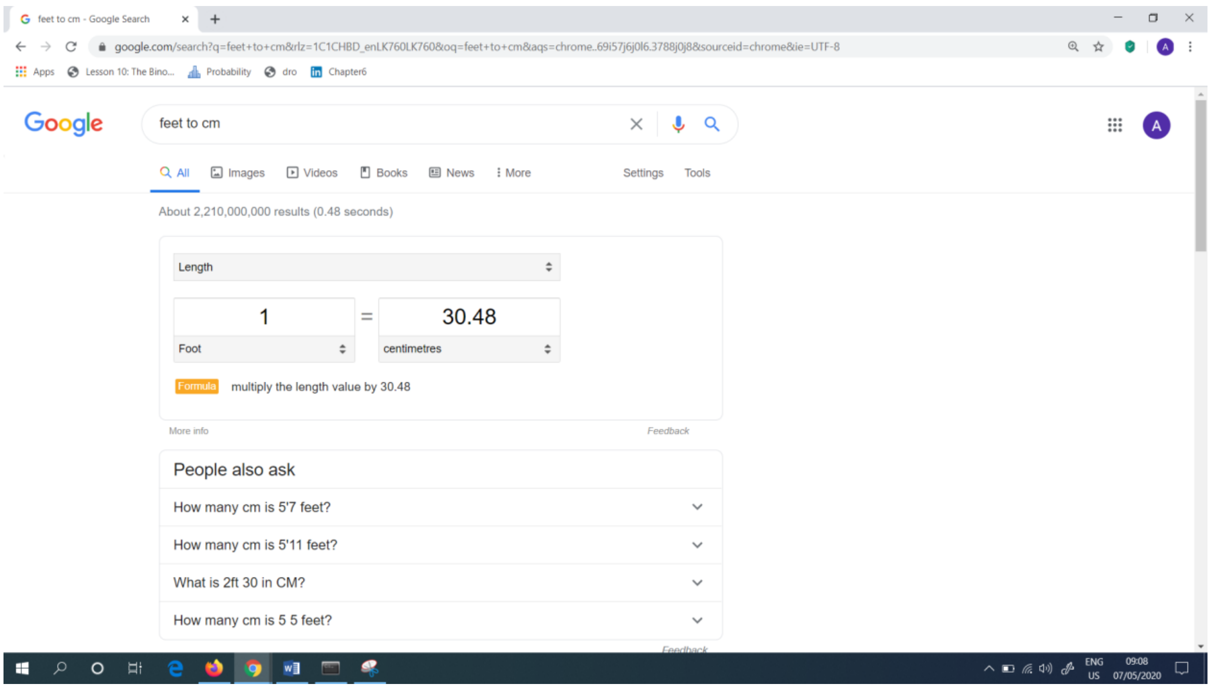
```
In [17]: X.head()
Out[17]:
```

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

Figure 23: Output gained after done the label encoding

From the above derived table, I came to know that, the “height” is not a numerical field. Height is presented in feet and inches. So, I planned to convert the particular field into centimetres (cm).

So, I converted the feet and inches, into centimetres. So, I opened the Google chrome and typed as “feet to cm”, in order to check the value (Figure 24).


feet to cm - Google Search
google.com/search?q=feet+to+cm&rlz=1C1CHBD_enLK760LK760&oq=feet+to+cm&aqs=chrome..69i57j6j0l6.3788j0j8&sourceid=chrome&ie=UTF-8
Apps
Lesson 10: The Bino...
Probability
dro
Chapter6
Google
feet to cm
All
Images
Videos
Books
News
More
Settings
Tools

About 2,210,000,000 results (0.48 seconds)

Length

1 Foot = 30.48 centimetres

Formula multiply the length value by 30.48

People also ask

How many cm is 5'7 feet?

How many cm is 5'11 feet?

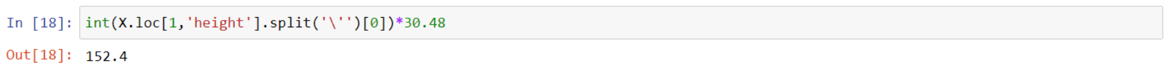
What is 2ft 30 in CM?

How many cm is 5 5 feet?

Feedback
ENG
US
09:08
07/05/2020

Figure 24: Conversion of length in feet to cm

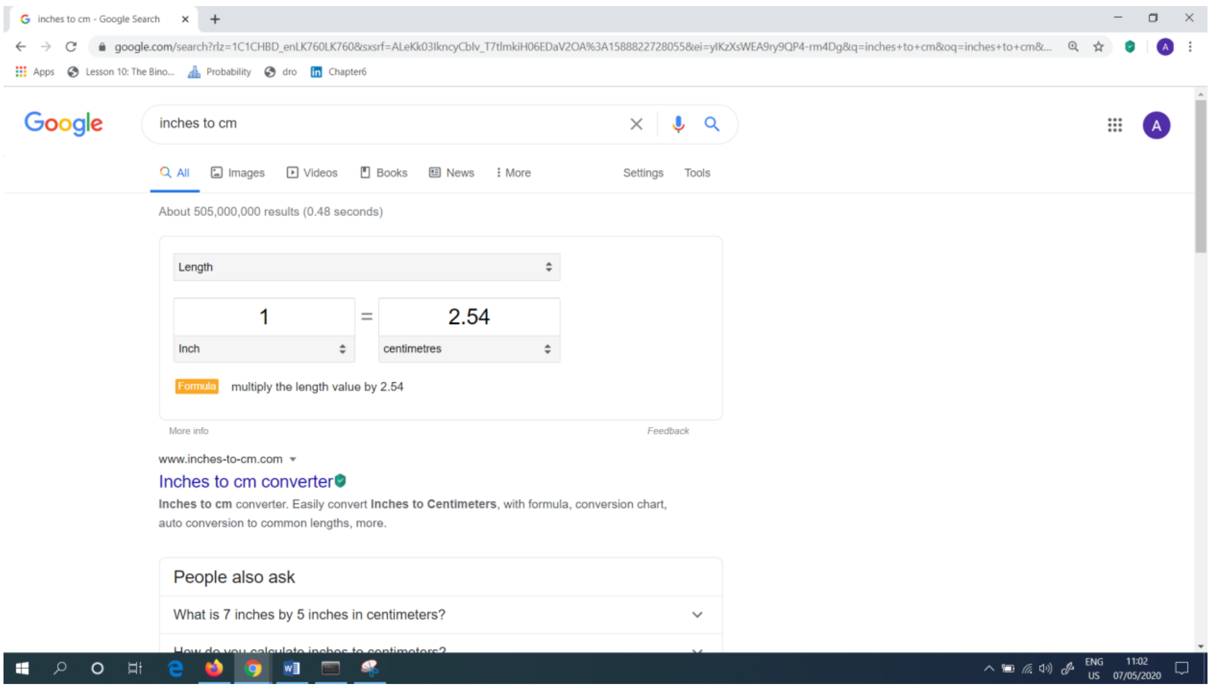
So, I multiplied the height value by 30.48 as shown in (Figure 25). The output was gained in centimetres (cm). In this part, I converted the feet to centimetres.



```
In [18]: int(X.loc[1,'height'].split('\'')[0])*30.48
Out[18]: 152.4
```

Figure 25: Output gained in centimetres

Next, I also converted the inches into centimetres. To do this, I went again to Google Chrome and typed as inches to centimetres (cm) as shown in (Figure 26).



inches to cm - Google Search

google.com/search?rlz=1C1CHBD_enLK760LK760&sxsrf=ALeKk03lkncyCblv_T7tlmkiH06EDaV2OA%3A1588822728055&ei=ylKzXsWEA9ry9QP4-rm4Dg&q=inches+to+cm&oq=inches+to+cm&...

Apps Lesson 10: The Bino... Probability dro Chapter6

inches to cm

All Images Videos Books News More Settings Tools

About 505,000,000 results (0.48 seconds)

Length

1 = 2.54

Inch centimetres

Formula multiply the length value by 2.54

More info Feedback

www.inches-to-cm.com

Inches to cm converter

Inches to cm converter. Easily convert **Inches to Centimeters**, with formula, conversion chart, auto conversion to common lengths, more.

People also ask

What is 7 inches by 5 inches in centimeters?

