STATISTICAL INFERENCE AND DATA MINING:

CODING ASSIGNMENT CONTEST 1

ASSIGNED BY: EFI KORENFELD SAAR BEN-YOCHANA

KAGGLE TEAM: EFI & SAAR | **KAGGLE USERNAME**: EFIKORENFELD237

FIRST LOOK AT THE DATA

We started our work by reviewing the given dataset: train and test separately.

Train:

	Year	Life expectancy	Adult Mortality	infant deaths	Alcohol	percentage expenditure	Hepatitis B	Measles	BMI	under-five deaths	Polio	Total expenditure	Diphtheria	HIV/AIDS
count	2342.000000	2333.000000	2333.000000	2342.000000	2186.000000	2342.000000	1916.000000	2342.000000	2314.000000	2342.000000	2327.000000	2156.000000	2327.000000	2342.000000
mean	2007.532878	69.302443	163.607801	29.577711	4.650284	725.542876	81.172756	2288.827071	38.523207	40.975662	82.424581	5.905751	82.397078	1.717165
std	4.621651	9.514162	123.751231	110.595193	4.063965	1902.153265	24.647995	10702.614535	20.021458	149.769331	23.665432	2.452172	23.559671	5.177235
min	2000.000000	39.000000	1.000000	0.000000	0.010000	0.000000	1.000000	0.000000	1.000000	0.000000	3.000000	0.370000	2.000000	0.100000
25%	2003.000000	63.500000	74.000000	0.000000	0.870000	4.685343	77.000000	0.000000	19.400000	0.000000	78.000000	4.280000	78.000000	0.100000
50%	2008.000000	72.100000	144.000000	3.000000	3.835000	65.268121	92.000000	15.500000	43.900000	4.000000	93.000000	5.740000	93.000000	0.100000
75%	2012.000000	75.800000	225.000000	21.000000	7.807500	446.887709	96.000000	372.000000	56.300000	26.000000	97.000000	7.462500	97.000000	0.700000
max	2015.000000	89.000000	723.000000	1700.000000	17.870000	19479.911610	99.000000	212183.000000	87.300000	2200.000000	99.000000	17.600000	99.000000	50.600000

Train-set shape: (2342,22)

Test:

	ID	Year	Adult Mortality	infant deaths	Alcohol	percentage expenditure	Hepatitis B	Measles	BMI	under-five deaths	Polio	Total expenditure	Diphtheria	HIV/AIDS	
count	595.000000	595.000000	595.000000	595.000000	557.000000	595.000000	468.000000	595.000000	589.000000	595.000000	591.000000	555.000000	591.000000	595.000000	497
mean	298.776471	2007.453782	169.457143	33.213445	4.424991	789.514100	79.952991	2938.368067	37.452292	46.278992	83.018613	6.068450	82.010152	1.843025	7739
std	172.207099	4.584684	126.387889	143.332182	4.004268	2296.898067	26.747249	14085.626757	20.060070	197.120644	22.493871	2.668645	24.355069	4.672701	15329
min	1.000000	2000.000000	1.000000	0.000000	0.010000	0.000000	2.000000	0.000000	2.100000	0.000000	3.000000	0.740000	3.000000	0.100000	14
25%	150.500000	2004.000000	74.000000	0.000000	0.920000	5.250309	77.000000	0.000000	19.100000	0.000000	79.000000	4.240000	78.500000	0.100000	458
50%	299.000000	2007.000000	145.000000	3.000000	3.480000	64.605901	92.000000	21.000000	41.900000	4.000000	93.000000	5.760000	93.000000	0.100000	1577
75%	447.500000	2012.000000	232.500000	23.500000	7.530000	419.874405	97.000000	341.500000	55.700000	31.500000	97.000000	7.565000	97.000000	1.200000	5593
max	596.000000	2015.000000	686.000000	1800.000000	15.520000	19099.045060	99.000000	182485.000000	75.200000	2500.000000	99.000000	17.200000	99.000000	42.100000	115761

Test-set shape: (595,22)

Missing values of the target feature – Life expectancy:

	country	year	status	life_expectancy	adult_mortality	infant_deaths	alcohol	percentage_expenditure	hepatitis_b	measles	...	polio	total_expenditure	diphtheria
610	Dominica	2013	Developing	NaN	NaN	0	0.01	11.419555	96.0	0	...	96.0	5.58	96.0
1313	Marshall Islands	2013	Developing	NaN	NaN	0	0.01	871.878317	8.0	0	...	79.0	17.24	79.0
1369	Monaco	2013	Developing	NaN	NaN	0	0.01	0.000000	99.0	0	...	99.0	4.30	99.0
1442	Nauru	2013	Developing	NaN	NaN	0	0.01	15.606596	87.0	0	...	87.0	4.65	87.0
1523	Niue	2013	Developing	NaN	NaN	0	0.01	0.000000	99.0	0	...	99.0	7.20	99.0
1563	Palau	2013	Developing	NaN	NaN	0	NaN	344.690631	99.0	0	...	99.0	9.27	99.0
1739	Saint Kitts and Nevis	2013	Developing	NaN	NaN	0	8.54	0.000000	97.0	0	...	96.0	6.14	96.0
1777	San Marino	2013	Developing	NaN	NaN	0	0.01	0.000000	69.0	0	...	69.0	6.50	69.0
2173	Tuvalu	2013	Developing	NaN	NaN	0	0.01	78.281203	9.0	0	...	9.0	16.61	9.0

