K L DEEMED TO BE UNIVERSITY

MACHINE LEARNING

19CS3021A

SKILL PROJECT- 01

PROJECT TEAM MEMBERS

190030909 – Deepak k

190030921 – Praveen k

190030386 – Kowshik D

PROJECT

Person's Life Expectancy

The term "life expectancy" refers to the number of years a person can expect to live. By definition, life expectancy is based on an estimate of the average age that members of a particular population group will be when they die.

An important point to bear in mind when interpreting life expectancy estimates is that very few people will die at precisely the age indicated by life expectancy, even if mortality patterns stay constant.

For example, very few of the infants born in South Africa in 2009 will die at 52.2 years of age, as per the figures in the map above. Most will die much earlier or much later, since the risk of death is not uniform across the lifetime. Life expectancy is the average.

In societies with high infant mortality rates many people die in the first few years of life; but once they survive childhood, people often live much longer. Indeed, this is a common source of confusion in the interpretation of life expectancy figures: It is perfectly possible that a given population has a low life expectancy at birth, and yet has a large proportion of old people.

ABSTRACT

Our goal is to Estimating the life expectancy entails predicting the probability of surviving successive years of life based on Education, number of infant deaths, alcohol consumption, and adult

Classifying countries based on income levels, we can see how economics impact health: a higher income is associated with a longer life expectancy, continuously between 1960 and 2015. High income countries see a steady, but slower, increase in life expectancy; middle income countries experienced large growth in the 1960s and saw a corresponding up-tick in life expectancy. In lowincome countries, life expectancy gains all but stopped between mid-80s — mid-90s, followed by a strong increase in life expectancy in recent years.

INDEX

Introduction

- Life expectancy is the average number of years a person in a population could expect to live after age x. It is the life table parameter most commonly used to compare the survival experience of populations. The age most often selected to make comparisons is 0.0 (i.e., birth), although, for many substantive and policy analyses, other ages such as 65+ and 85+ are more relevant and may be used (e.g., for determining person-years of Medicare and Social Security benefit entitlement)

  Humans live longer now. In fact, life expectancy and the maximum observed age at death have increased significantly in recent decades. Up to now, it is not practical, useful or ethical to prolong a healthy life by simply changing human genes or limiting food intake. The maximum theoretical life span in humans remains the subject of considerable debate and the extension of life is one of the great challenges of the 21stcentury. Many scientists believe that human life has an intrinsic upper limit, although they do not agree that it is 85, 100 or 150. The maximum human life span is generally postulated at around 125 years, while the record of older age is that of death growing today. The conventional analysis or theoretical model has not yet presented a plausible explanation for this disagreement.

Software Requirements:

- Anaconda (Jupyter Notebook)/ Google Colab.
- Browser

Some Python Modules:

- Numpy
- Pandas
- Matplot lib
- Seaborn
- Sklearn

Hardware Requirements:

- OS: Windows 8/8.1/10 (64-bit)
- Screen resolution of at least 800x600
- Ram of at least 1GB.

Requirement Analysis

The required data was collected from WHO and United Nations website

Finalizing Requirements

The final merged file(final dataset) consists of 22 Columns and 2938 rows which meant 20 predicting variables. All predicting variables was then divided into several broad categories:Immunization related factors, Mortality factors, Economical factors and Social factors.

Problem Solving Model

Steps Involved in Problem Solving

- **Defining the Problem**

  -> Describing the initial state of a problem that's to be solved.

- **Selecting Best Strategy**

  ->Regression Model to predict life expectancy according to the requirements during test window.

- **Implement Strategy**

  -> we'll use Random Forest Regressor with Random and Grid search CV. Using Random Search CV, we'll select the expected neighbourhood of top parameters for the model

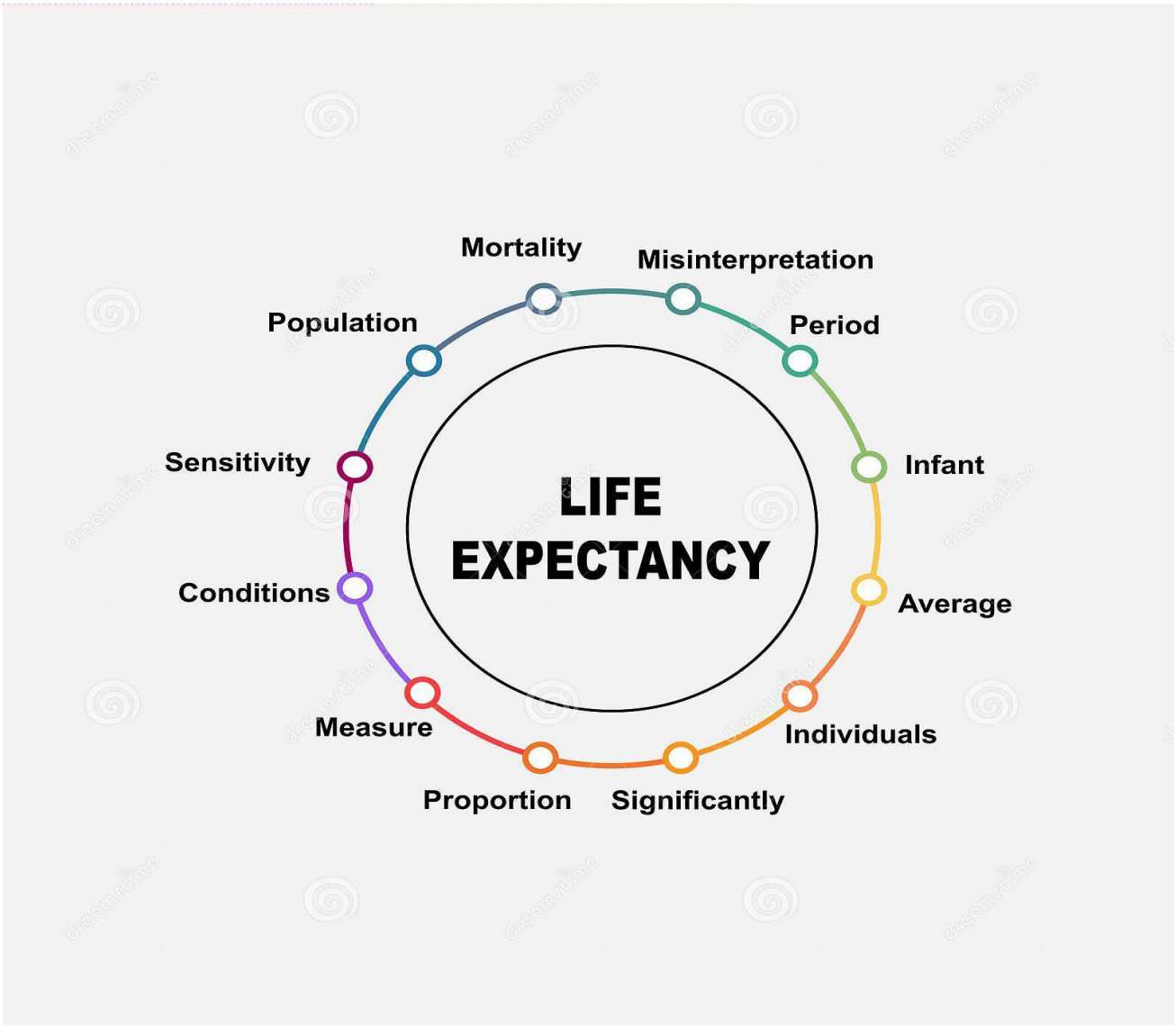
- **Evaluate Results**

  -> Evaluating the Output with the desired output.

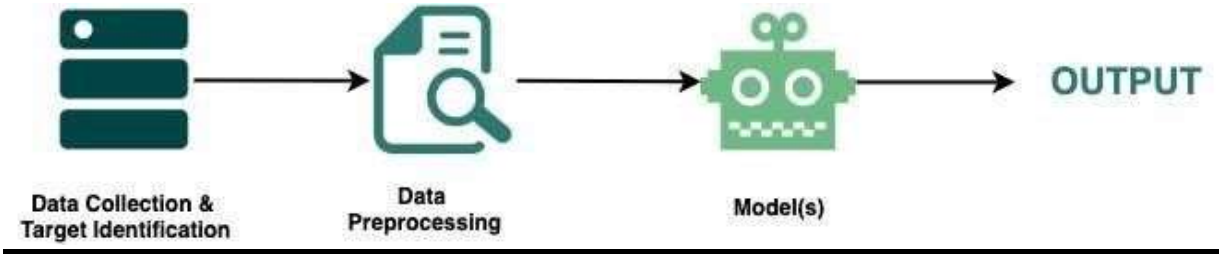
- **Implement Appropriate Changes in the Process** -> Verify any changes are required.