ENG US 11:02 07/05/2020

Figure 26: Conversion of length in inches to cm

Next, I wrote the following command as shown in (Figure 27), where I replaced the function. Also I multiplied it by 2.54. In this part, I converted the inches to centimetres.

```
In [19]: int(X.loc[1,'height'].split('\'')[1].replace('"',''))*2.54

Out[19]: 17.78
```

Figure 27: Output gained in centimetres

Next, I combined the above written two commands. So, I wrote a function for this as shown in (Figure 28).

```
In [20]: def h_cms(h):
             return int(h.split('\'')[0])*30.48+\
             int(h.split('\'')[1].replace('"',''))*2.54
```

Figure 28: Combined the two commands with the function

Next, I applied the above generated commands for the height. In this, I created a new column named as "height_cms". This is shown in (Figure 29).

```
In [21]: X['height_cms'] = X.height.apply(h_cms)
```

Figure 29: Combined the two commands with the function

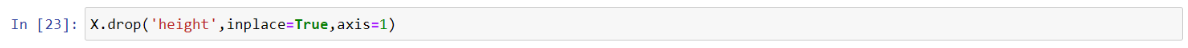
Then, I was able to see whether the "height" have been converted into the new column named as "height_cms". (Figure 30) shows that the new column has been created.

```
In [22]: X.head()

Out[22]:
```

	gender	height	religion	caste	mother_tongue	country	height_cms
1	1	5'7"	2	34	6	19	170.18
2	1	5'7"	1	14	8	5	170.18
3	0	5'0"	1	36	8	5	152.40
4	1	5'5"	0	13	13	5	165.10
5	1	5'5"	1	38	8	5	165.10

Figure 30: New column is created with the name of “height_cms”

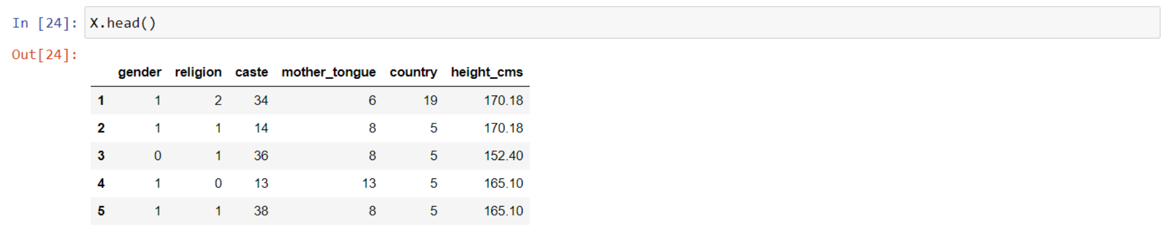
Next, as the new column have been created, I thought that the column named as “height” is not needed. So, I have written the command as shown in (Figure 31).



```
In [23]: X.drop('height',inplace=True,axis=1)
```

Figure 31: Dropped the “height” column

Finally, I again checked the table by typing the command as “X.head()”. This is shown in (Figure 32).

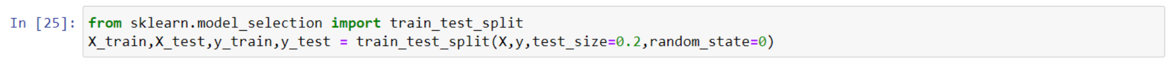


```
In [24]: X.head()
Out[24]:
```

	gender	religion	caste	mother_tongue	country	height_cms
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

Figure 32: Got the cleaned data as output

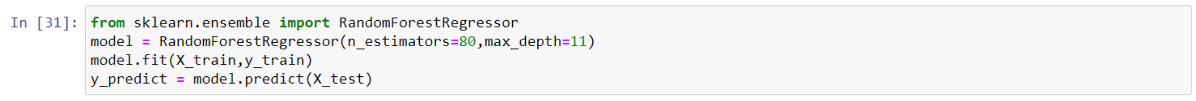
Up to now, all the data cleaning was done and the data is clear. Then, I thought to build the model. To do this, next I done the data splitting. In this, I put 20% as the test size, due to less number of data. I also put, random_state = 0, so that shuffling will be presented. The command I wrote is shown in (Figure 33) for this purpose.



```
In [25]: 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=0)
```

Figure 33: Splitting the data

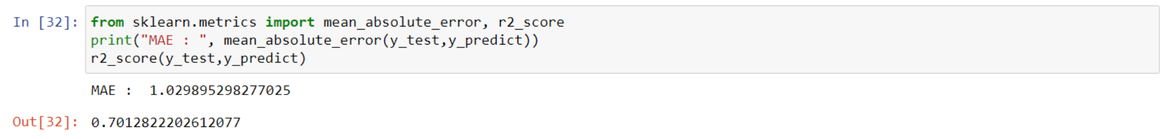
Next, I done the modelling part. In this, as “age” is the continuous value, I used the “RandomForestRegressor”. I have also done the fine tunning in order to get the better results. The algorithm I used was “Random forest”. All the commands are shown in (Figure 34).



```
In [31]: from sklearn.ensemble import RandomForestRegressor
         model = RandomForestRegressor(n_estimators=80,max_depth=11)
         model.fit(X_train,y_train)
         y_predict = model.predict(X_test)
```

Figure 34: Modelling for the model

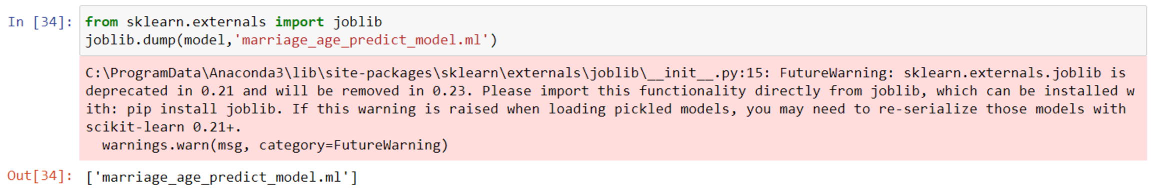
Under the evaluation of the model, I wrote the commands as shown in (Figure 35). The developed model predicts the marriage age with one year. As there are only few variables I had taken, I think that the produced “mean_absolute_error” (MAE) was fine. I also printed the “r2_score”. It gave the output as 70% for r2_score.



```
In [32]: from sklearn.metrics import mean_absolute_error, r2_score
         print("MAE : ", mean_absolute_error(y_test,y_predict))
         r2_score(y_test,y_predict)

         MAE :  1.029895298277025

Out[32]: 0.7012822202612077
```

Figure 35: Produced the MAE as 1.0298 and r2_score as 70%

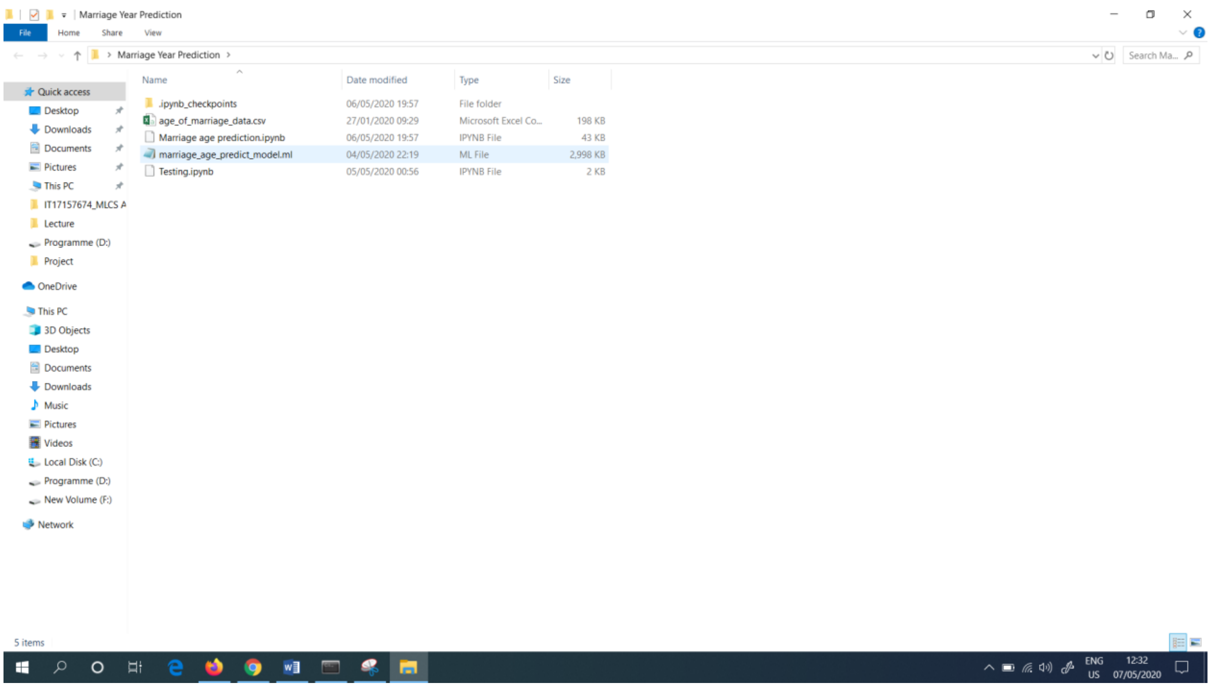
Next, in order to deploy the model in API form, so that it can be seen whether in webpage or anything. To do this, first I need to exported the model. Then, I dumped the model under the file name of "marriage_age_predict_model.ml". This is shown in (Figure 36).



```
In [34]: from sklearn.externals import joblib
         joblib.dump(model,'marriage_age_predict_model.ml')

         C:\ProgramData\Anaconda3\lib\site-packages\sklearn\externals\joblib\__init__.py:15: FutureWarning: sklearn.externals.joblib is
         deprecated in 0.21 and will be removed in 0.23. Please import this functionality directly from joblib, which can be installed w
         ith: pip install joblib. If this warning is raised when loading pickled models, you may need to re-serialize those models with
         scikit-learn 0.21+.
           warnings.warn(msg, category=FutureWarning)

Out[34]: ['marriage_age_predict_model.ml']
```

Figure 36: Created the Machine Learning (ML) model

So, the created "marriage_age_predict_model.ml" was produced under the folder as shown in (Figure 37). This created ml model file, was a trained model.