According to our observation, all the countries where the target feature is missing are appearing only once in the data and contain multiple missing features, therefore, we decided to drop them.

Next, reviewing the train and test sets, we figured out that the features: **under-five deaths**, **infant deaths**, and **measles** which are defined as count per 1,000 people contain some unreasonable samples.

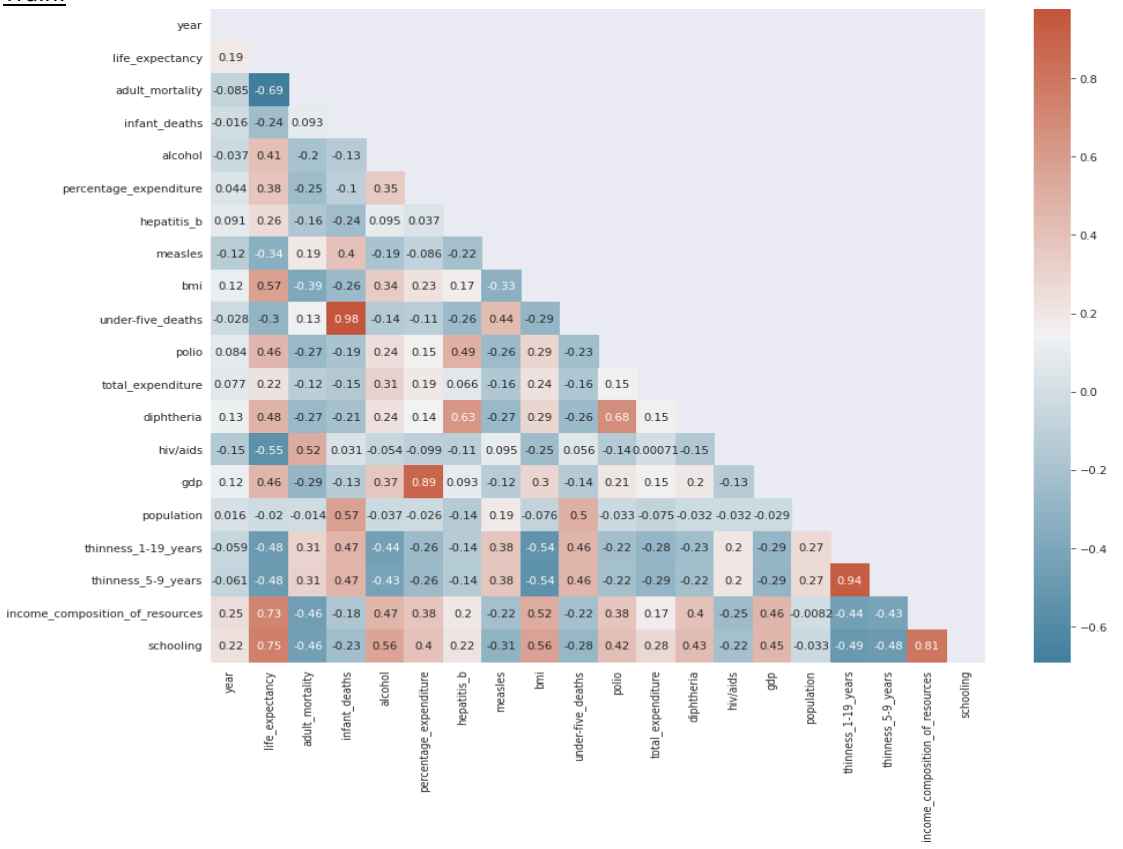
(Based on the metadata of the official dataset on Kaggle)We decided to replace all values greater than 1000 with the maximum possible (1000).

Note: The **BMI** values also do not make sense, the normal range is 18.5-30, Extreme values such as 87 or 2 are not reasonable, but they appear both in the train and test set, thus, we decided to keep them assuming maybe the scale of this measurement is different.