- **Continuous Improvement**

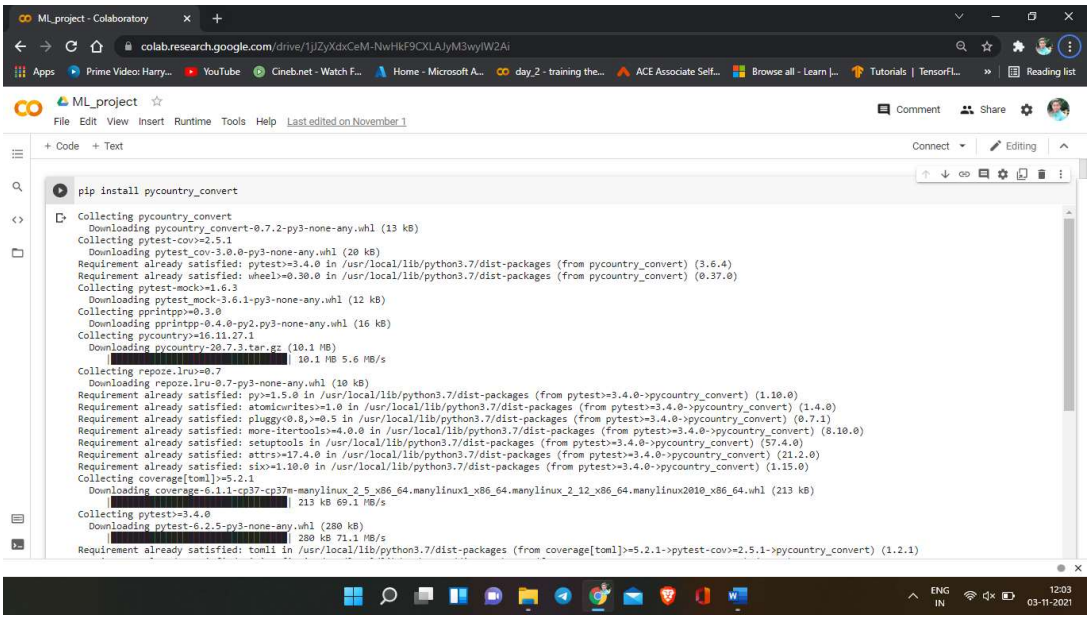
Modeling-Data flow diagram

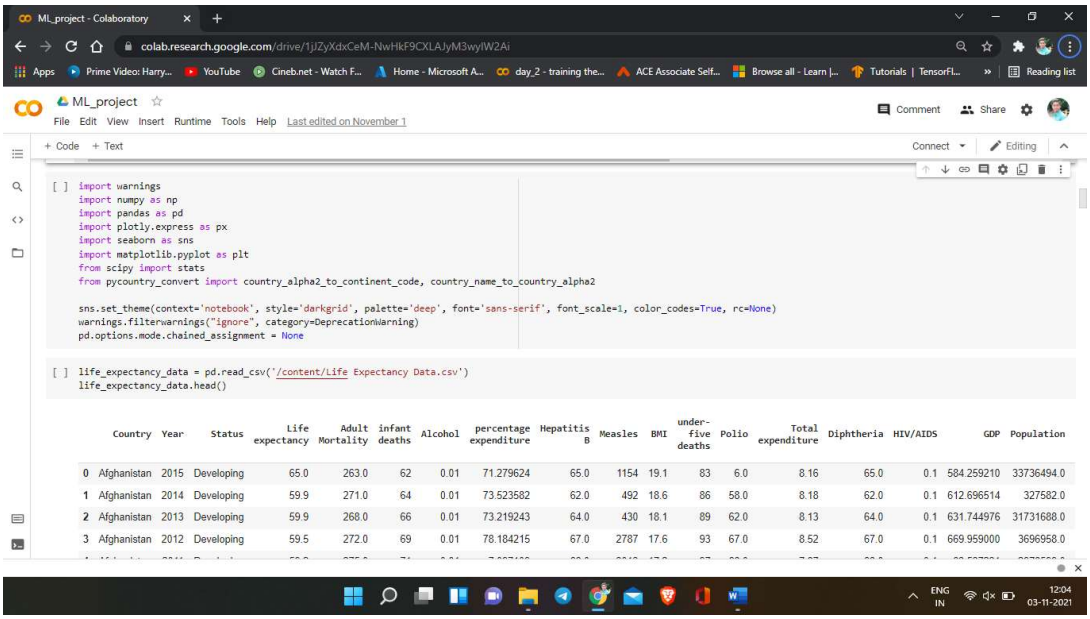


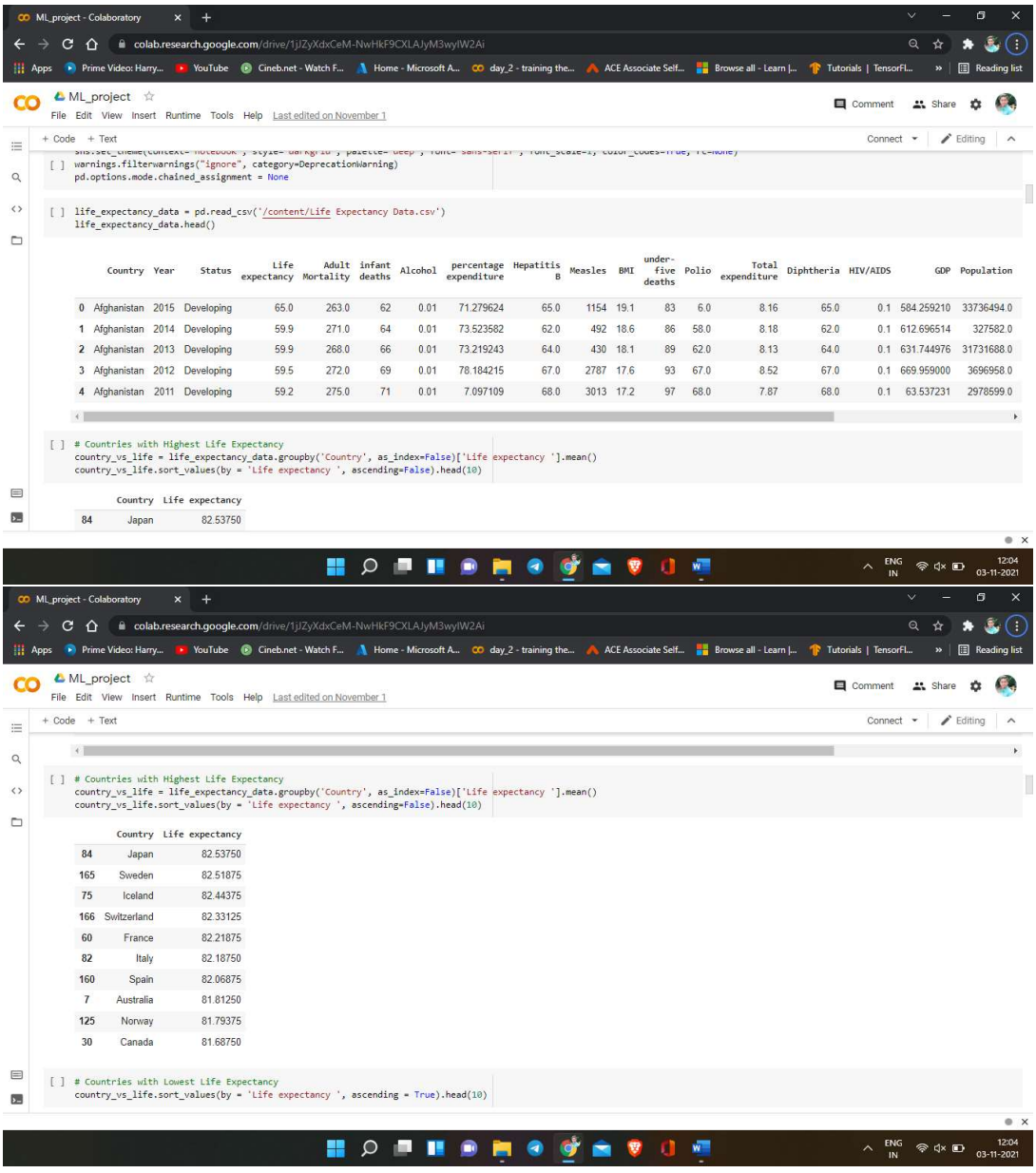
System Design



Implementation Output Screenshots along with System test and evaluation







ML_project - Colaboratory

colab.research.google.com/drive/1jIZyXdxCeM-NwHkF9CXLAJyM3wyIW2Ai

ML_project

File Edit View Insert Runtime Tools Help Last edited on November 1

Comment Share Connect Editing

+ Code + Text

```
[ ] warnings.filterwarnings("ignore", category=DeprecationWarning)
    pd.options.mode.chained_assignment = None
```

```
[ ] life_expectancy_data = pd.read_csv('/content/Life Expectancy Data.csv')
    life_expectancy_data.head()
```

	Country	Year	Status	Life expectancy	Adult Mortality	infant deaths	Alcohol	percentage expenditure	Hepatitis B	Measles	BMI	under-five deaths	Polio	Total expenditure	Diphtheria	HIV/AIDS	GDP	Population
0	Afghanistan	2015	Developing	65.0	263.0	62	0.01	71.279624	65.0	1154	19.1	83	6.0	8.16	65.0	0.1	584.259210	33736494.0
1	Afghanistan	2014	Developing	59.9	271.0	64	0.01	73.523582	62.0	492	18.6	86	58.0	8.18	62.0	0.1	612.696514	327582.0
2	Afghanistan	2013	Developing	59.9	268.0	66	0.01	73.219243	64.0	430	18.1	89	62.0	8.13	64.0	0.1	631.744976	31731688.0
3	Afghanistan	2012	Developing	59.5	272.0	69	0.01	78.184215	67.0	2787	17.6	93	67.0	8.52	67.0	0.1	669.959000	3696958.0
4	Afghanistan	2011	Developing	59.2	275.0	71	0.01	7.097109	68.0	3013	17.2	97	68.0	7.87	68.0	0.1	63.537231	2978599.0

```
[ ] # Countries with Highest Life Expectancy
    country_vs_life = life_expectancy_data.groupby('Country', as_index=False)['Life expectancy '].mean()
    country_vs_life.sort_values(by = 'Life expectancy ', ascending=False).head(10)
```

	Country	Life expectancy
84	Japan	82.53750

ENG IN 12:04 03-11-2021

ML_project - Colaboratory

colab.research.google.com/drive/1jIZyXdxCeM-NwHkF9CXLAJyM3wyIW2Ai

ML_project

File Edit View Insert Runtime Tools Help Last edited on November 1

Comment Share Connect Editing

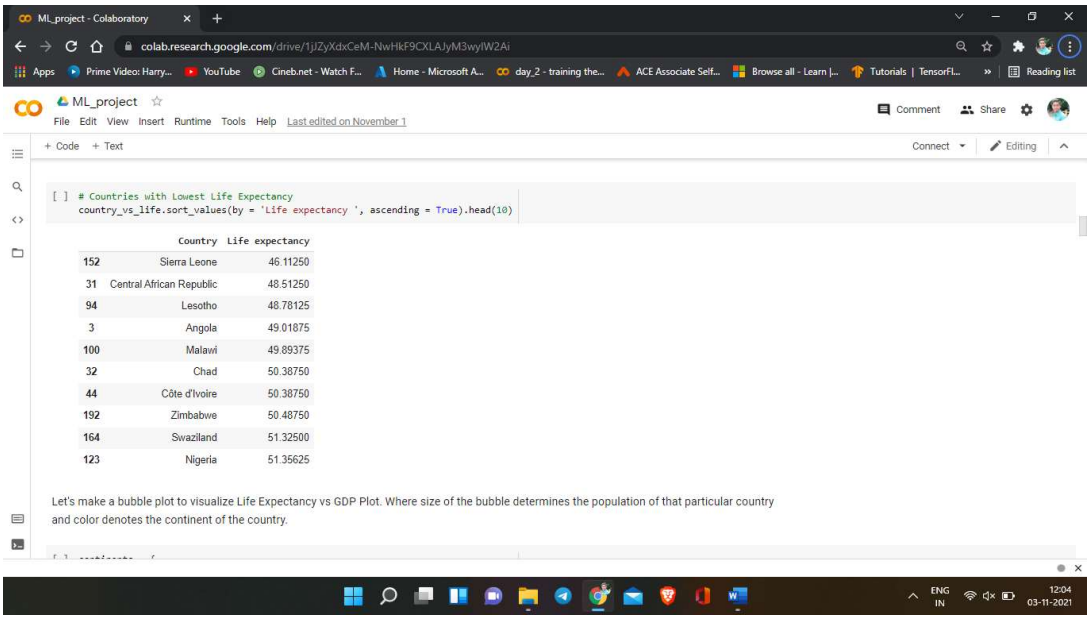
+ Code + Text

```
[ ] # Countries with Highest Life Expectancy
    country_vs_life = life_expectancy_data.groupby('Country', as_index=False)['Life expectancy '].mean()
    country_vs_life.sort_values(by = 'Life expectancy ', ascending=False).head(10)
```

	Country	Life expectancy
84	Japan	82.53750
165	Sweden	82.51875
75	Iceland	82.44375
166	Switzerland	82.33125
60	France	82.21875
82	Italy	82.18750
160	Spain	82.06875
7	Australia	81.81250
125	Norway	81.79375
30	Canada	81.68750

```
[ ] # Countries with Lowest Life Expectancy
    country_vs_life.sort_values(by = 'Life expectancy ', ascending = True).head(10)
```

ENG IN 12:04 03-11-2021



ML_project - Colaboratory

colab.research.google.com/drive/1jIZyXdxCeM-NwHkF9CXLAJyM3wyIW2Ai

ML_project

File Edit View Insert Runtime Tools Help Last edited on November 1

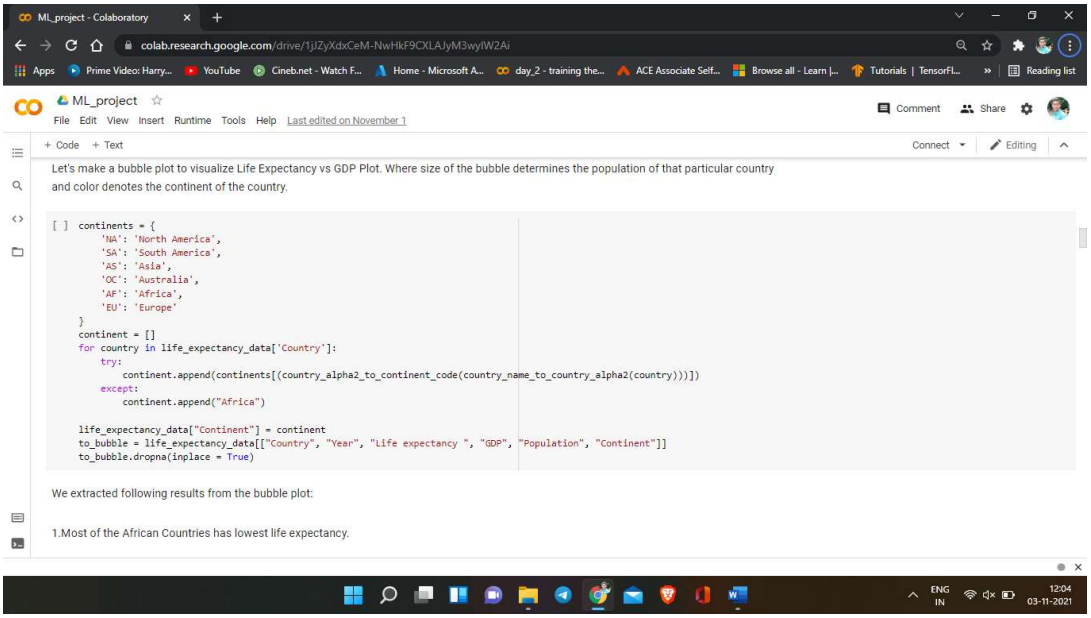
Comment Share Connect Editing

+ Code + Text

```
[ ] # Countries with Lowest Life Expectancy
    country_vs_life.sort_values(by = 'Life expectancy ', ascending = True).head(10)
```

	Country	Life expectancy
152	Sierra Leone	46.11250
31	Central African Republic	48.51250
94	Lesotho	48.78125
3	Angola	49.01875
100	Malawi	49.89375
32	Chad	50.38750
44	Côte d'Ivoire	50.38750
192	Zimbabwe	50.48750
164	Swaziland	51.32500
123	Nigeria	51.35625

Let's make a bubble plot to visualize Life Expectancy vs GDP Plot. Where size of the bubble determines the population of that particular country and color denotes the continent of the country.