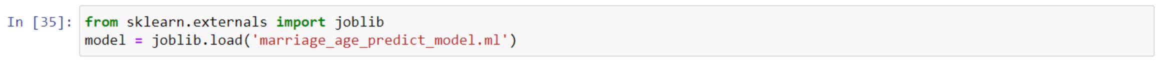
Marriage Year Prediction

Name	Date modified	Type	Size
.ipynb_checkpoints	06/05/2020 19:57	File folder	
age_of_marriage_data.csv	27/01/2020 09:29	Microsoft Excel Co...	198 KB
Marriage age prediction.ipynb	06/05/2020 19:57	IPYNB File	43 KB
marriage_age_predict_model.ml	04/05/2020 22:19	ML File	2,998 KB
Testing.ipynb	05/05/2020 00:56	IPYNB File	2 KB

5 items

Figure 37: Created the Machine Learning (ML) model under the folder named as "Marriage Year Prediction"

Next, in order to see whether the created model is working or not, I used the exported model. Then I imported the model. Then, I used the same library as mentioned above (Figure 38) to see whether the model is working fine or not.



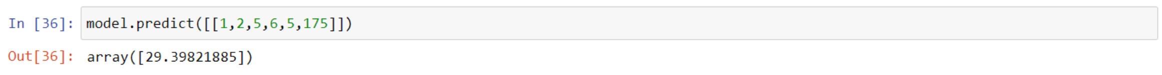
```
In [35]: from sklearn.externals import joblib
         model = joblib.load('marriage_age_predict_model.ml')
```

Figure 38: Imported the model

Then to check whether the model is working fine or not, I gave all the predictors as shown below. As the label encoding have been used it is hard to find the numbers for the variables like "gender", "height", "religion", "caste", "mother_tongue" and "country".

For example, I took the values as gender = male, height = 5'7" , religion = Hindu, caste = Brahmin, mother_tongue = Hindi and country = India.

As shown in (Figure 39) I typed the command and I ran it. The output I gained was 29 years (Predicted age of marriage).



```
In [36]:  model.predict([[1,2,5,6,5,175]])

Out[36]:  array([29.39821885])
```

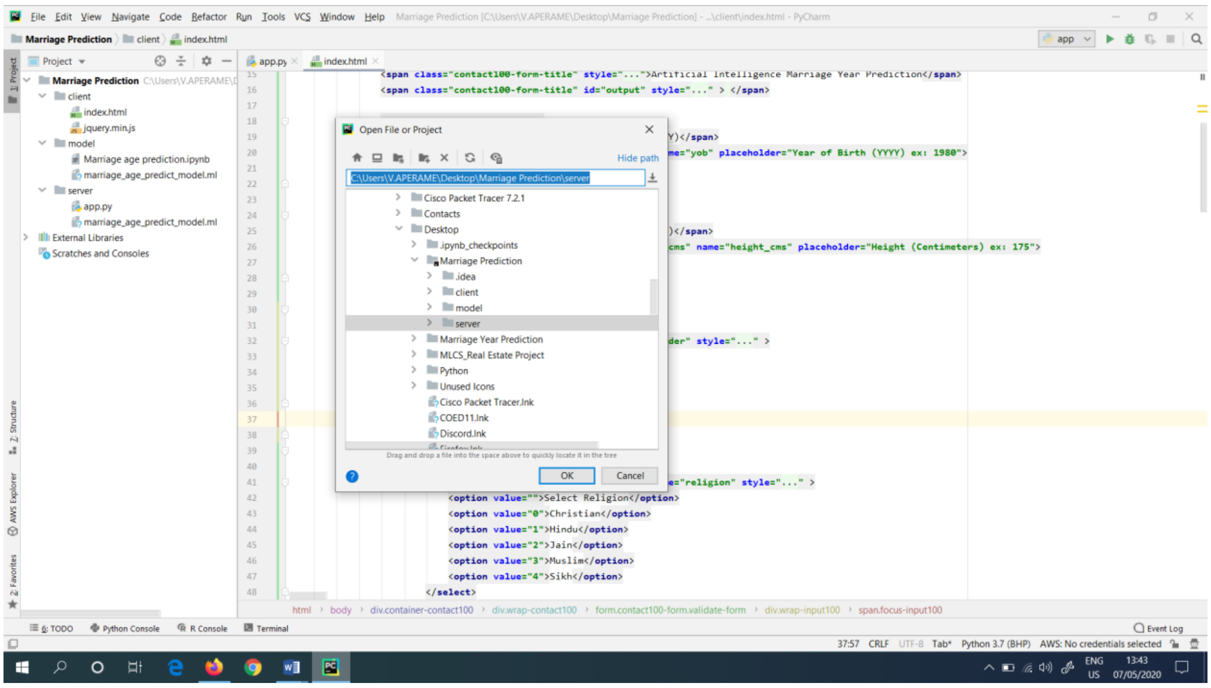
Figure 39: Predicted age of marriage

Up to now I have done the modelling evaluation, exported the model, loaded the model in a separate file and managed to predict the age with the help of the static value.

As the next step, I created the rest API, where anyone can request the service by sending the predictors and get the predicted marriage age.

Next to run the local rest API with the flask server, I created a python file in PyCharm under the name of "app.py". So, I opened the PyCharm. It is a community edition. So I downloaded it for free and I installed it in my PC (Personal Computer). And the version I used is 2019.3.3. Also it is a Python editor.

Next, I clicked the "File" and then I clicked the "Open" option. Then, I selected the file named as "Marriage Prediction" which I saved it in my desktop. This is shown in (Figure 40).



```
<span class="contact100-form-title" style="...">Artificial Intelligence Marriage Year Prediction</span>
<span class="contact100-form-title" id="output" style="..." > </span>

Y)</span>
me="yob" placeholder="Year of Birth (YYYY) ex: 1980">

)</span>
cms" name="height_cms" placeholder="Height (Centimeters) ex: 175">

der" style="..." >

e="religion" style="..." >
        <option value="">Select Religion</option>
        <option value="0">Christian</option>
        <option value="1">Hindu</option>
        <option value="2">Jain</option>
        <option value="3">Muslim</option>
        <option value="4">Sikh</option>
    </select>
```

Open File or Project

C:\Users\V.APERAME\Desktop\Marriage Prediction\server

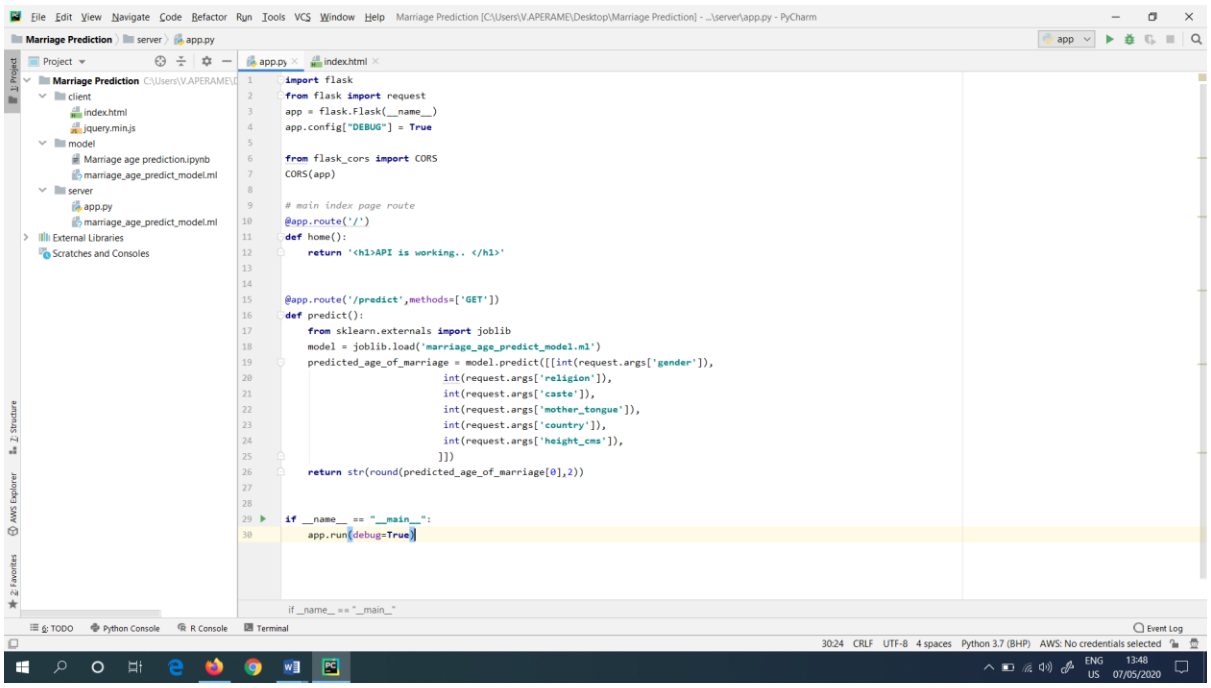
Drag and drop a file into the space above to quickly locate it in the tree