DEEPER LOOK AT THE FEATURES

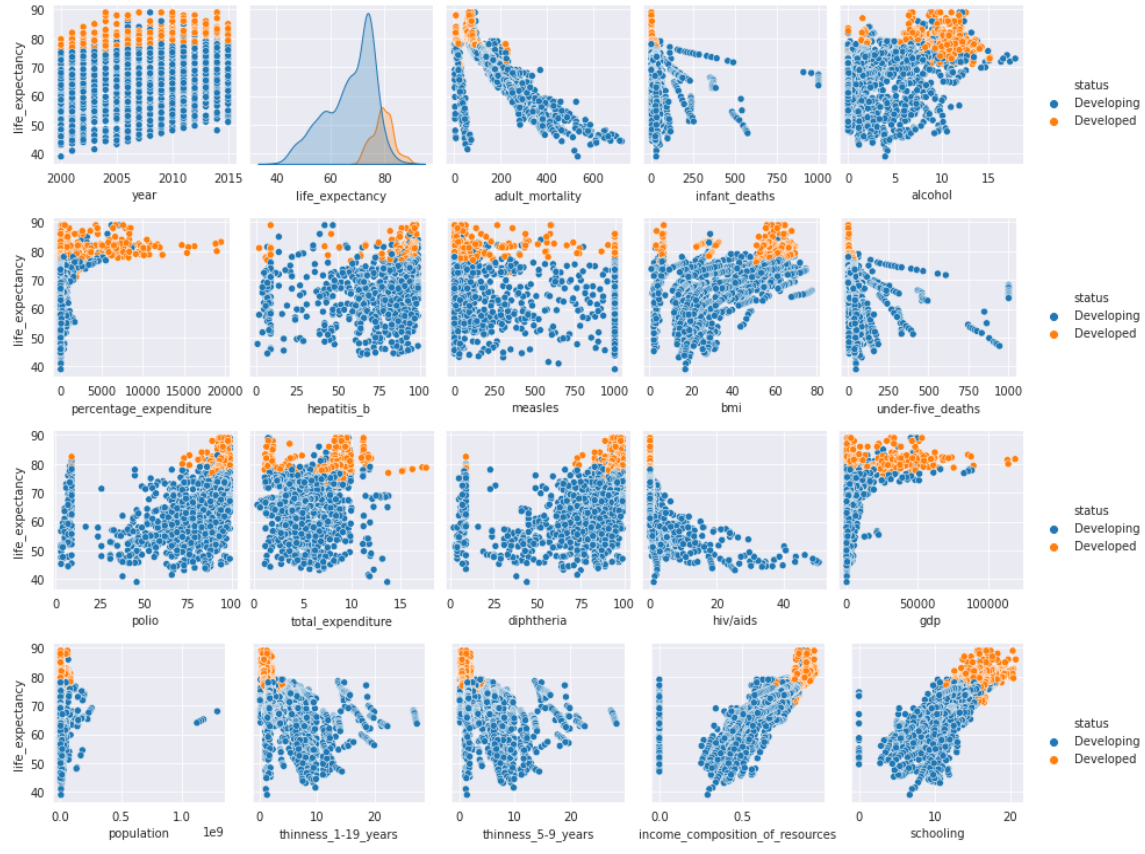
Correlation check:

Train:



Similarly, we checked for the test-set and the results were very close, some features are very correlated with each other and specifically with our target feature – Life expectancy.

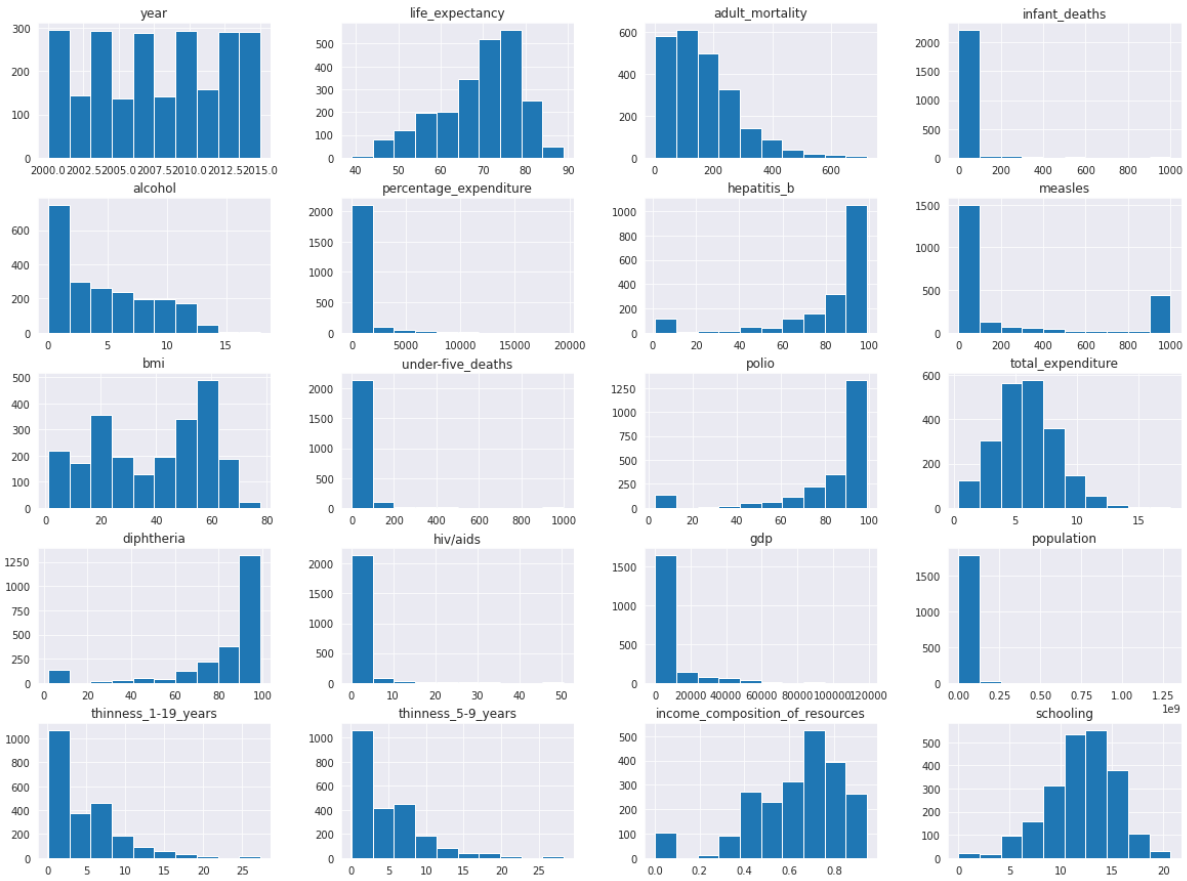
Target feature distribution as a function of all other numeric features separated by Status feature:



From the plots above you can see that the distributions of **Developed** and **Developing** countries are different, where developed countries have a higher average life expectancy for each feature, moreover, by observing the distribution of life expectancy based on the status feature we can tell that the developed countries life expectancy mean is slightly higher than the developing countries and its variance is much lower.

Visualize the distribution of each feature:

Train:



Similarly for test.

After looking at correlations, distributions, and the data itself, we assume the train and test have the same distribution for all features (Moreover, they probably complete each other - we saw that in countries that have a missing row then the row appears on the test set).

Since we are required to handle the missing values (both in train and test), we will merge them together during pre-processing, and later split them when we train the model.

After merging them we checked again the correlations and distributions of features and saw no significant change.

DATA CLEANING

First, we should handle the missing values, our strategy was:

- Filling the missing values instead of dropping them since they are a great share of the data.
- Interpolating some of the missing values by country and year because we saw that year has a non-zero correlation with other features. (A finer average).

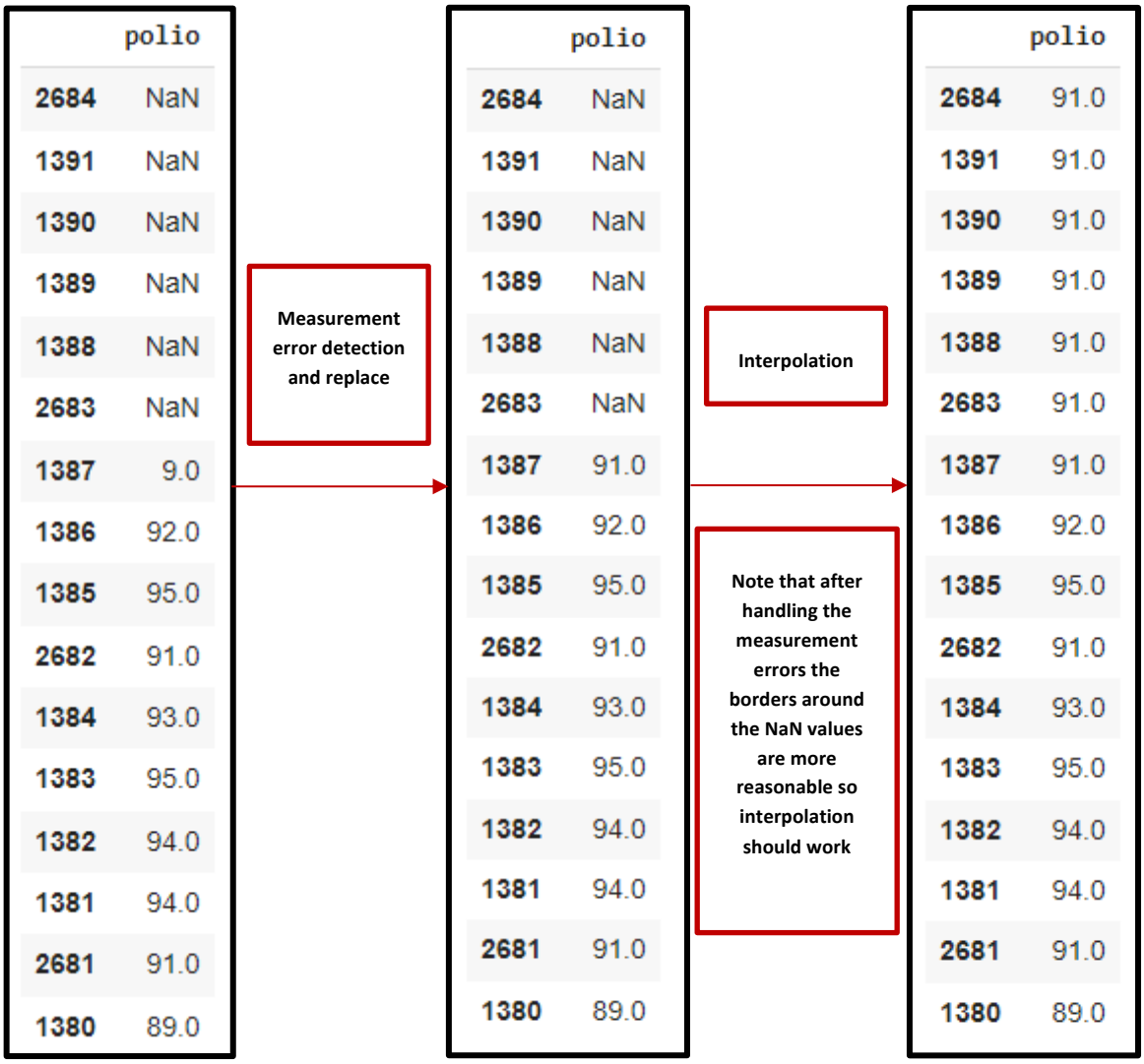
Before interpolating, we noticed that a significant amount of our data contains what we believe are measurement errors, most of them are too small measurements (including zeros).