ML_project - Colaboratory

colab.research.google.com/drive/1jIZyXdxCeM-NwHkF9CXLAJyM3wyIW2Ai

ML_project

File Edit View Insert Runtime Tools Help Last edited on November 1

Comment Share Connect Editing

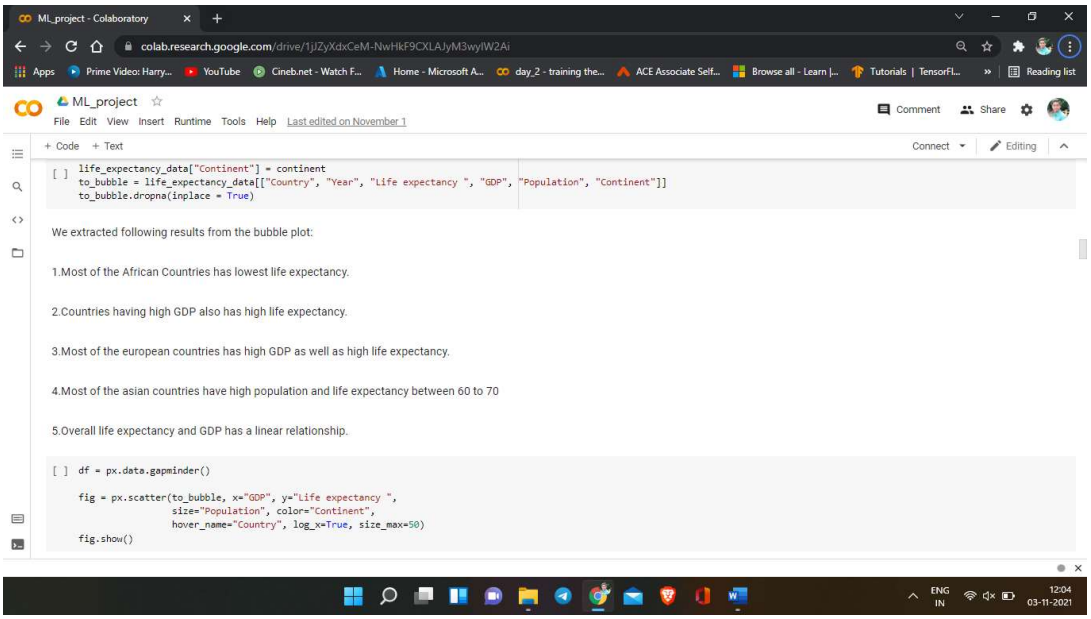
+ Code + Text

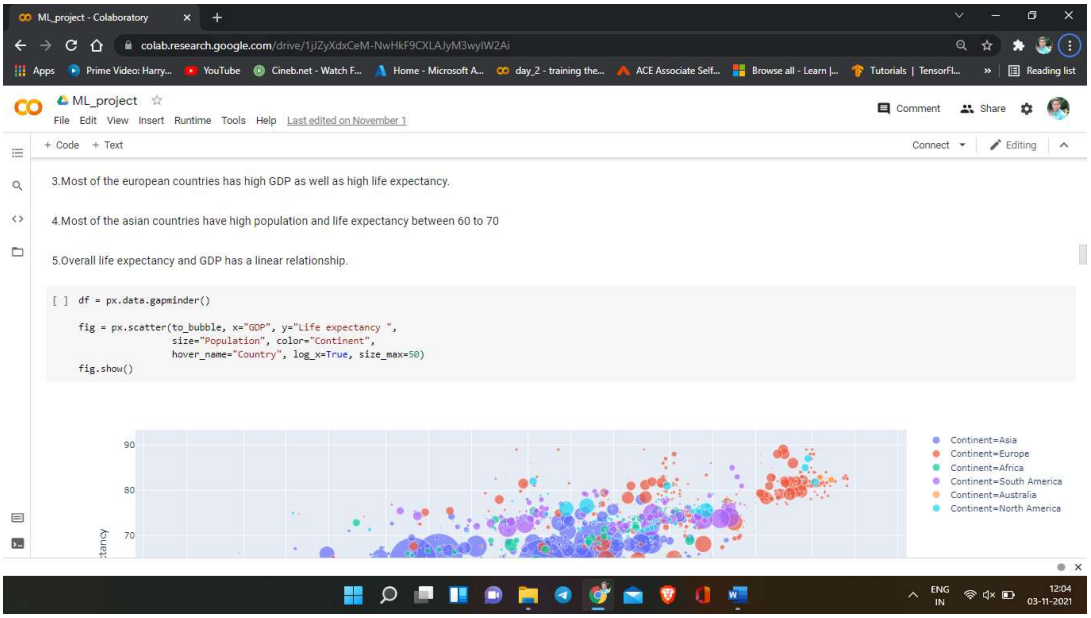
Let's make a bubble plot to visualize Life Expectancy vs GDP Plot. Where size of the bubble determines the population of that particular country and color denotes the continent of the country.

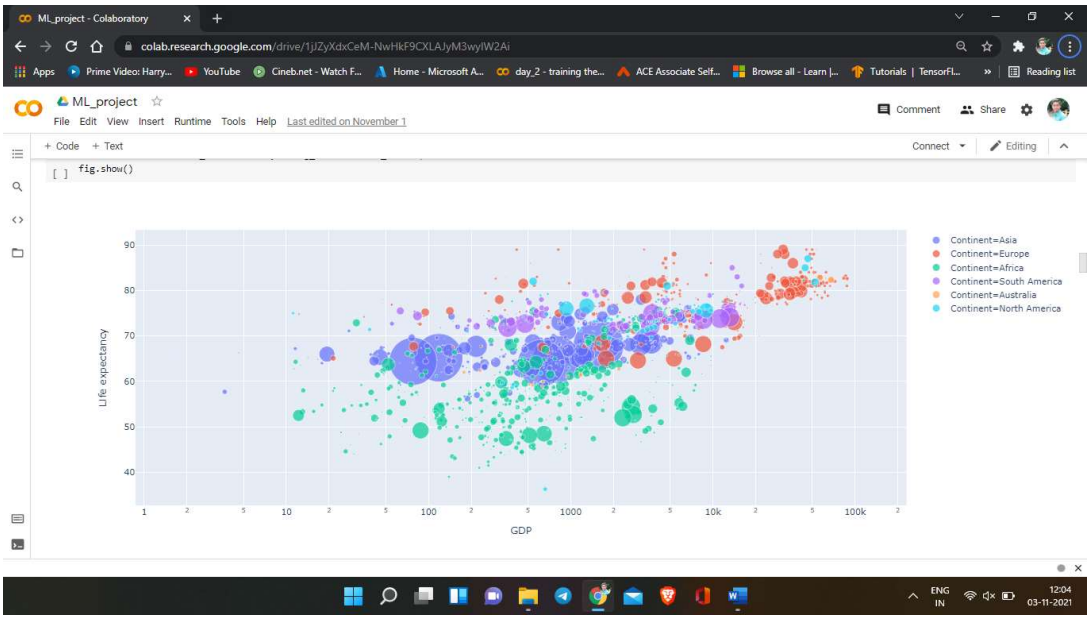
```
[ ] continents = {
        'NA': 'North America',
        'SA': 'South America',
        'AS': 'Asia',
        'OC': 'Australia',
        'AF': 'Africa',
        'EU': 'Europe'
    }
    continent = []
    for country in life_expectancy_data['Country']:
        try:
            continent.append(continents[(country_alpha2_to_continent_code(country_name_to_country_alpha2(country)))])
        except:
            continent.append("Africa")

    life_expectancy_data["Continent"] = continent
    to_bubble = life_expectancy_data[["Country", "Year", "Life expectancy ", "GDP", "Population", "Continent"]]
    to_bubble.dropna(inplace = True)
```

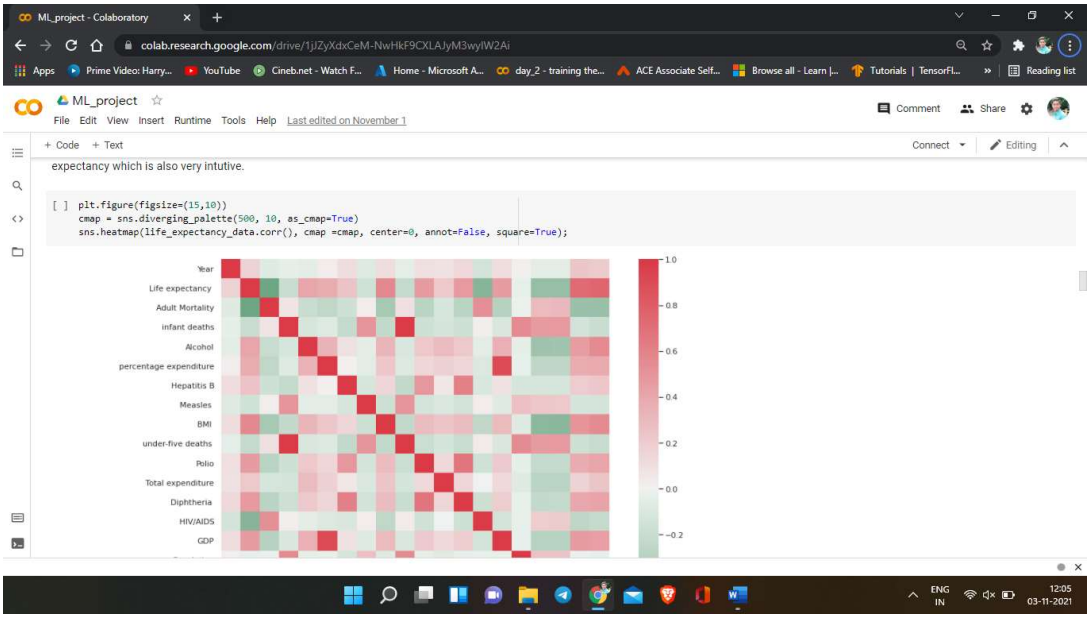
We extracted following results from the bubble plot:

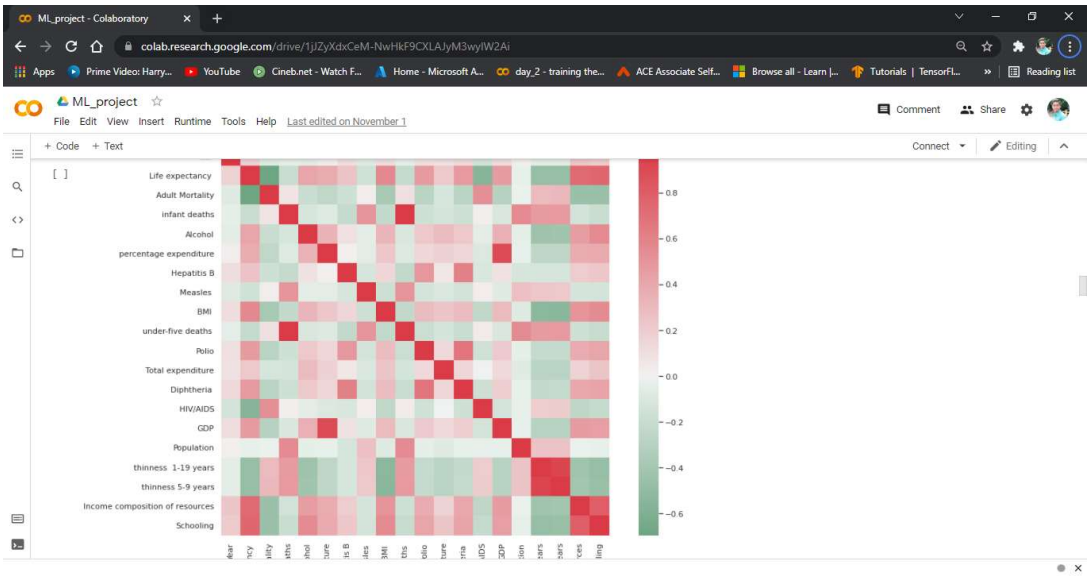
1.Most of the African Countries has lowest life expectancy.