Marriage Prediction [C:\Users\V.APERAME\Desktop\Marriage Prediction] - ...\client\index.html - PyCharm

Figure 40: Opened the Project file in "PyCharm"

Next, I have created 3 folders namely client, model and server under the folder named as "Marriage Prediction". Under the client folder, the UI Applications are presented (HTML and jQuery files). Under the server folder, the Python Flask Server have presented under the name of "app.py". And the model folder contains the jupyter notebook and the machine learning model.

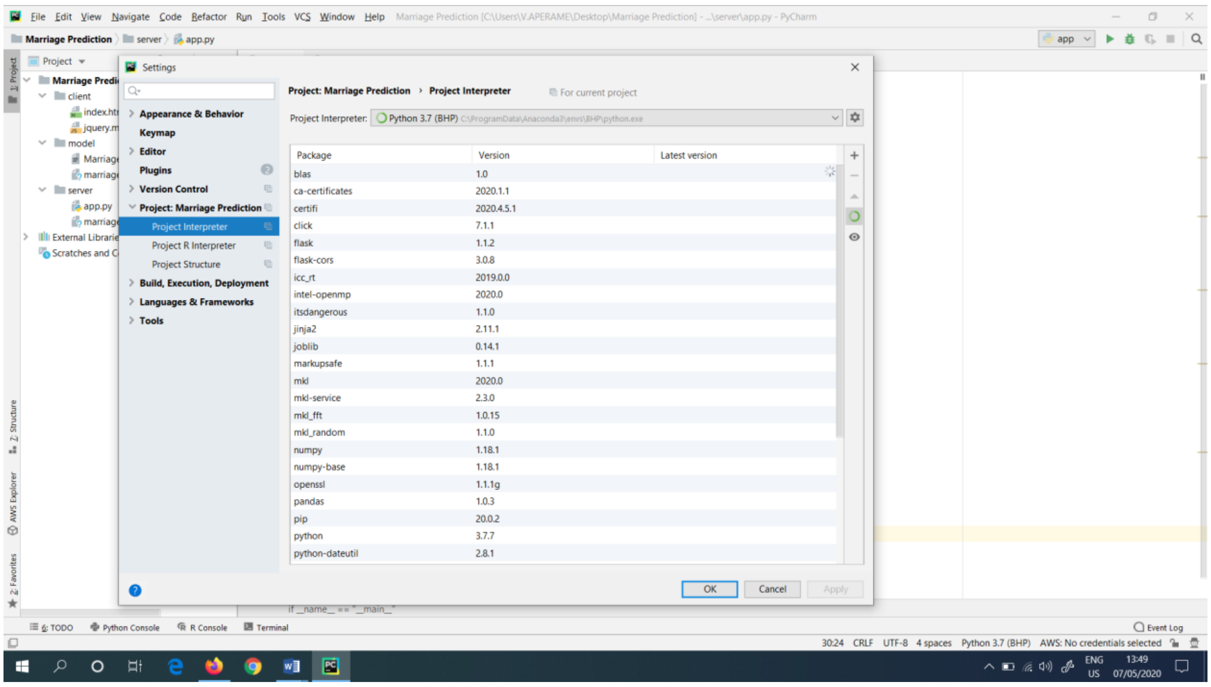
Then, in the PyCharm, under the server folder I created a file with the name as "app.py". This is shown in (Figure 41).


File Edit View Navigate Code Refactor Run Tools VCS Window Help
Marriage Prediction [C:\Users\V.APERAME\Desktop\Marriage Prediction] - ...\server\app.py - PyCharm
Marriage Prediction
server
app.py
index.html
client
jquery.min.js
model
Marriage age prediction.ipynb
marriage_age_predict_model.ml
External Libraries
Scratches and Consoles
TODO
Python Console
R Console
Terminal
Event Log

```python
import flask
from flask import request
app = flask.Flask(__name__)
app.config["DEBUG"] = True

from flask_cors import CORS
CORS(app)

# main index page route
@app.route('/')
def home():
    return '<h1>API is working.. </h1>'


@app.route('/predict',methods=['GET'])
def predict():
    from sklearn.externals import joblib
    model = joblib.load('marriage_age_predict_model.ml')
    predicted_age_of_marriage = model.predict([[int(request.args['gender']),
                            int(request.args['religion']),
                            int(request.args['caste']),
                            int(request.args['mother_tongue']),
                            int(request.args['country']),
                            int(request.args['height_cms']),
                           ]])
    return str(round(predicted_age_of_marriage[0],2))


if __name__ == "__main__":
    app.run(debug=True)
```

Figure 41: app.py file in "PyCharm"

Next, I configured the Interpreter to "Anaconda". Because Anaconda normally comes with Flask. So, first I went to "File". Then clicked the "Settings". Next, I searched for the project interpreter. This is shown in (Figure 42).



Project: Marriage Prediction › Project Interpreter

Project Interpreter: Python 3.7 (BHP) C:\ProgramData\Anaconda3\envs\BHP\python.exe

Package	Version	Latest version
blas	1.0	
ca-certificates	2020.1.1	
certifi	2020.4.5.1	
click	7.1.1	
flask	1.1.2	
flask-cors	3.0.8	
icc_rt	2019.0.0	
intel-openmp	2020.0	
itsdangerous	1.1.0	
jinja2	2.11.1	
joblib	0.14.1	
markupsafe	1.1.1	
mkl	2020.0	
mkl-service	2.3.0	
mkl_fft	1.0.15	
mkl_random	1.1.0	
numpy	1.18.1	
numpy-base	1.18.1	
openssl	1.1.1g	
pandas	1.0.3	
pip	20.0.2	
python	3.7.7	
python-dateutil	2.8.1	

Figure 42: Configured the Interpreter to "Anaconda"

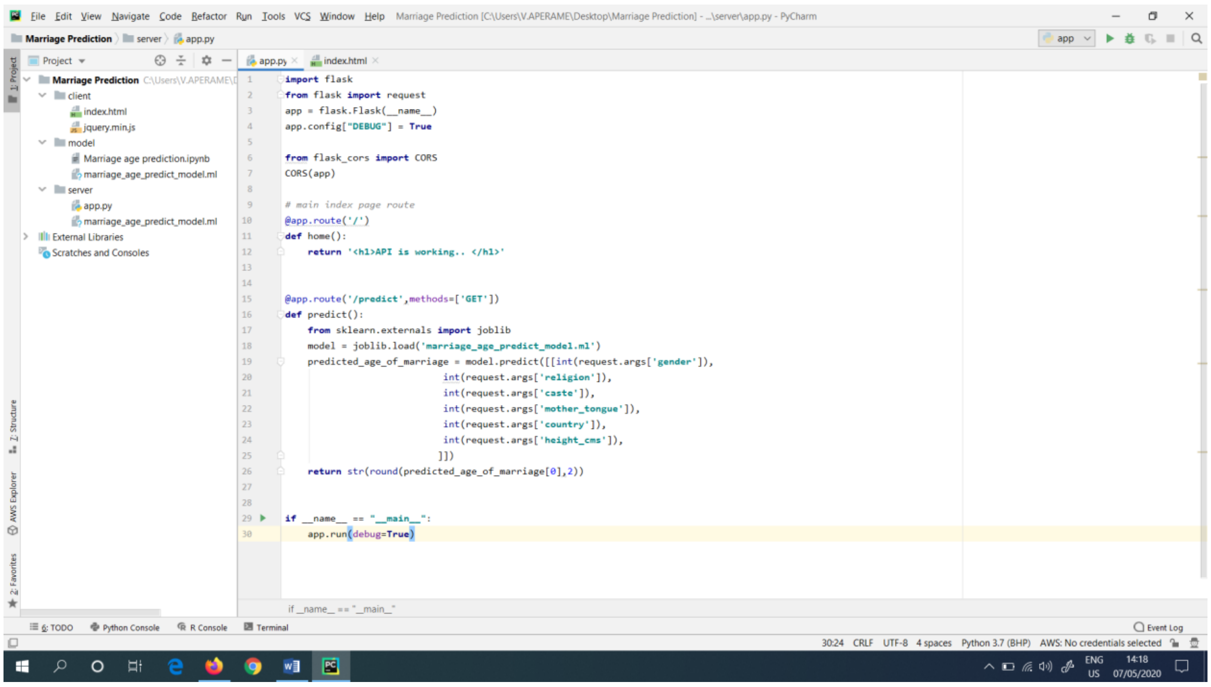
"app.py" is the main server file, where I imported the flask module. In this I installed the "flask" and "flask-cors".