For example, after reviewing the **polio** feature of Montenegro, we saw that most of the measurements are around the value - 90, whereas a single measurement was with a value of 9. Direct interpolation of the data will result in this error propagating.

To avoid it, we computed the medians of all features, grouped by status developed/developing, and we replaced the error measurements with this median according to the status of the country before interpolating. We defined measurement error as a sample less than 10% of the corresponding median.

After doing so, we interpolated the missing values by country, sorted by years. We used interpolation's limit_direction parameter and set it to "both", in order to fill the first and last values in case they are missing.

<u>Demonstrate the steps of our method on the polio feature of Montenegro</u>:



We performed the same procedure we have demonstrated on the polio feature of Montenegro on each one of features and for each country separately.

After interpolating, we still have some missing values:

	Zero Values	Missing Values	Total	% Total	>1000	Mean	Median	Data Type
country	0	0	0	0.00	-	-	-	object
hiv/aids	0	0	0	0.00	0.0	1.75	0.1	float64
diphtheria	0	0	0	0.00	0.0	87.08	93.0	float64
polio	0	0	0	0.00	0.0	87.36	93.0	float64
measles	0	0	0	0.00	0.0	259.04	18.5	float64
percentage_expenditure	0	0	0	0.00	413.0	786.65	91.2	float64
type	0	0	0	0.00	0.0	0.59	1.0	int64
adult_mortality	0	0	0	0.00	0.0	174.4	159.0	float64
status	0	0	0	0.00	-	-	-	object
year	0	0	0	0.00	2928.0	2007.5	2007.5	int64
alcohol	0	16	16	0.55	0.0	5.03	3.7	float64
bmi	0	32	32	1.09	0.0	39.38	43.6	float64
total_expenditure	0	32	32	1.09	0.0	5.93	5.73	float64
thinness_1-19_years	0	32	32	1.09	0.0	4.95	3.5	float64
thinness_5-9_years	0	32	32	1.09	0.0	4.99	3.5	float64
hepatitis_b	0	144	144	4.92	0.0	83.08	91.0	float64
schooling	0	160	160	5.46	0.0	12.11	12.3	float64
income_composition_of_resources	0	160	160	5.46	0.0	0.66	0.68	float64
under-five_deaths	298	0	298	10.18	0.0	38.88	7.0	float64
infant_deaths	321	0	321	10.96	0.0	29.42	6.0	float64
gdp	0	400	400	13.66	1748.0	7694.31	1848.42	float64
population	0	640	640	21.86	2288.0	13023116.07	1412966.5	float64

Since the **Population** and **GDP** features have a massive amount of missing values even after the interpolation, we decided to drop them completely.

Features skewness:

By observing the skewness of all the features, we can see that some of them are very skewed.