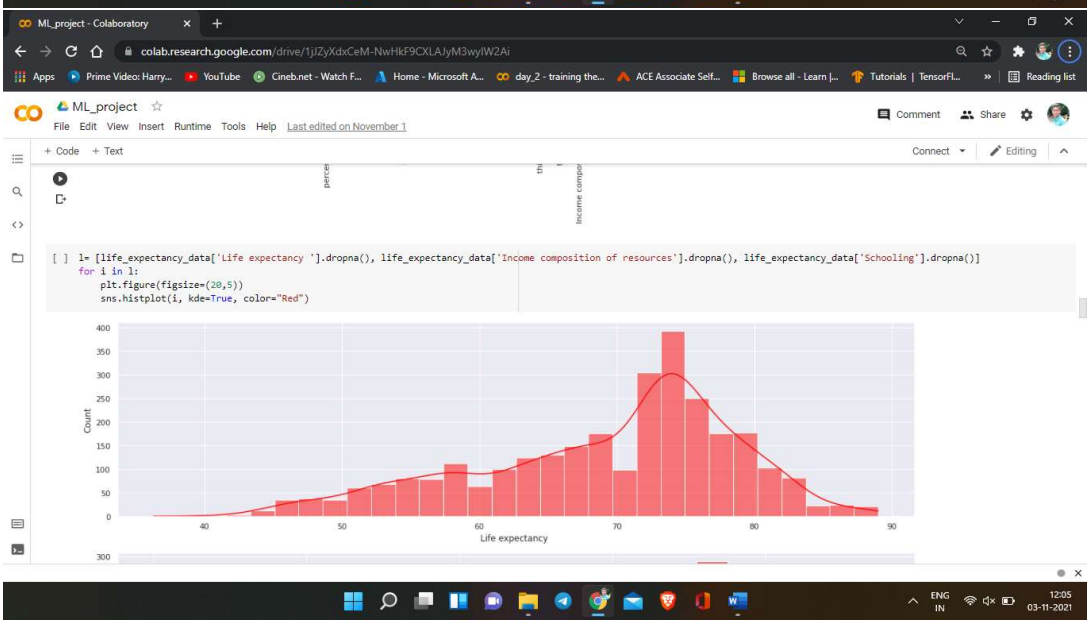

ML_project - Colaboratory
colab.research.google.com/drive/1jIZyXdxCeM-NwHkF9CXLAJyM3wyIW2Ai
ML_project
File Edit View Insert Runtime Tools Help Last edited on November 1
Comment Share
+ Code + Text
Connect Editing
[ ] life_expectancy_data["Continent"] = continent
to_bubble = life_expectancy_data[["Country", "Year", "Life expectancy ", "GDP", "Population", "Continent"]]
to_bubble.dropna(inplace = True)
We extracted following results from the bubble plot:
1.Most of the African Countries has lowest life expectancy.
2.Countries having high GDP also has high life expectancy.
3.Most of the european countries has high GDP as well as high life expectancy.
4.Most of the asian countries have high population and life expectancy between 60 to 70
5.Overall life expectancy and GDP has a linear relationship.
[ ] df = px.data.gapminder()
fig = px.scatter(to_bubble, x="GDP", y="Life expectancy ",
size="Population", color="Continent",
hover_name="Country", log_x=True, size_max=50)
fig.show()

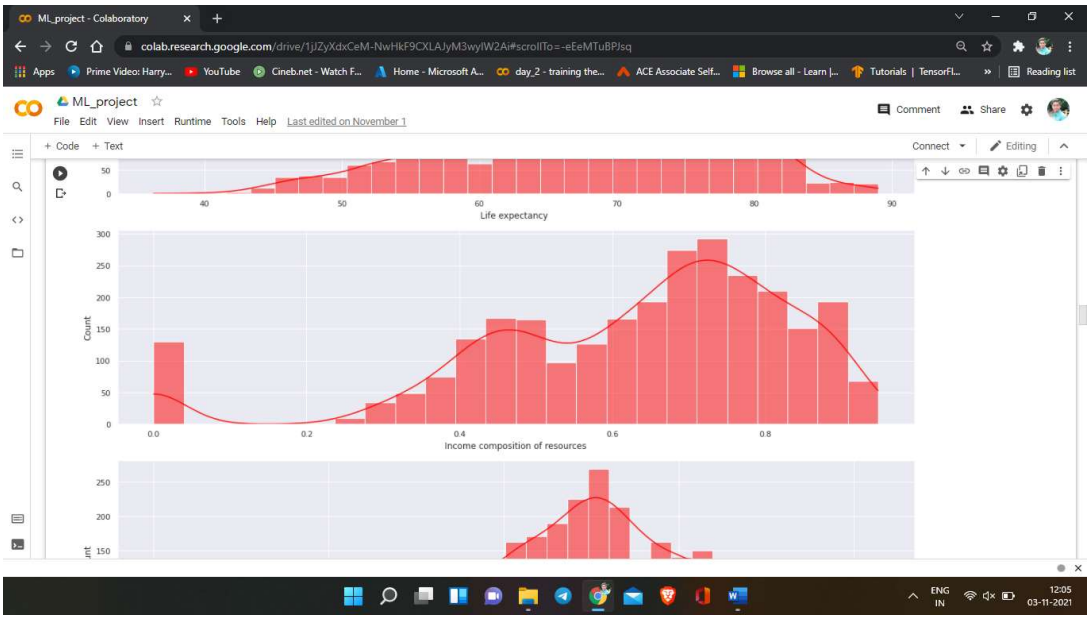

ML_project - Colaboratory
colab.research.google.com/drive/1jIZyXdxCeM-NwHkF9CXLAJyM3wyIW2Ai
ML_project
File Edit View Insert Runtime Tools Help Last edited on November 1
Comment Share
+ Code + Text
Connect Editing
3.Most of the european countries has high GDP as well as high life expectancy.
4.Most of the asian countries have high population and life expectancy between 60 to 70
5.Overall life expectancy and GDP has a linear relationship.
[ ] df = px.data.gapminder()
fig = px.scatter(to_bubble, x="GDP", y="Life expectancy ",
size="Population", color="Continent",
hover_name="Country", log_x=True, size_max=50)
fig.show()
90
80
70
Continent=Asia
Continent=Europe
Continent=Africa
Continent=South America
Continent=Australia
Continent=North America

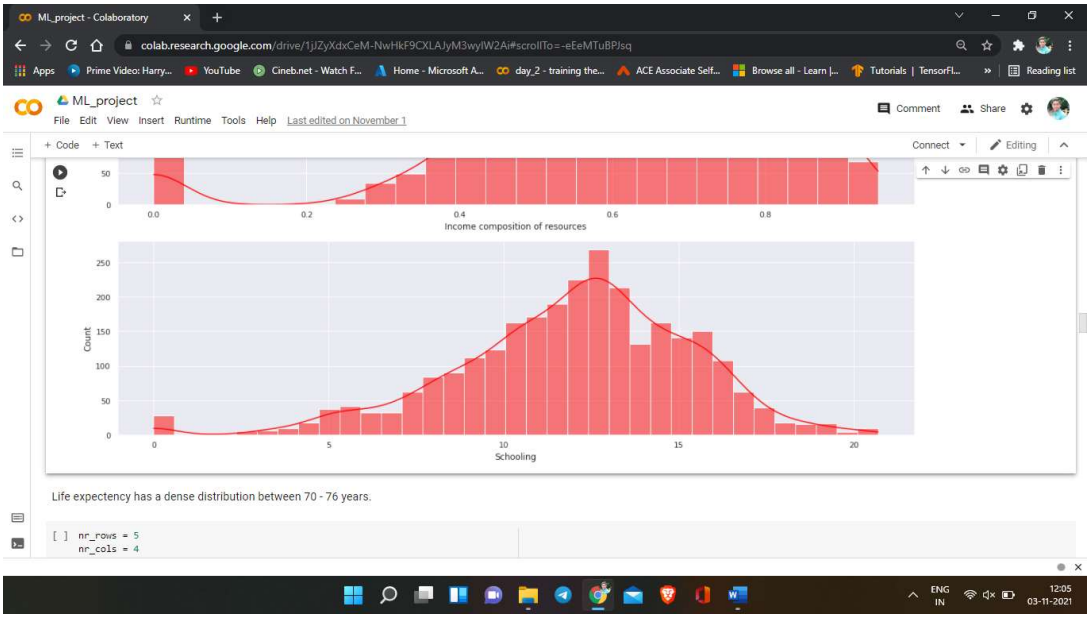

ML_project - Colaboratory
colab.research.google.com/drive/1jIZyXdxCeM-NwHkF9CXLAJyM3wyIW2Ai
ML_project
File Edit View Insert Runtime Tools Help Last edited on November 1
Comment Share
+ Code + Text
Connect Editing
[ ] fig.show()
Life expectancy
GDP
Continent=Asia
Continent=Europe
Continent=Africa
Continent=South America
Continent=Australia
Continent=North America

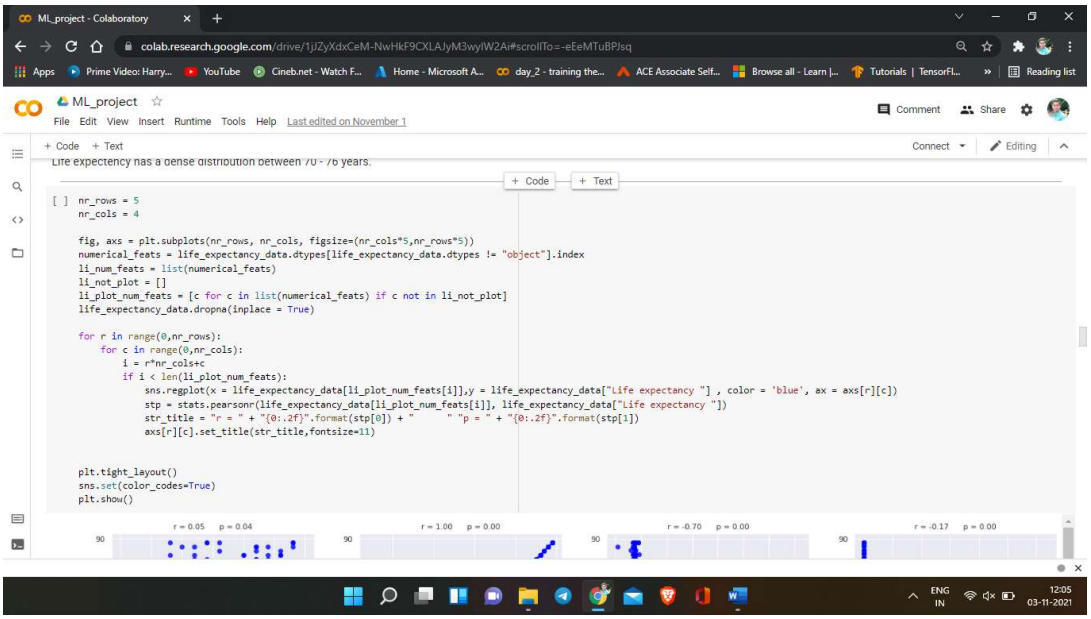

ML_project - Colaboratory
colab.research.google.com/drive/1jIZyXdxCeM-NwHkF9CXLAJyM3wyIW2Ai
ML_project
File Edit View Insert Runtime Tools Help Last edited on November 1
Comment Share
+ Code + Text
Connect Editing
expectancy which is also very intutive.
[ ] plt.figure(figsize=(15,10))
cmap = sns.diverging_palette(500, 10, as_cmap=True)
sns.heatmap(life_expectancy_data.corr(), cmap =cmap, center=0, annot=False, square=True);
Year
Life expectancy
Adult Mortality
infant deaths
Alcohol
percentage expenditure
Hepatitis B
Measles
BMI
under-five deaths
Polio
Total expenditure
Diphtheria
HIV/AIDS
GDP

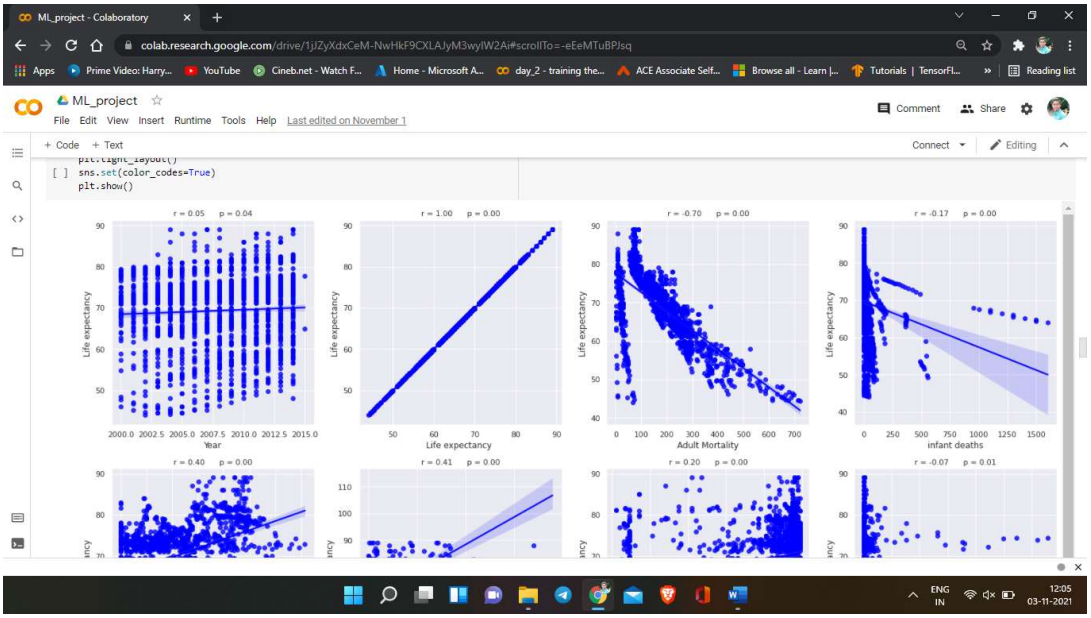

ML_project - Colaboratory
colab.research.google.com/drive/1jIZyXdxCeM-NwHkF9CXLAJyM3wyIW2Ai
ML_project
File Edit View Insert Runtime Tools Help Last edited on November 1
Comment Share
+ Code + Text
Connect Editing
[ ]
Life expectancy
Adult Mortality
infant deaths
Alcohol
percentage expenditure
Hepatitis B
Measles
BMI
under-five deaths
Polio
Total expenditure
Diphtheria
HIV/AIDS
GDP
Population
thinness 1-19 years
thinness 5-9 years
Income composition of resources
Schooling
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4
−0.6