In the “app.py”, python file, the “import request” is needed in order to read the URL parameters. Next, I initiated the app.py file. The CORS domain is needed in order to host in cross domain application. Then I wrote the default way to write the route.

Next, I created another route with the name as “predict”. Here the function name that I used was “predict”. In this, I imported the “joblib” and loaded the model. Then, I predicted the age of marriage. Finally, I returned the predicted age of marriage.

Then, I would like to have the dynamic value for the variables. To do that I had the flask request module. And also, I typed the command as “request.args[‘gender’]”, “request.args[‘religion’]”, “request.args[‘caste’]”, “request.args[‘mother_tongue’]”, “request.args[‘country’]”, “request.args[‘height_cms’]”, and replaced all, with the arguments. So that, I can pass all the arguments, in order to get the predicted age of marriage.

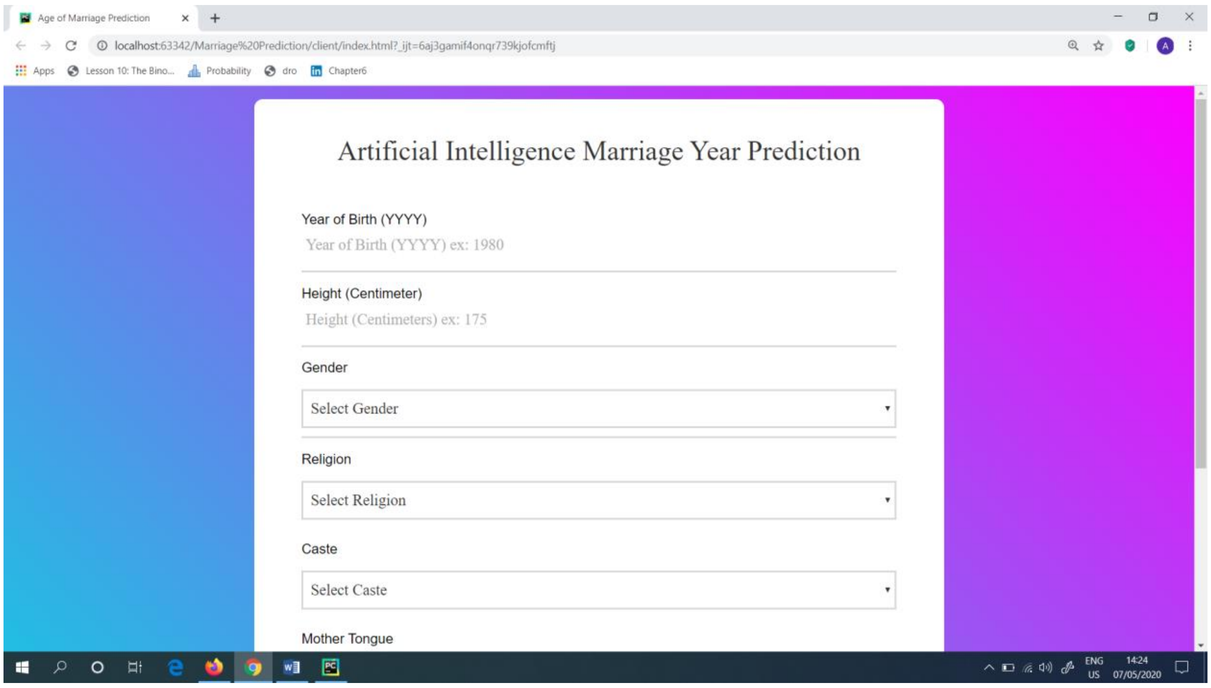
The following (Figure 43) shows the “app.py” file.



```python
import flask
from flask import request
app = flask.Flask(__name__)
app.config["DEBUG"] = True

from flask_cors import CORS
CORS(app)

# main index page route
@app.route('/')
def home():
    return '<h1>API is working.. </h1>'


@app.route('/predict',methods=['GET'])
def predict():
    from sklearn.externals import joblib
    model = joblib.load('marriage_age_predict_model.ml')
    predicted_age_of_marriage = model.predict([[int(request.args['gender']),
                                int(request.args['religion']),
                                int(request.args['caste']),
                                int(request.args['mother_tongue']),
                                int(request.args['country']),
                                int(request.args['height_cms']),
                                ]])
    return str(round(predicted_age_of_marriage[0],2))


if __name__ == "__main__":
    app.run(debug=True)
```

Figure 43: Created “app.py” file

Next, I created a simple html file, with the parameters like gender, religion, caste, mother tongue, country and height (cm). These are same parameters which I already used. The following (Figure 44) shows the interface of the created model for the age prediction.



Artificial Intelligence Marriage Year Prediction

Year of Birth (YYYY)

Year of Birth (YYYY) ex: 1980

Height (Centimeter)