(Note: the values are shown in absolute values)

```
year                              0.000000
bmi                               0.228074
schooling                         0.287945
income_composition_of_resources   0.345260
total_expenditure                 0.589546
alcohol                           0.733634
measles                           1.237974
adult_mortality                   1.294901
type                              1.475899
thinness_1-19_years               1.736771
hepatitis_b                       1.755153
polio                             1.800568
thinness_5-9_years                1.808592
diphtheria                        1.972836
percentage_expenditure            4.648384
hiv/aids                          5.386623
under-five_deaths                 6.200016
infant_deaths                     7.444253
dtype: float64
```

Since the data is skewed and still contains unreasonable values such as the **BMI** and classification errors in the **Status**, we decided to fill the missing values each one by the closest neighbor rows, using the KNN-imputer, the idea behind it is that because we think we don't have enough information to fill the missing values by ourselves, we used the most-likely values based on the data we have.

After the imputation, we verified that we have no missing values left.

FEATURE SELECTION

First, we calculate the variance inflation factor for each feature whereas, the thumb rule is if a feature has a VIF value that is higher than 4 then it implies that there are features with a high correlation with each other.

The VIF value per feature:

	Variable	VIF
15	total_expenditure	1.198786
1	percentage_expenditure	1.341191
13	measles	1.388364
5	hiv/aids	1.483533
4	hepatitis_b	1.619581
9	alcohol	1.768555
10	bmi	1.788426
6	adult_mortality	1.905120
0	polio	4.775738
11	diphtheria	5.005989
3	schooling	6.354435
7	income_composition_of_resources	7.792360
2	thinness_5-9_years	8.222832
14	thinness_1-19_years	8.244004
8	infant_deaths	28.446451
12	under-five_deaths	29.227470

There are some features that have a VIF value that is higher than 4, In order to decide which features to drop we checked the correlations between them.

- **under-five_deaths** and **infant_death** both have high VIF and high correlation with each other, therefore we decided to drop the one with the higher VIF value, which is: **under-five_deaths**.
- For **thinness_1-19_years** vs **thinness_5-9_years**, dropped: **thinness_1-19_years**.
- For **income_composition_of_resources** vs **schooling** dropped: **income_composition_of_resources**
- For **diphtheria** vs **polio** dropped: **diphtheria**

The updated VIF value per feature:

	Variable	VIF
8	total_expenditure	1.160288
4	percentage_expenditure	1.285900
6	measles	1.356768
10	hiv/aids	1.441495
0	infant_deaths	1.452552
9	hepatitis_b	1.519067
2	alcohol	1.699827
1	bmi	1.721765
11	adult_mortality	1.791954
3	polio	1.913296
5	thinness_5-9_years	1.920494
7	schooling	2.932745

Encoding:

Although the **Year** feature is a numeric feature, the number itself doesn't have a useful meaning for the prediction. Since the **Year** feature is an ordinal feature that could indicate changes over a time series, we decided to encode it by using a label encoder.

There are two more categorical features in the dataset: **Status** and **Country** so we encoded them too using one-hot encoder.

MODEL SELECTION

First, we perform a training process using the classic Linear Regression model, we achieved the following score:

Linear Regression Score: 0.9673520327592923

Checking the Linear Regression assumptions:

- **Mean of residuals** – the value should be close to zero.
- **Homoscedasticity** - the residuals should have equal or almost equal variance across the regression line.
- **Normality of residuals** – The residue should be distributed normally.
- **Autocorrelation of residuals** - should be non-significant.

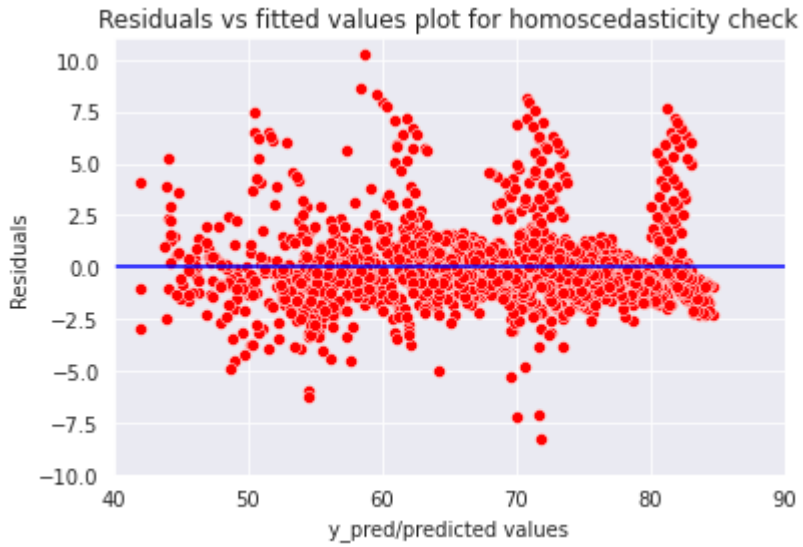
1. **Mean of residuals:**

Mean of Residuals: 1.7899277344961264e-08

The mean is indeed close to 0.
Assumption 1 holds.