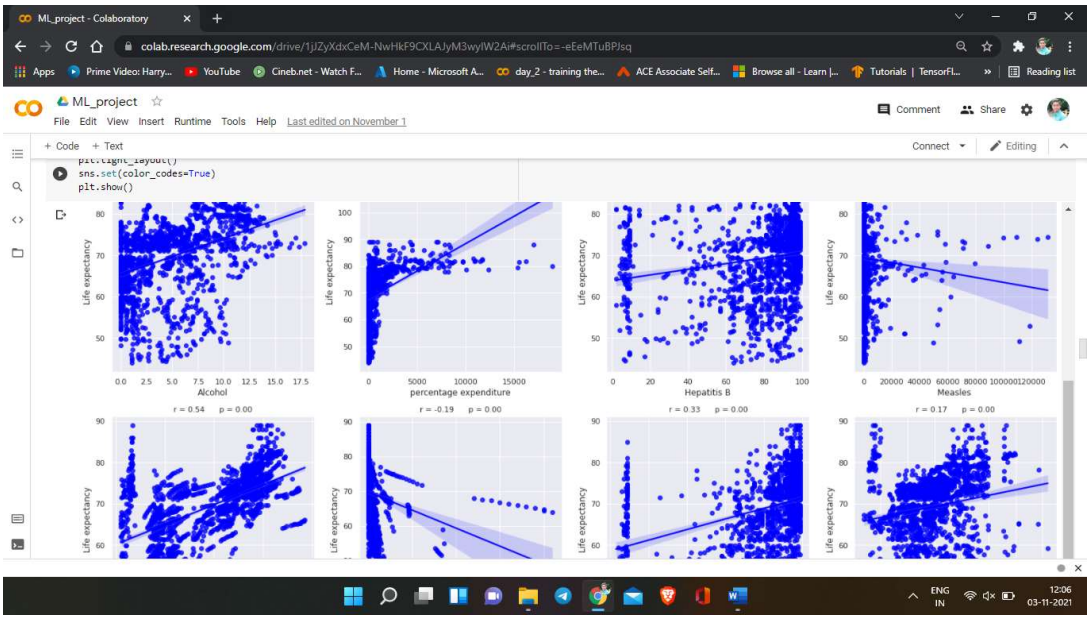

ML_project - Colaboratory
colab.research.google.com/drive/1jIZyXdxCeM-NwHkF9CXLAJyM3wyIW2Ai
ML_project
File Edit View Insert Runtime Tools Help Last edited on November 1
Comment Share
+ Code + Text
Connect Editing
[ ] l= [life_expectancy_data['Life expectancy '].dropna(), life_expectancy_data['Income composition of resources'].dropna(), life_expectancy_data['Schooling'].dropna()]
for i in l:
plt.figure(figsize=(20,5))
sns.histplot(i, kde=True, color="Red")
Count
Life expectancy
400
350
300
250
200
150
100
50
0
40
50
60
70
80
90

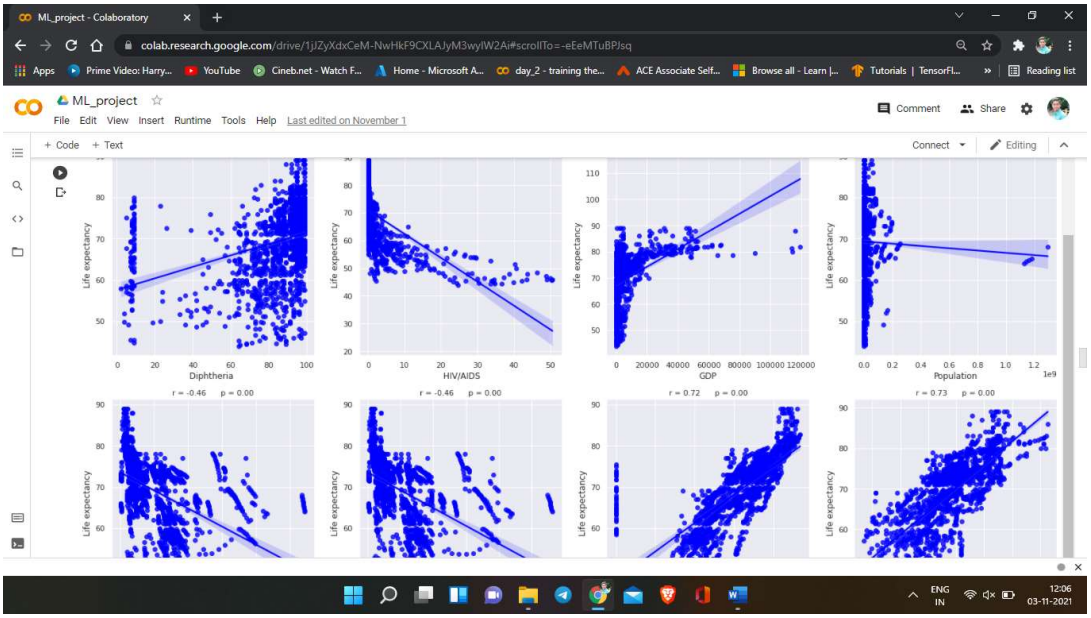

ML_project - Colaboratory
colab.research.google.com/drive/1jIZyXdxCeM-NwHkF9CXLAJyM3wyIW2Ai#scrollTo=-eEeMTuBPJsq
ML_project
File Edit View Insert Runtime Tools Help Last edited on November 1
Comment
Share
+ Code + Text
Connect
Editing
Life expectancy
Count
Income composition of resources

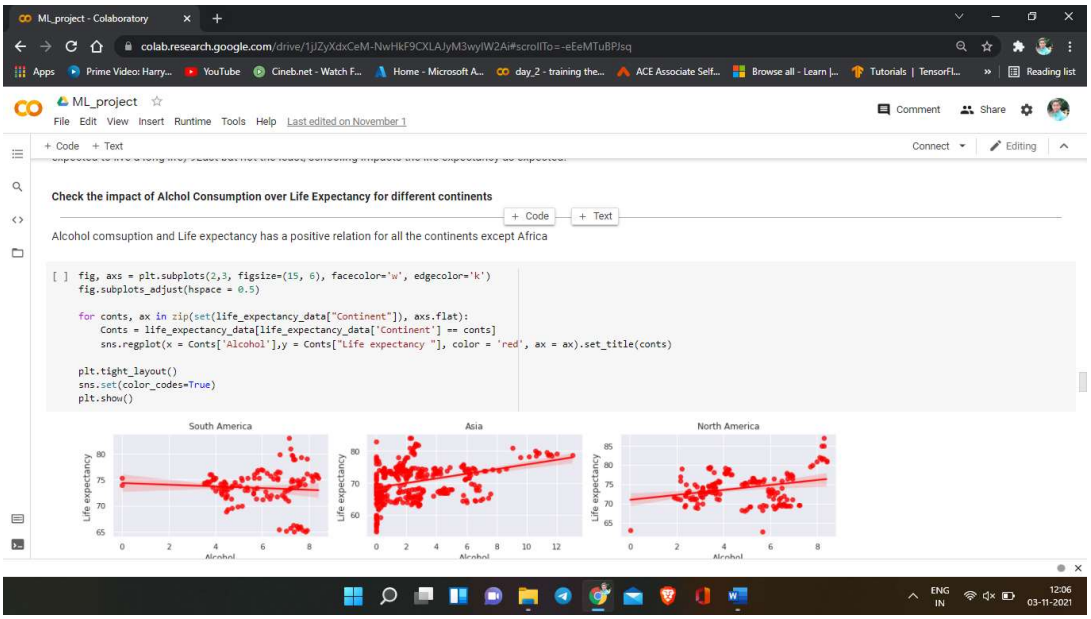

ML_project - Colaboratory
colab.research.google.com/drive/1jIZyXdxCeM-NwHkF9CXLAJyM3wyIW2Ai#scrollTo=-eEeMTuBPJsq
ML_project
File Edit View Insert Runtime Tools Help Last edited on November 1
Comment
Share
+ Code + Text
Connect
Editing
Income composition of resources
Count
Schooling
Life expectency has a dense distribution between 70 - 76 years.
[ ] nr_rows = 5
nr_cols = 4

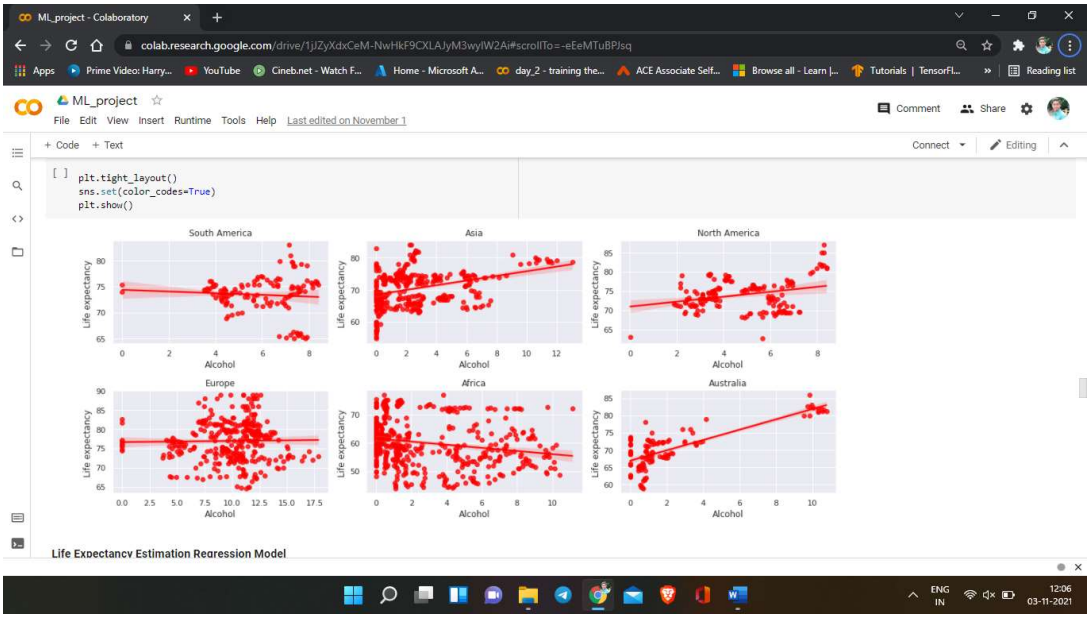

ML_project - Colaboratory
colab.research.google.com/drive/1jIZyXdxCeM-NwHkF9CXLAJyM3wyIW2Ai#scrollTo=-eEeMTuBPJsq
ML_project
File Edit View Insert Runtime Tools Help Last edited on November 1
Comment Share
+ Code + Text
Connect Editing
Life expectancy has a dense distribution between 70 - 75 years.
+ Code + Text
[ ] nr_rows = 5
nr_cols = 4
fig, axs = plt.subplots(nr_rows, nr_cols, figsize=(nr_cols*5,nr_rows*5))
numerical_feats = life_expectancy_data.dtypes[life_expectancy_data.dtypes != "object"].index
li_num_feats = list(numerical_feats)
li_not_plot = []
li_plot_num_feats = [c for c in list(numerical_feats) if c not in li_not_plot]
life_expectancy_data.dropna(inplace = True)
for r in range(0,nr_rows):
for c in range(0,nr_cols):
i = r*nr_cols+c
if i < len(li_plot_num_feats):
sns.regplot(x = life_expectancy_data[li_plot_num_feats[i]],y = life_expectancy_data["Life expectancy "] , color = 'blue', ax = axs[r][c])
stp = stats.pearsonr(life_expectancy_data[li_plot_num_feats[i]], life_expectancy_data["Life expectancy "])
str_title = "r = " + "{0:.2f}".format(stp[0]) + " " "p = " + "{0:.2f}".format(stp[1])
axs[r][c].set_title(str_title,fontsize=11)
plt.tight_layout()
sns.set(color_codes=True)
plt.show()
r = 0.05 p = 0.04
r = 1.00 p = 0.00
r = -0.70 p = 0.00
r = -0.17 p = 0.00
ENG IN
12:05 03-11-2021