Height (Centimeters) ex: 175

Gender

Select Gender

Religion

Select Religion

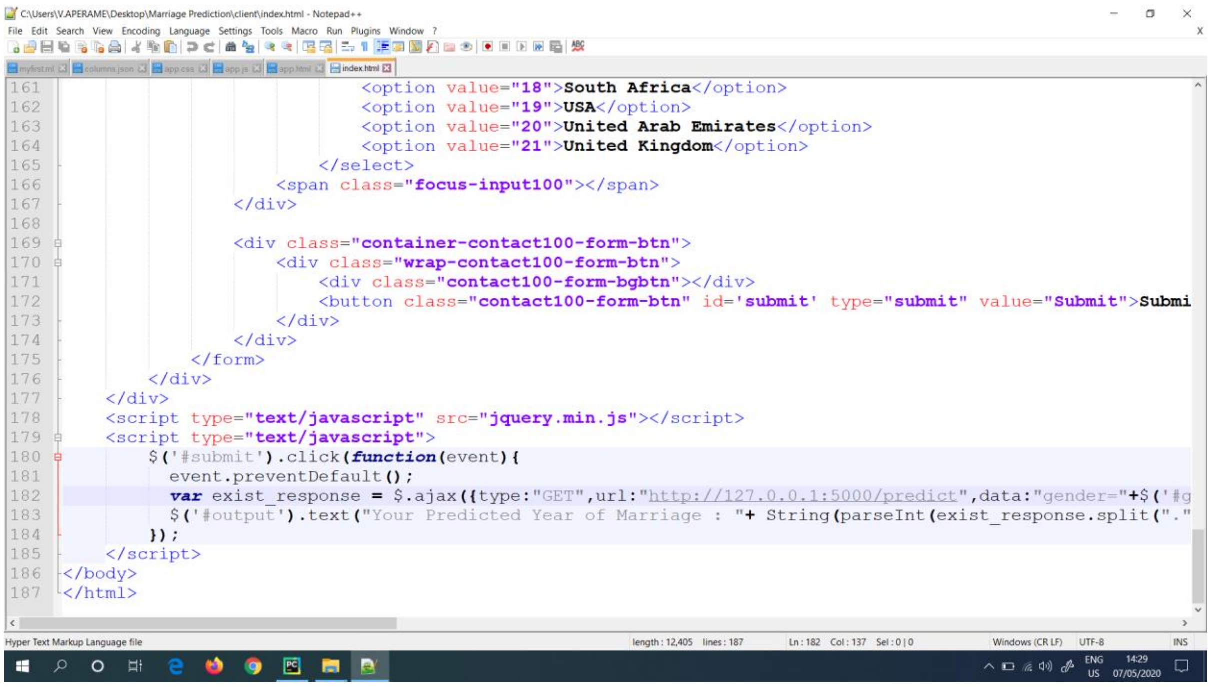
Caste

Select Caste

Mother Tongue

Figure 44: User Interface (UI) for the age prediction

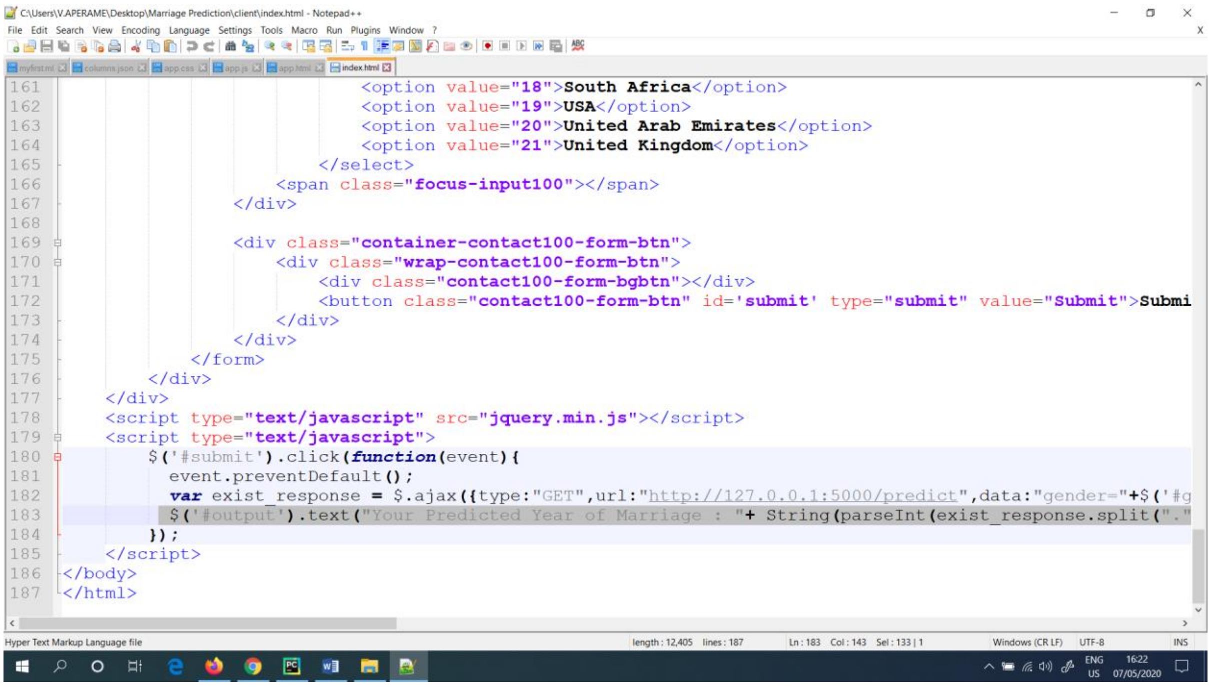
For the above created UI, the simple webpage was developed with the help of the HTML. I wrote the HTML, in Notepad++. In order to run the webpage in local host, I gave the web page address as http://127.0.0.1:5000/predict. Which is my local host. This is shown in (Figure 45).



```
                                    <option value="18">South Africa</option>
                                    <option value="19">USA</option>
                                    <option value="20">United Arab Emirates</option>
                                    <option value="21">United Kingdom</option>
                                </select>
                            <span class="focus-input100"></span>
                        </div>

                        <div class="container-contact100-form-btn">
                            <div class="wrap-contact100-form-btn">
                                <div class="contact100-form-bgbtn"></div>
                                <button class="contact100-form-btn" id='submit' type="submit" value="Submit">Submi
                            </div>
                        </div>
                    </form>
                </div>
            </div>
            <script type="text/javascript" src="jquery.min.js"></script>
            <script type="text/javascript">
                $('#submit').click(function(event){
                  event.preventDefault();
                  var exist_response = $.ajax({type:"GET",url:"http://127.0.0.1:5000/predict",data:"gender="+$('#g
                  $('#output').text("Your Predicted Year of Marriage : "+ String(parseInt(exist_response.split("."
                });
            </script>
        </body>
        </html>
```

Figure 45: HTML is used to create the web page

Rather than showing the age prediction, I thought it would be better when there is a prediction for year of marriage. Because, it will be more exciting. For this, I have taken the age prediction and the current age. And I added the current age with the predicted age, which helped to produce the actual year of marriage. This is shown in (Figure 46).


C:\Users\V.APERAME\Desktop\Marriage Prediction\client\index.html - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
myfirst.ml
columns.json
app.css
app.js
app.html
index.html
Hyper Text Markup Language file
length : 12,405 lines : 187
Ln : 183 Col : 143 Sel : 133 | 1
Windows (CR LF) UTF-8 INS
ENG US 16:22 07/05/2020

```
                            <option value="18">South Africa</option>
                            <option value="19">USA</option>
                            <option value="20">United Arab Emirates</option>
                            <option value="21">United Kingdom</option>
                        </select>
                    <span class="focus-input100"></span>
                </div>

                <div class="container-contact100-form-btn">
                    <div class="wrap-contact100-form-btn">
                        <div class="contact100-form-bgbtn"></div>
                        <button class="contact100-form-btn" id='submit' type="submit" value="Submit">Submi
                    </div>
                </div>
            </form>
        </div>
    </div>
    <script type="text/javascript" src="jquery.min.js"></script>
    <script type="text/javascript">
        $('#submit').click(function(event){
          event.preventDefault();
          var exist_response = $.ajax({type:"GET",url:"http://127.0.0.1:5000/predict",data:"gender="+$('#g
          $('#output').text("Your Predicted Year of Marriage : "+ String(parseInt(exist_response.split("."
        });
    </script>
</body>
</html>
```

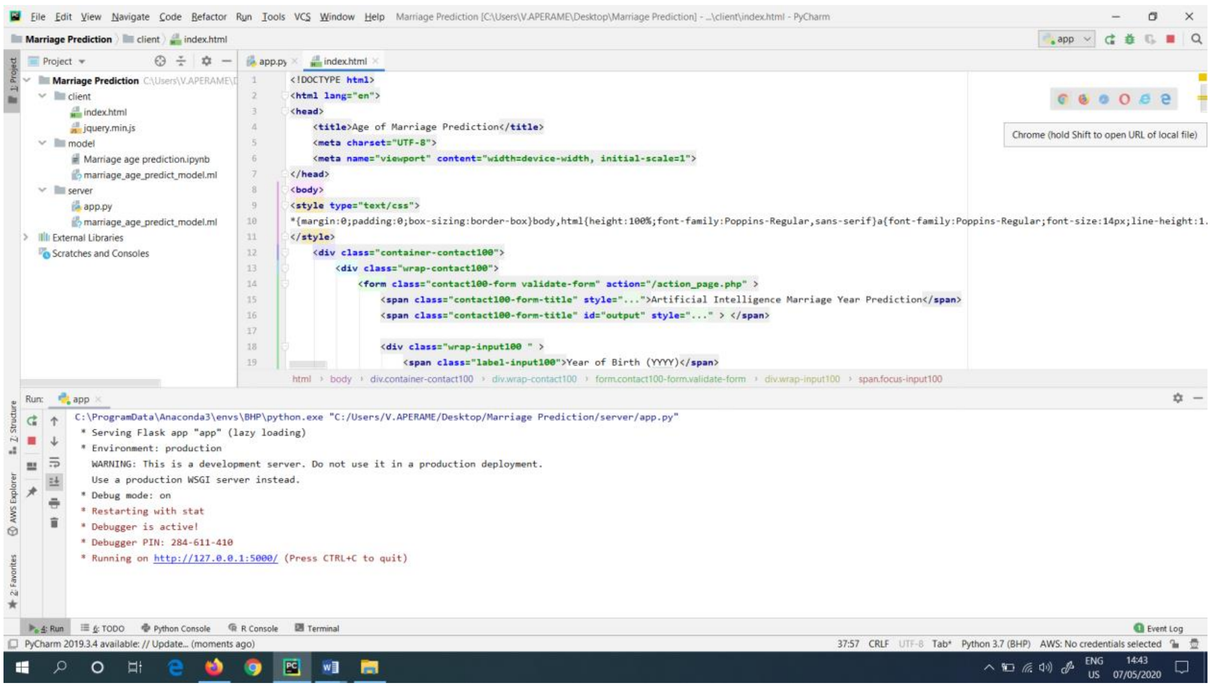
Figure 46: HTML code which helped to produce the actual year of marriage

All the Machine Learning model codes as well as the web page development codes are presented in the GitHub.

Finally, I tested my project in the Localhost. For this, in PyCharm first I ran the “app.py” in order to up the server. Next, I ran the “index.html” which I wrote in PyCharm earlier.

Rather than this, I also imported the Python Flask Server and CORS as well as I also installed the Python (All these steps are done before).

Finally, in order to run the website, I clicked the “Google Chrome” icon which is presented in the right hand corner side, in the “index.html”. (Figure 47) shows it.