2. **Homoscedasticity:**

Residuals visualization:



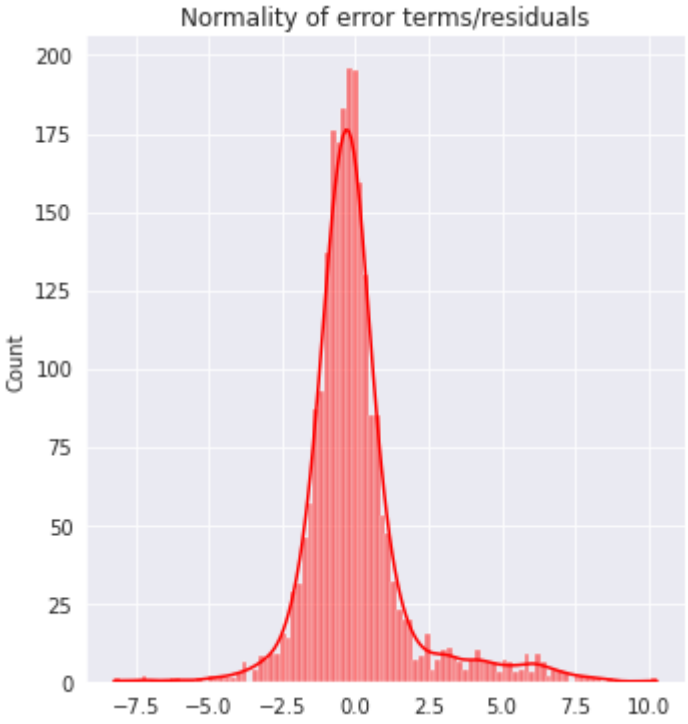
Checking heteroscedasticity:
Using Goldfeld Quandt test for heteroscedasticity with significance (alpha) value of 0.05.

Null Hypothesis: Error terms are homoscedastic
Alternative Hypothesis: Error terms are heteroscedastic.

[('F statistic', 1.4342001143046996), ('p-value', 2.328818939077946e-09)]
Since p-value is less than 0.05 in Goldfeld Quandt Test, we reject the null hypothesis meaning that the error terms are not homoscedastic.

Assumption 2 does not hold.

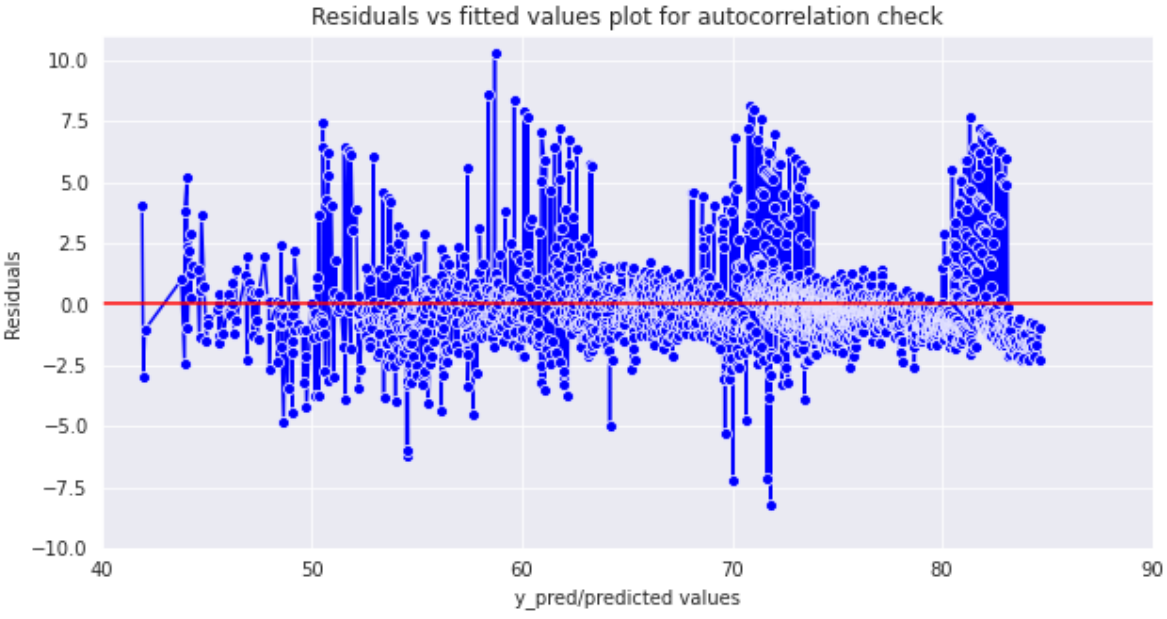
3. **Normality of residuals:**



The residual terms are pretty much normally distributed for the number of test points we took. This claim is due to the central limit theorem.
Assumption 3 holds.