ML_project - Colaboratory
colab.research.google.com/drive/1jIZyXdxCeM-NwHkF9CXLAJyM3wyIW2Ai#scrollTo=-eEeMTuBPJsq
ML_project
File Edit View Insert Runtime Tools Help Last edited on November 1
Comment Share
+ Code + Text
Connect Editing
[ ] sns.set(color_codes=True)
plt.show()
r = 0.05 p = 0.04
r = 1.00 p = 0.00
r = -0.70 p = 0.00
r = -0.17 p = 0.00
Life expectancy
Year
Life expectancy
Adult Mortality
infant deaths
r = 0.40 p = 0.00
r = 0.41 p = 0.00
r = 0.20 p = 0.00
r = -0.07 p = 0.01
ENG IN
12:05 03-11-2021

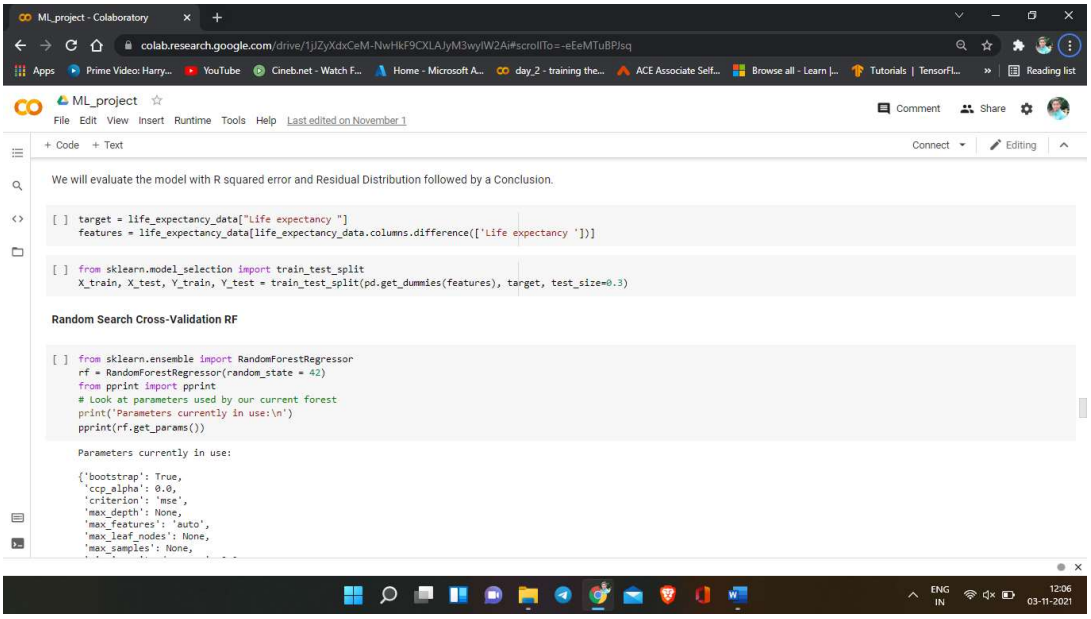

ML_project - Colaboratory
colab.research.google.com/drive/1jIZyXdxCeM-NwHkF9CXLAJyM3wylW2Ai#scrollTo=-eEeMTuBPJsq
ML_project
File Edit View Insert Runtime Tools Help Last edited on November 1
Comment Share
+ Code + Text
Connect Editing
sns.set(color_codes=True)
plt.show()
Life expectancy
Alcohol
percentage expenditure
Hepatitis B
Measles
r = 0.54 p = 0.00
r = -0.19 p = 0.00
r = 0.33 p = 0.00
r = 0.17 p = 0.00

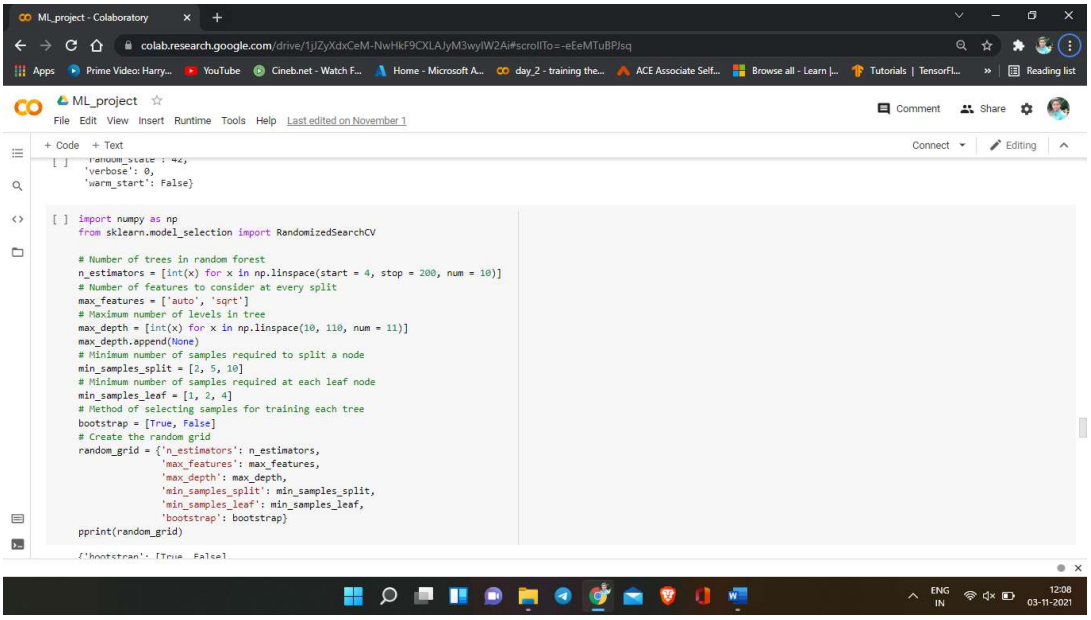

ML_project - Colaboratory
colab.research.google.com/drive/1jIZyXdxCeM-NwHkF9CXLAJyM3wylW2Ai#scrollTo=-eEeMTuBPJsq
ML_project
File Edit View Insert Runtime Tools Help Last edited on November 1
Comment Share
+ Code + Text
Connect Editing
Life expectancy
Diphtheria
HIV/AIDS
GDP
Population
r = -0.46 p = 0.00
r = -0.46 p = 0.00
r = 0.72 p = 0.00
r = 0.73 p = 0.00

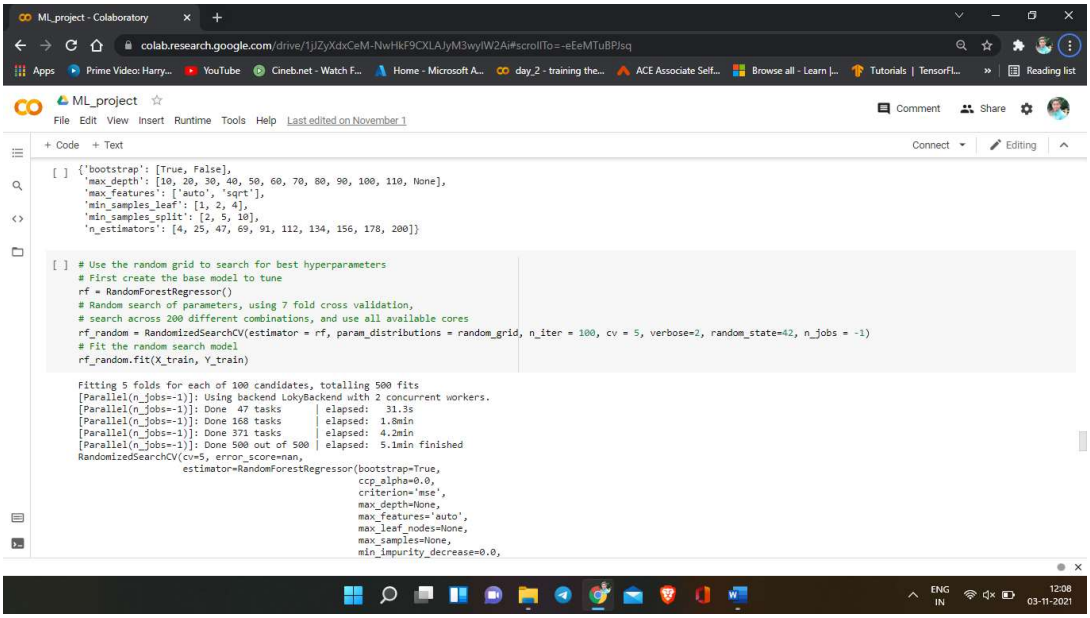

ML_project - Colaboratory
colab.research.google.com/drive/1jIZyXdxCeM-NwHkF9CXLAJyM3wyIW2Ai#scrollTo=-eEeMTuBPJsq
ML_project
File Edit View Insert Runtime Tools Help Last edited on November 1
Comment Share
+ Code + Text
Connect Editing
Check the impact of Alchol Consumption over Life Expectancy for different continents
+ Code + Text
Alcohol comsuption and Life expectancy has a positive relation for all the continents except Africa
[ ] fig, axs = plt.subplots(2,3, figsize=(15, 6), facecolor='w', edgecolor='k')
fig.subplots_adjust(hspace = 0.5)
for conts, ax in zip(set(life_expectancy_data["Continent"]), axs.flat):
Conts = life_expectancy_data[life_expectancy_data['Continent'] == conts]
sns.regplot(x = Conts['Alcohol'],y = Conts["Life expectancy "], color = 'red', ax = ax).set_title(conts)
plt.tight_layout()
sns.set(color_codes=True)
plt.show()
South America
Asia
North America
Life expectancy
Alcohol
ENG IN
12:06
03-11-2021

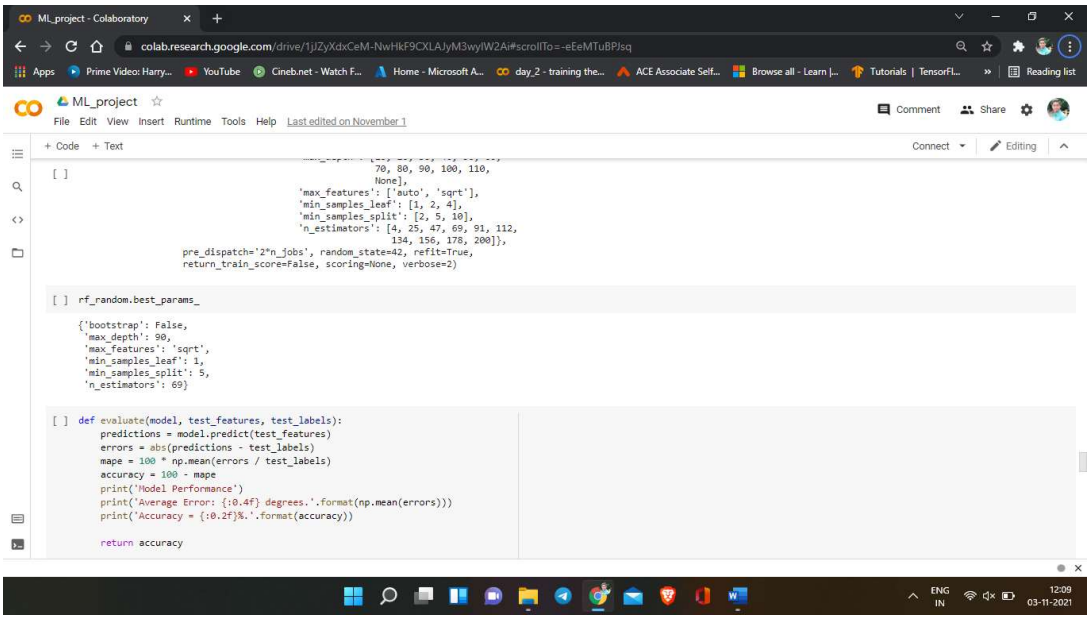

ML_project - Colaboratory
colab.research.google.com/drive/1jIZyXdxCeM-NwHkF9CXLAJyM3wyIW2Ai#scrollTo=-eEeMTuBPJsq
ML_project
File Edit View Insert Runtime Tools Help Last edited on November 1
Comment Share
+ Code + Text
Connect Editing
[ ] plt.tight_layout()
sns.set(color_codes=True)
plt.show()
South America
Asia
North America
Europe
Africa
Australia
Life expectancy
Alcohol
Life Expectancy Estimation Regression Model
ENG IN
12:06
03-11-2021