File Edit View Navigate Code Refactor Run Tools VCS Window Help Marriage Prediction [C:\Users\V.APERAME\Desktop\Marriage Prediction] - ...\client\index.html - PyCharm
Marriage Prediction
client
index.html
Project
Marriage Prediction C:\Users\V.APERAME\D
client
index.html
jquery.min.js
model
Marriage age prediction.ipynb
marriage_age_predict_model.ml
server
app.py
marriage_age_predict_model.ml
External Libraries
Scratches and Consoles
app.py
index.html
Chrome (hold Shift to open URL of local file)

```
<!DOCTYPE html>
<html lang="en">
<head>
    <title>Age of Marriage Prediction</title>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
</head>
<body>
<style type="text/css">
*{margin:0;padding:0;box-sizing:border-box}body,html{height:100%;font-family:Poppins-Regular,sans-serif}a{font-family:Poppins-Regular;font-size:14px;line-height:1.
</style>
    <div class="container-contact100">
        <div class="wrap-contact100">
            <form class="contact100-form validate-form" action="/action_page.php" >
                <span class="contact100-form-title" style="...">Artificial Intelligence Marriage Year Prediction</span>
                <span class="contact100-form-title" id="output" style="..." > </span>

                <div class="wrap-input100 " >
                    <span class="label-input100">Year of Birth (YYYY)</span>
```

html › body › div.container-contact100 › div.wrap-contact100 › form.contact100-form.validate-form › div.wrap-input100 › span.focus-input100
Run: app

```
C:\ProgramData\Anaconda3\envs\BHP\python.exe "C:/Users/V.APERAME/Desktop/Marriage Prediction/server/app.py"
 * Serving Flask app "app" (lazy loading)
 * Environment: production
   WARNING: This is a development server. Do not use it in a production deployment.
   Use a production WSGI server instead.
 * Debug mode: on
 * Restarting with stat
 * Debugger is active!
 * Debugger PIN: 284-611-410
 * Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
```

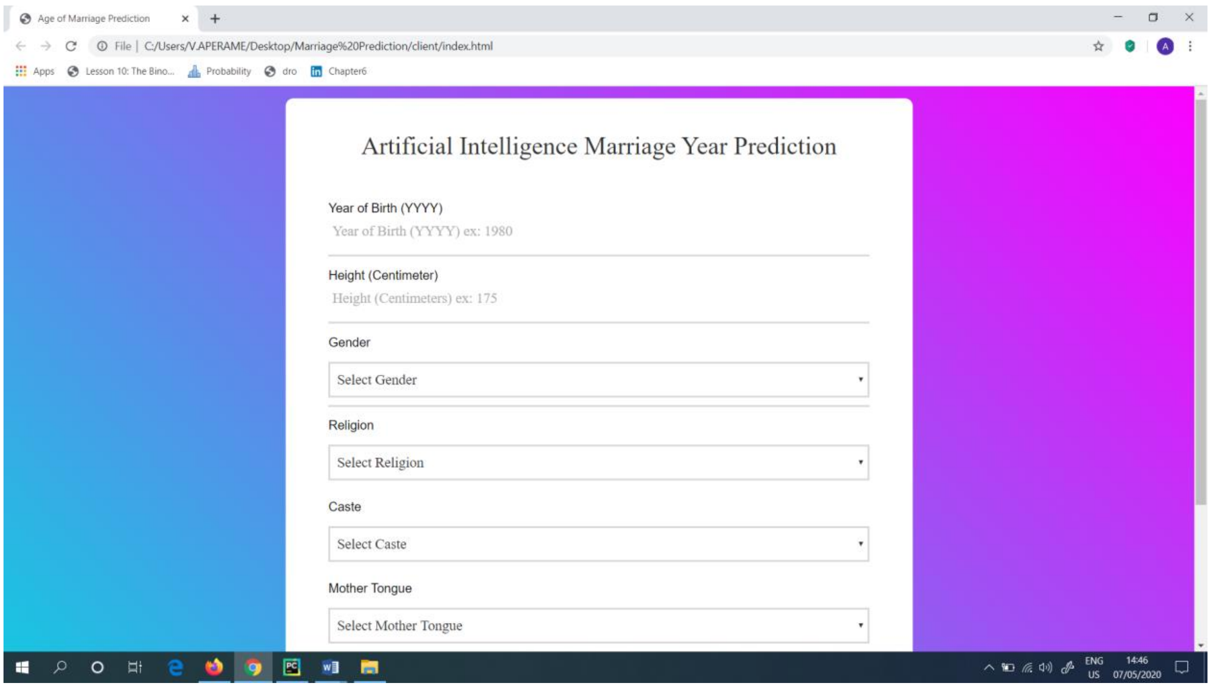
4: Run
6: TODO
Python Console
R Console
Terminal
Event Log
PyCharm 2019.3.4 available: // Update... (moments ago)
37:57 CRLF UTF-8 Tab* Python 3.7 (BHP) AWS: No credentials selected
ENG US 14:43 07/05/2020

Figure 47: Clicked the Google Chrome icon in “index.html” file

OR

There is another method (Second Method) presented, in order to run the project. This is explained below.

Directly I went to the “index.html” file, which I saved it in Desktop, and I double clicked it in the file. It also prompted me the similar website which I got as shown below. This is shown in (Figure 48). This is directly hosted in the local host of the PC (Personal Computer).


Age of Marriage Prediction
File | C:/Users/V.APERAME/Desktop/Marriage%20Prediction/client/index.html
Apps
Lesson 10: The Bino...
Probability
dro
Chapter6

Artificial Intelligence Marriage Year Prediction

Year of Birth (YYYY)
Year of Birth (YYYY) ex: 1980

Height (Centimeter)
Height (Centimeters) ex: 175

Gender
Select Gender

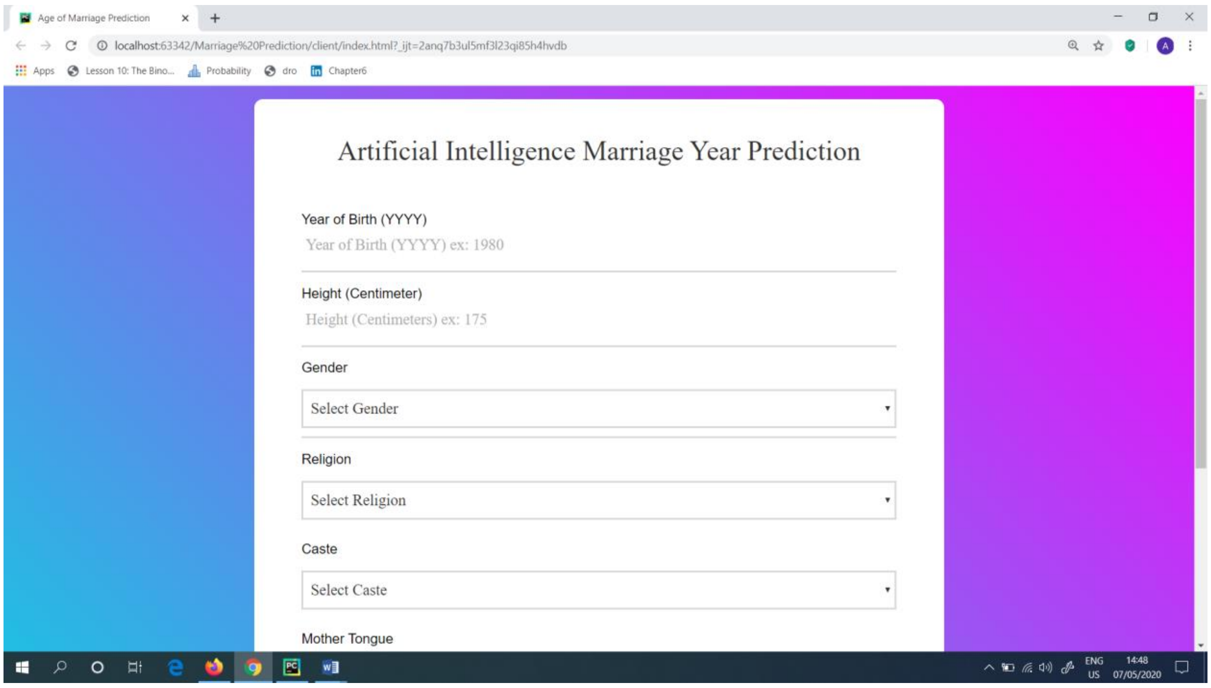
Religion
Select Religion

Caste
Select Caste

Mother Tongue
Select Mother Tongue

Figure 48: Got the same website by double clicking the “index.html” file

Next, when I continued from the first method which I showed above, suddenly, in the Local host of PyCharm, the website which I created was get hosted as shown in (Figure 49). This is hosted with the help of the PyCharm.