4. **Autocorrelation of residuals:**

There should not be auto-correlation in the data so the error terms should not form any pattern:

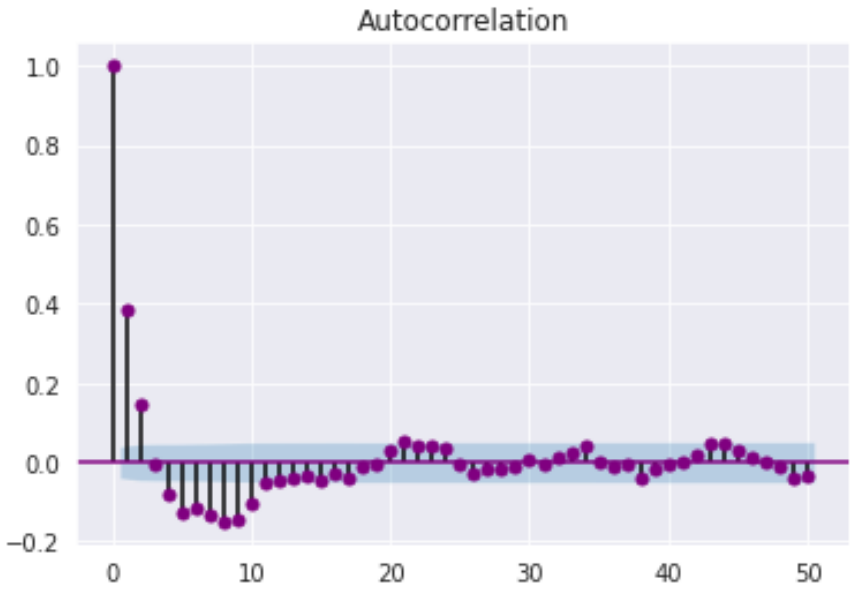


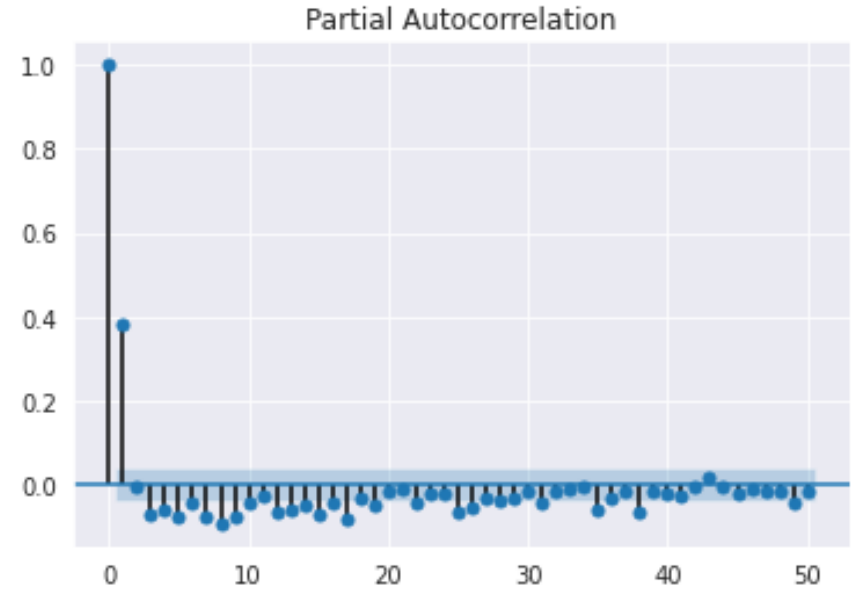
To ensure the absence of autocorrelation we used Ljungbox test with a significance (alpha) value of 0.05.

Null Hypothesis: Autocorrelation is absent.
Alternative Hypothesis: Autocorrelation is present.

p-value: 6.9067510477617954e-133

Since the p-value is less than 0.05 we will reject the null hypothesis that error terms are not autocorrelated.





The results show signs of autocorelation since there are spikes outside the blue confidence interval region.

Assumption 4 does not hold.

Alternative Models:

Since not all of the assumptions for the simple linear regression hold, we assume that there are other data-cleaning steps that can be done in order to fit it into the model.
We also tried other models to optimize the results.
Eventually we gained the best results using Kernel-Ridge Regression:

- **Lasso**
  Using alpha value of 0.001 and standard scaler.

Lasso score: 0.9672037104326425

- **Ridge**
  Using alpha value of 0.01 and standard scaler.

Ridge score: 0.9673520294521117

- **Kernel Regression**
  Using Polynomial kernel with degree 2, an alpha value of 0.04 and standard scaler.

Kernel-Regression score: 0.984629312352697

predictions_final2.csv

After a long trial and error of the hyperparameters optimization, we achieved our best score using the kernel-ridge model with a Laplacian kernel and an alpha value of 0.005.
We combined the kernel-ridge model into a pipeline with a standard scaler and a polynomial feature with degree 2.

- **Our best results:**

Optimized Kernel-Ridge score: 0.9921556306459366

predictions_final.csv