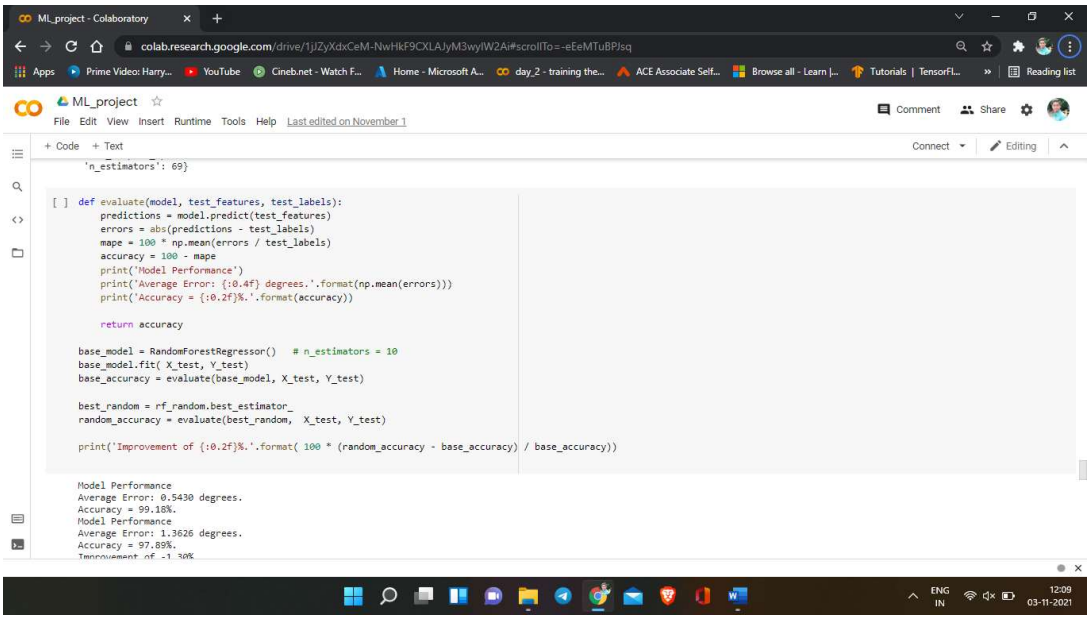
Reading the csv files and performing the prediction by using test validation

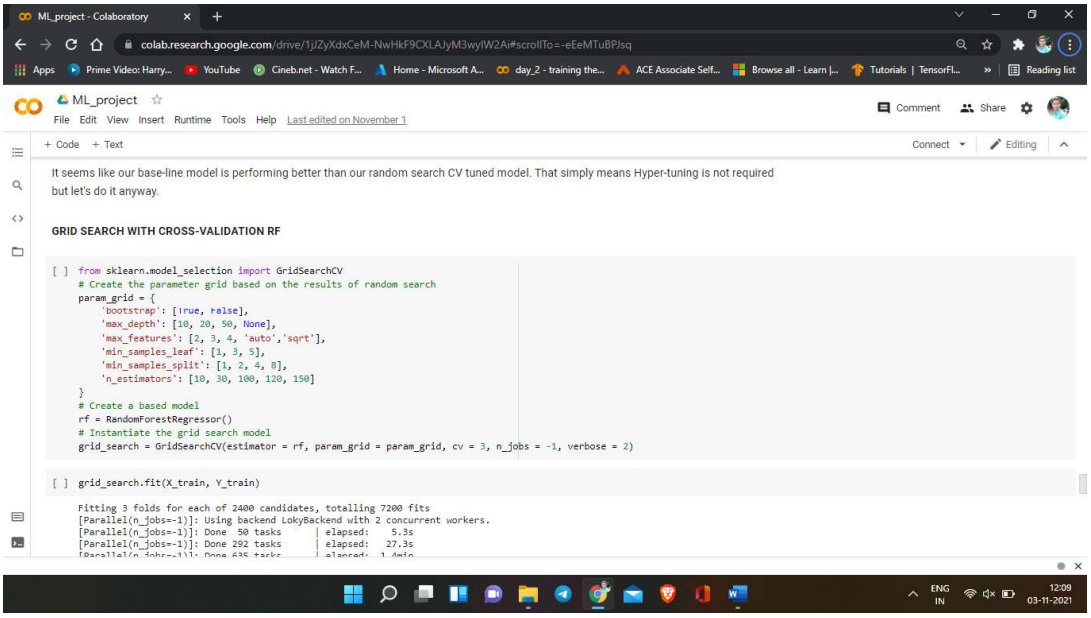


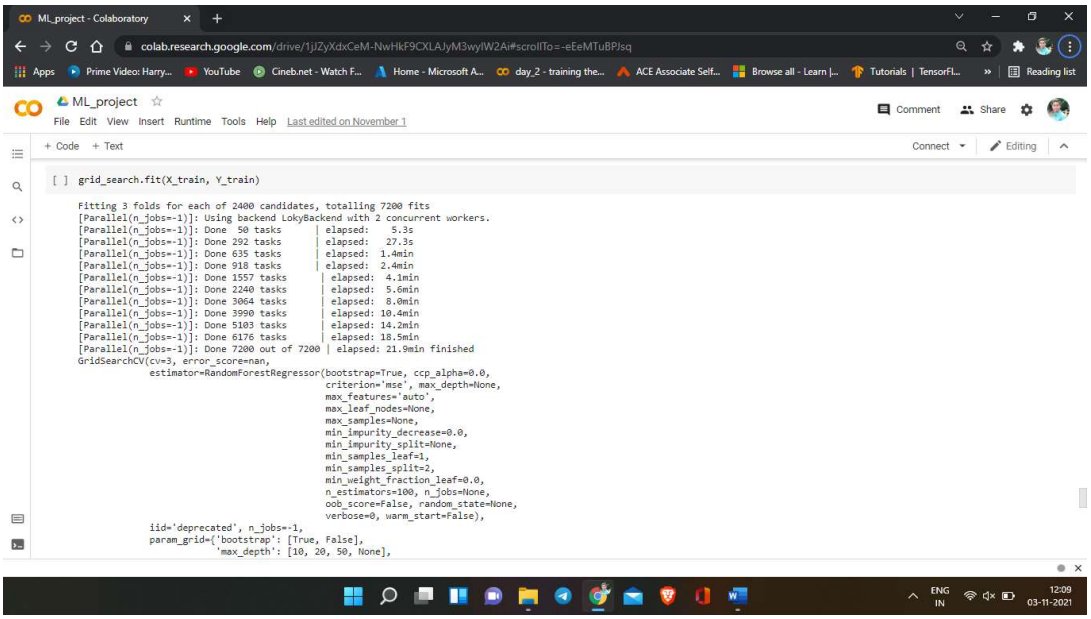


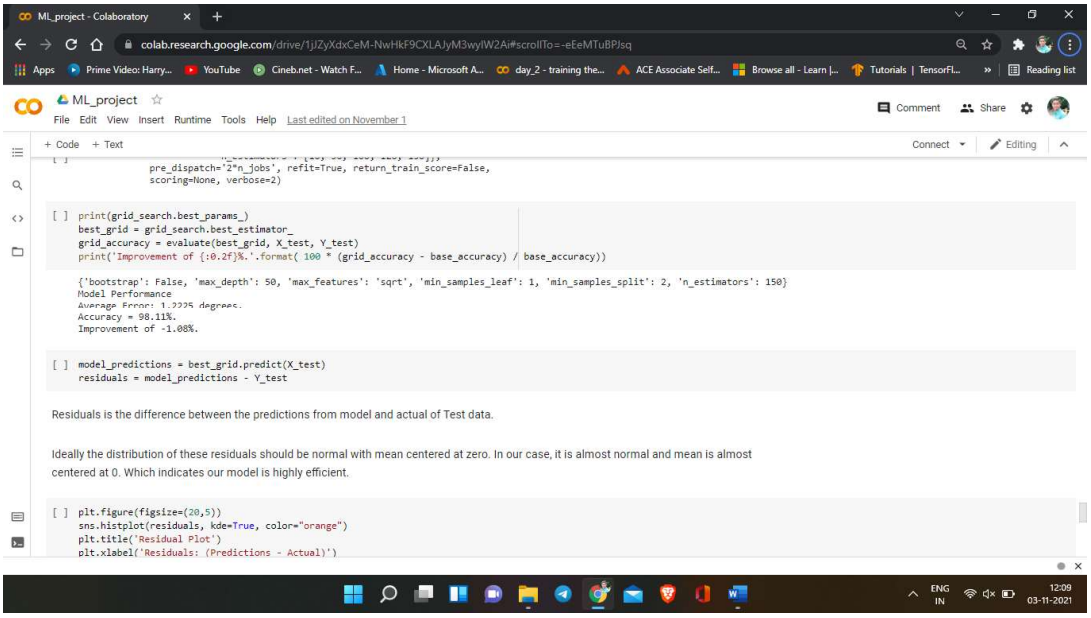

ML_project - Colaboratory
colab.research.google.com/drive/1jlZyXdxCeM-NwHkF9CXLAJyM3wylW2Ai#scrollTo=-eEeMTuBPJsq
ML_project
File Edit View Insert Runtime Tools Help Last edited on November 1
Comment Share
+ Code + Text
Connect Editing
[ ] {'bootstrap': [True, False],
'max_depth': [10, 20, 30, 40, 50, 60, 70, 80, 90, 100, 110, None],
'max_features': ['auto', 'sqrt'],
'min_samples_leaf': [1, 2, 4],
'min_samples_split': [2, 5, 10],
'n_estimators': [4, 25, 47, 69, 91, 112, 134, 156, 178, 200]}
[ ] # Use the random grid to search for best hyperparameters
# First create the base model to tune
rf = RandomForestRegressor()
# Random search of parameters, using 7 fold cross validation,
# search across 200 different combinations, and use all available cores
rf_random = RandomizedSearchCV(estimator = rf, param_distributions = random_grid, n_iter = 100, cv = 5, verbose=2, random_state=42, n_jobs = -1)
# Fit the random search model
rf_random.fit(X_train, Y_train)
Fitting 5 folds for each of 100 candidates, totalling 500 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 2 concurrent workers.
[Parallel(n_jobs=-1)]: Done 47 tasks | elapsed: 31.3s
[Parallel(n_jobs=-1)]: Done 168 tasks | elapsed: 1.8min
[Parallel(n_jobs=-1)]: Done 371 tasks | elapsed: 4.2min
[Parallel(n_jobs=-1)]: Done 500 out of 500 | elapsed: 5.1min finished
RandomizedSearchCV(cv=5, error_score=nan,
estimator=RandomForestRegressor(bootstrap=True,
ccp_alpha=0.0,
criterion='mse',
max_depth=None,
max_features='auto',
max_leaf_nodes=None,
max_samples=None,
min_impurity_decrease=0.0,
ENG IN
12:08
03-11-2021

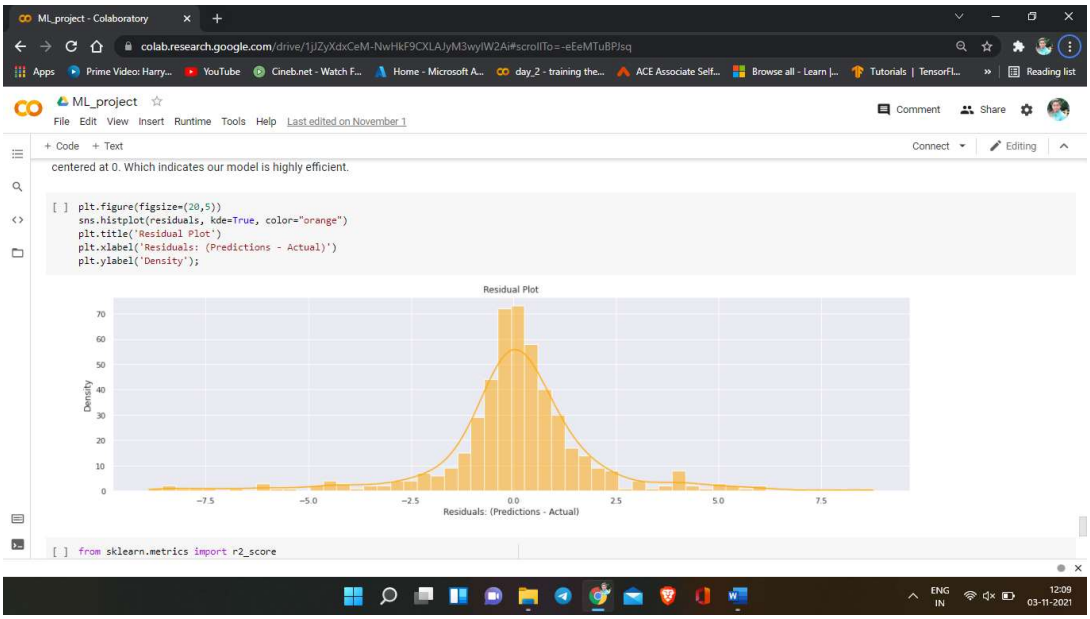

ML_project - Colaboratory
colab.research.google.com/drive/1jlZyXdxCeM-NwHkF9CXLAJyM3wylW2Ai#scrollTo=-eEeMTuBPJsq
ML_project
File Edit View Insert Runtime Tools Help Last edited on November 1
Comment Share
+ Code + Text
Connect Editing
[ ]
70, 80, 90, 100, 110,
None],
'max_features': ['auto', 'sqrt'],
'min_samples_leaf': [1, 2, 4],
'min_samples_split': [2, 5, 10],
'n_estimators': [4, 25, 47, 69, 91, 112,
134, 156, 178, 200]},
pre_dispatch='2*n_jobs', random_state=42, refit=True,
return_train_score=False, scoring=None, verbose=2)
[ ] rf_random.best_params_
{'bootstrap': False,
'max_depth': 90,
'max_features': 'sqrt',
'min_samples_leaf': 1,
'min_samples_split': 5,
'n_estimators': 69}
[ ] def evaluate(model, test_features, test_labels):
predictions = model.predict(test_features)
errors = abs(predictions - test_labels)
mape = 100 * np.mean(errors / test_labels)
accuracy = 100 - mape
print('Model Performance')
print('Average Error: {:0.4f} degrees.'.format(np.mean(errors)))
print('Accuracy = {:0.2f}%.'.format(accuracy))
return accuracy
ENG IN
12:09
03-11-2021