Age of Marriage Prediction
localhost:63342/Marriage%20Prediction/client/index.html?_ijt=2anq7b3ul5mf3l23qi85h4hvdb
Apps
Lesson 10: The Bino...
Probability
dro
Chapter6
ENG US
14:48
07/05/2020

Artificial Intelligence Marriage Year Prediction

Year of Birth (YYYY)
Year of Birth (YYYY) ex: 1980

Height (Centimeter)
Height (Centimeters) ex: 175

Gender
Select Gender

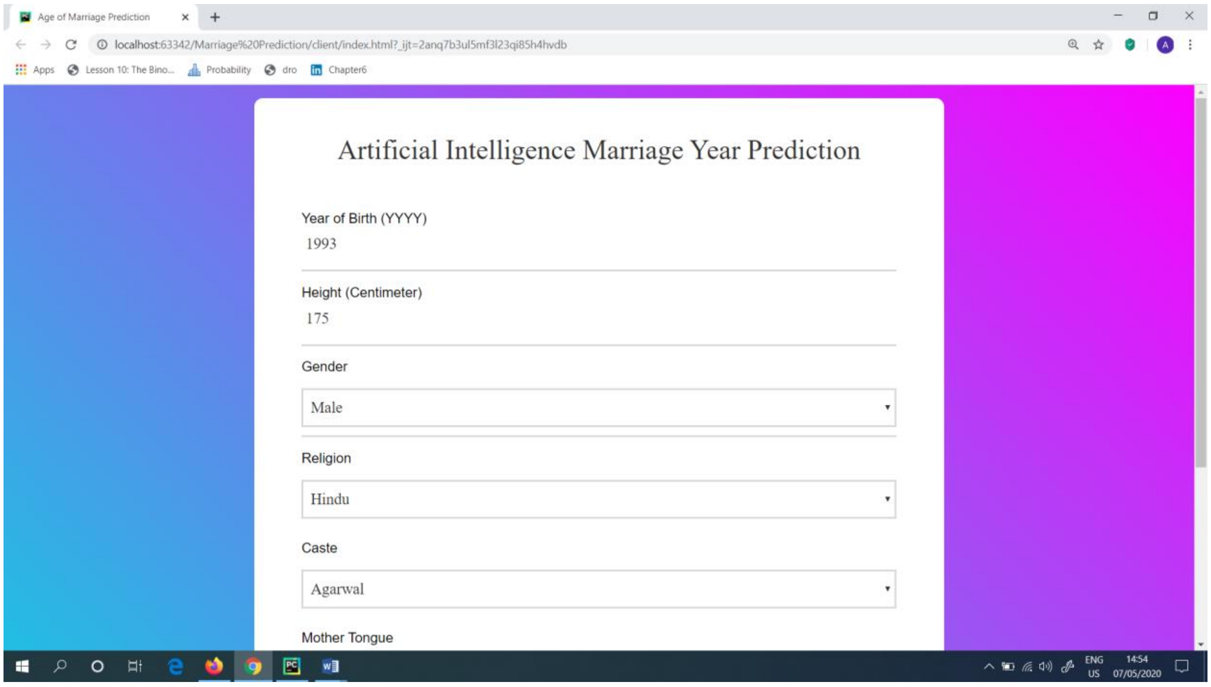
Religion
Select Religion

Caste
Select Caste

Mother Tongue

Figure 49: Website hosted in the localhost (PyCharm)

Then, I thought to check the model, whether it is working fine or not. So, as the input for the website I gave the following as shown in (Figure 50). I.e.: Year of Birth – 1993, Height (cm) – 175, Gender – Male, Religion – Hindu, Caste – Agarwal, Mother Tongue – Telugu and Country – India.


Age of Marriage Prediction
localhost:63342/Marriage%20Prediction/client/index.html?_ijt=2anq7b3ul5mf3l23qi85h4hvdb
Apps
Lesson 10: The Bino...
Probability
dro
Chapter6
ENG US
14:54
07/05/2020

Artificial Intelligence Marriage Year Prediction

Year of Birth (YYYY)
1993

Height (Centimeter)
175

Gender
Male

Religion
Hindu

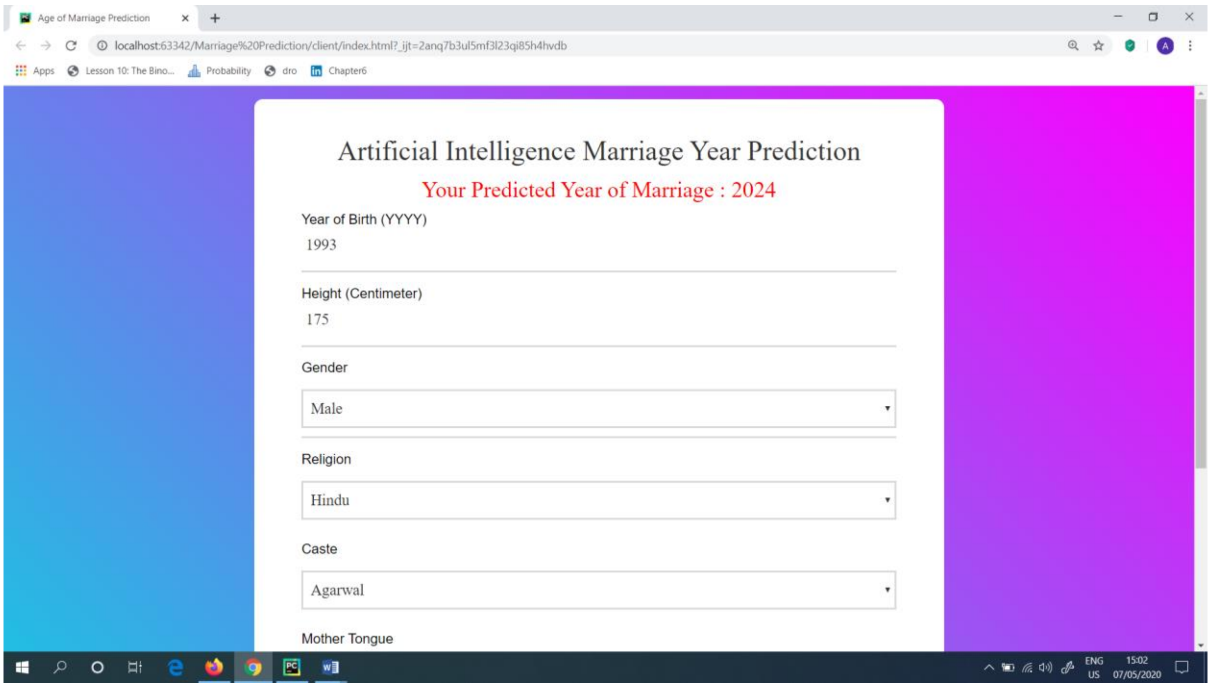
Caste
Agarwal

Mother Tongue

Figure 50: Checked the model with inputs given

In the Artificial Intelligence Marriage Year Prediction system, the Year of Birth is important in order to identify the predicted year of marriage. That is the reason the Year of Birth is also considered in this project.

The output, which I gained when I ran this project, with the below mentioned inputs is shown in (Figure 51). Next, I clicked the "Submit" button which is presented in the webpage to get the output.


Artificial Intelligence Marriage Year Prediction

Your Predicted Year of Marriage : 2024

Year of Birth (YYYY)
1993

Height (Centimeter)
175

Gender
Male

Religion
Hindu

Caste
Agarwal

Mother Tongue

Age of Marriage Prediction
localhost:63342/Marriage%20Prediction/client/index.html?_ijt=2anq7b3ul5mf3l23qi85h4hvdb
Apps
Lesson 10: The Bino...
Probability
dro
Chapter6
ENG US
15:02
07/05/2020

Figure 51: Output gained

Next, I also tried the Predicted Year of Marriage, by giving different inputs. The model which I developed worked fine without any errors. Also the Predicted Year of Marriage got varied according to the user inputs which were given as in the webpage.

Steps Followed (Summary)

1. Got the data and cleaned it
2. Trained the model and evaluated it
3. Exported the trained model
4. Created a flask API
5. Connected the flask API to the trained model
6. Take the URL arguments and pass it, until it gives the results
7. Filled the HTML file, and when clicked the "submit" button, it goes back to API and it predicts from, by loading the exported trained model and it showed the final result (Predicted Year of Marriage).

Advantages of the product produced

- This above produced Machine Learning model can be used for Indians who are surviving in India and out of India (Indians who are living in foreign countries). Because the data set which is used in this project is only collected from Indians.

- As it is predicting the Year of Marriage, it will be helpful in order to make the arrangements prior to the wedding.
- When there are data sets produced for Other countries (Like Sri Lanka, United Kingdom, United States etc.) then there also, the Year of Marriage can be easily predicted with the help of the produced Machine Learning model.
- This Marriage Year Prediction model will be very useful to the marriage brokers. Because, when this project is hosted in the webpage, and when it is connected with the marriage brokers, then they will be easily notified about the Marriage Year of Prediction of every individual.
- This Machine Learning model can be further developed, when the data sets are collected all over the world. And, the Marriage Year Prediction can be further developed from the gained data sets to all over the world.

References

- https://www.jenunderwood.com/2019/02/10/predict-your-relationship-future/
- https://www.kaggle.com/c/marriage-prediction
- https://github.com/ashokveda/ML_deployment_Flask_AWS_marriage_age_prediction/
- https://www.youtube.com/watch?v=5mDYijMfSzs&t=620s
- https://www.youtube.com/watch?v=SZUNUB6nz3g