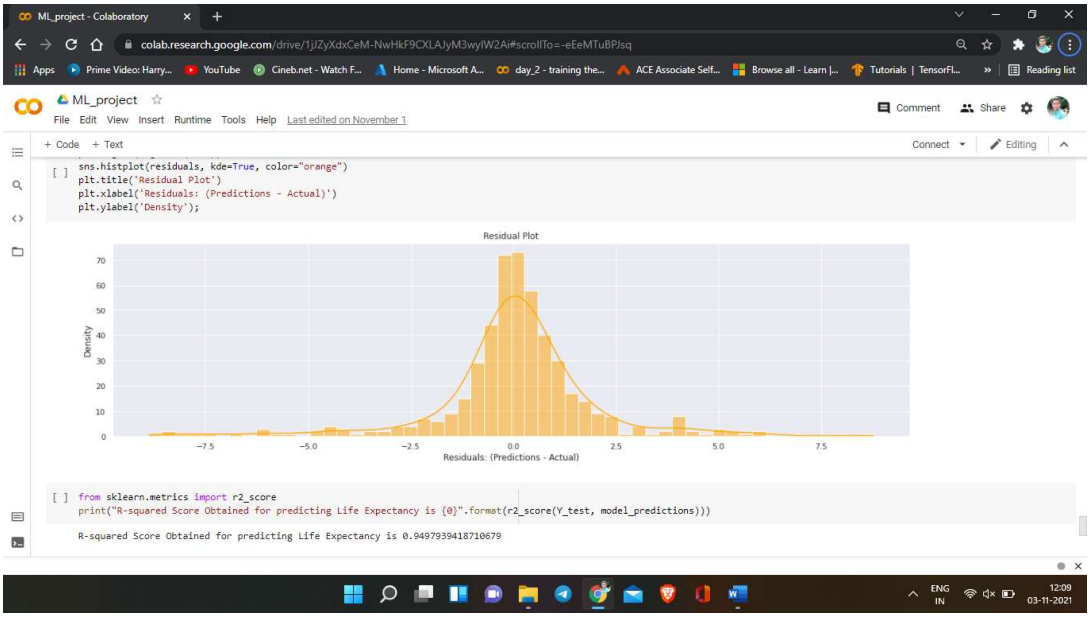

ML_project - Colaboratory
colab.research.google.com/drive/1jIZyXdxCeM-NwHkF9CXLAJyM3wyIW2Ai#scrollTo=-eEeMTuBPJsq
ML_project
File Edit View Insert Runtime Tools Help Last edited on November 1
Comment
Share
Connect
Editing
+ Code + Text
'n_estimators': 69}
[ ] def evaluate(model, test_features, test_labels):
predictions = model.predict(test_features)
errors = abs(predictions - test_labels)
mape = 100 * np.mean(errors / test_labels)
accuracy = 100 - mape
print('Model Performance')
print('Average Error: {:0.4f} degrees.'.format(np.mean(errors)))
print('Accuracy = {:0.2f}%.'.format(accuracy))
return accuracy
base_model = RandomForestRegressor() # n_estimators = 10
base_model.fit(X_test, Y_test)
base_accuracy = evaluate(base_model, X_test, Y_test)
best_random = rf_random.best_estimator_
random_accuracy = evaluate(best_random, X_test, Y_test)
print('Improvement of {:0.2f}%.'.format(100 * (random_accuracy - base_accuracy) / base_accuracy))
Model Performance
Average Error: 0.5430 degrees.
Accuracy = 99.18%.
Model Performance
Average Error: 1.3626 degrees.
Accuracy = 97.89%.
ENG
IN
12:09
03-11-2021


ML_project - Colaboratory
colab.research.google.com/drive/1jIZyXdxCeM-NwHkF9CXLAJyM3wyIW2Ai#scrollTo=-eEeMTuBPJsq
ML_project
File Edit View Insert Runtime Tools Help Last edited on November 1
Comment
Share
Connect
Editing
+ Code + Text
It seems like our base-line model is performing better than our random search CV tuned model. That simply means Hyper-tuning is not required
but let's do it anyway.
GRID SEARCH WITH CROSS-VALIDATION RF
[ ] from sklearn.model_selection import GridSearchCV
# Create the parameter grid based on the results of random search
param_grid = {
'bootstrap': [True, False],
'max_depth': [10, 20, 50, None],
'max_features': [2, 3, 4, 'auto','sqrt'],
'min_samples_leaf': [1, 3, 5],
'min_samples_split': [1, 2, 4, 8],
'n_estimators': [10, 30, 100, 120, 150]
}
# Create a based model
rf = RandomForestRegressor()
# Instantiate the grid search model
grid_search = GridSearchCV(estimator = rf, param_grid = param_grid, cv = 3, n_jobs = -1, verbose = 2)
[ ] grid_search.fit(X_train, Y_train)
Fitting 3 folds for each of 2400 candidates, totalling 7200 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 2 concurrent workers.
[Parallel(n_jobs=-1)]: Done 50 tasks | elapsed: 5.3s
[Parallel(n_jobs=-1)]: Done 292 tasks | elapsed: 27.3s
ENG
IN
12:09
03-11-2021


ML_project - Colaboratory
colab.research.google.com/drive/1jlZyXdxCeM-NwHkF9CXLAJyM3wylW2Ai#scrollTo=-eEeMTuBPJsq
ML_project
File Edit View Insert Runtime Tools Help Last edited on November 1
Comment
Share
+ Code + Text
Connect
Editing
[ ] grid_search.fit(X_train, Y_train)
Fitting 3 folds for each of 2400 candidates, totalling 7200 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 2 concurrent workers.
[Parallel(n_jobs=-1)]: Done 50 tasks | elapsed: 5.3s
[Parallel(n_jobs=-1)]: Done 292 tasks | elapsed: 27.3s
[Parallel(n_jobs=-1)]: Done 635 tasks | elapsed: 1.4min
[Parallel(n_jobs=-1)]: Done 918 tasks | elapsed: 2.4min
[Parallel(n_jobs=-1)]: Done 1557 tasks | elapsed: 4.1min
[Parallel(n_jobs=-1)]: Done 2240 tasks | elapsed: 5.6min
[Parallel(n_jobs=-1)]: Done 3064 tasks | elapsed: 8.0min
[Parallel(n_jobs=-1)]: Done 3990 tasks | elapsed: 10.4min
[Parallel(n_jobs=-1)]: Done 5103 tasks | elapsed: 14.2min
[Parallel(n_jobs=-1)]: Done 6176 tasks | elapsed: 18.5min
[Parallel(n_jobs=-1)]: Done 7200 out of 7200 | elapsed: 21.9min finished
GridSearchCV(cv=3, error_score=nan,
estimator=RandomForestRegressor(bootstrap=True, ccp_alpha=0.0,
criterion='mse', max_depth=None,
max_features='auto',
max_leaf_nodes=None,
max_samples=None,
min_impurity_decrease=0.0,
min_impurity_split=None,
min_samples_leaf=1,
min_samples_split=2,
min_weight_fraction_leaf=0.0,
n_estimators=100, n_jobs=None,
oob_score=False, random_state=None,
verbose=0, warm_start=False),
iid='deprecated', n_jobs=-1,
param_grid={'bootstrap': [True, False],
'max_depth': [10, 20, 50, None],
ENG IN
12:09
03-11-2021


ML_project - Colaboratory
colab.research.google.com/drive/1jlZyXdxCeM-NwHkF9CXLAJyM3wylW2Ai#scrollTo=-eEeMTuBPJsq
ML_project
File Edit View Insert Runtime Tools Help Last edited on November 1
Comment
Share
+ Code + Text
Connect
Editing
pre_dispatch='2*n_jobs', refit=True, return_train_score=False,
scoring=None, verbose=2)
[ ] print(grid_search.best_params_)
best_grid = grid_search.best_estimator_
grid_accuracy = evaluate(best_grid, X_test, Y_test)
print('Improvement of {:0.2f}%.'.format(100 * (grid_accuracy - base_accuracy) / base_accuracy))
{'bootstrap': False, 'max_depth': 50, 'max_features': 'sqrt', 'min_samples_leaf': 1, 'min_samples_split': 2, 'n_estimators': 150}
Model Performance
Average Error: 1.2225 degrees.
Accuracy = 98.11%.
Improvement of -1.08%.
[ ] model_predictions = best_grid.predict(X_test)
residuals = model_predictions - Y_test
Residuals is the difference between the predictions from model and actual of Test data.
Ideally the distribution of these residuals should be normal with mean centered at zero. In our case, it is almost normal and mean is almost centered at 0. Which indicates our model is highly efficient.
[ ] plt.figure(figsize=(20,5))
sns.histplot(residuals, kde=True, color="orange")
plt.title('Residual Plot')
plt.xlabel('Residuals: (Predictions - Actual)')
ENG IN
12:09
03-11-2021


ML_project - Colaboratory
colab.research.google.com/drive/1jlZyXdxCeM-NwHkF9CXLAJyM3wylW2Ai#scrollTo=-eEeMTuBPJsq
ML_project
File Edit View Insert Runtime Tools Help Last edited on November 1
Comment Share
+ Code + Text
Connect Editing
centered at 0. Which indicates our model is highly efficient.
[ ] plt.figure(figsize=(20,5))
sns.histplot(residuals, kde=True, color="orange")
plt.title('Residual Plot')
plt.xlabel('Residuals: (Predictions - Actual)')
plt.ylabel('Density');
Residual Plot
Density
Residuals: (Predictions - Actual)
[ ] from sklearn.metrics import r2_score
ENG IN
12:09 03-11-2021


ML_project - Colaboratory
colab.research.google.com/drive/1jlZyXdxCeM-NwHkF9CXLAJyM3wylW2Ai#scrollTo=-eEeMTuBPJsq
ML_project
File Edit View Insert Runtime Tools Help Last edited on November 1
Comment Share
+ Code + Text
Connect Editing
[ ] sns.histplot(residuals, kde=True, color="orange")
plt.title('Residual Plot')
plt.xlabel('Residuals: (Predictions - Actual)')
plt.ylabel('Density');
Residual Plot
Density
Residuals: (Predictions - Actual)
[ ] from sklearn.metrics import r2_score
print("R-squared Score Obtained for predicting Life Expectancy is {0}".format(r2_score(Y_test, model_predictions)))
R-squared Score Obtained for predicting Life Expectancy is 0.9497939418710679
ENG IN
12:09 03-11-2021

Result

- ➔ Hence, we have identified the life expectancy that can be obtained from the WHO dataset from the test and train Data using Regression Techniques of Supervised Machine learning by Implementing in Python Programming language.
- ➔ We observed life expectancy from given data using machine learning technique in Python language.
  - ➔ This application is a perfect use case for regression, which determines the relationship between one dependent variable (life expectancy) and a number of independent variables (development indicators).

Conclusion

- ➔ One can extend his/her life span by adopting a healthy life-style, proper education, and by getting vaccinated. Ofcourse Demographic location plays an important role. In our analysis, we saw people living in Europe has a higher life-span as compared to other continents. Country's GDP and Income composition affects the Life Expectancy in a broader way.
  - ➔ There are some parameters like pollution and environmental index that has been missing in this analysis and expected to be highly related with Life Expectancy. Getting Vaccinated for various diseases also increases the expected life span. Consumption of Alcohol and Life expectancy has an unexpected positive relation except for Africa.

Thank You