A Machine Learning Model using Flask

Alireza Ehiaei

26-June-2021

Table Of Contents

Introduction

In this research, a machine learning model is used to predict the export value of a country based on the variables; year, the number of the labor force, and manufacturing as a percentage of GDP of the country.

We will have a look at the variables, then introduce the method used to create the model. Finally, in order to make it possible for users to use this machine learning model, a web app using Flask is built and the model is deployed using it.

The goal of this research is familiarizing with machine learning algorithm and Flask, not an economic analysis.

Introduction

Exports of goods and services (current US$) [1]

- Exports of goods and services comprise all transactions between residents of a country and the rest of the world involving a change of ownership from residents to nonresidents of general merchandise, net exports of goods under merchanting, nonmonetary gold, and services. Data are in current U.S. dollars.

1 -https://www.indexmundi.com/facts/indicators/BX.GSR.GNFS.CD

Introduction

Manufacturing, value added (% of GDP) [1]

- Manufacturing refers to industries belonging to ISIC divisions 15-37. Value added is the net output of a sector after adding up all outputs and subtracting intermediate inputs. It is calculated without making deductions for depreciation of fabricated assets or depletion and degradation of natural resources. The origin of value added is determined by the International Standard Industrial Classification (ISIC), revision 3. Note: For VAB countries, gross value added at factor cost is used as the denominator.

1 -https://datacatalog.worldbank.org/manufacturing-value-added-gdp

Introduction

Labor force, total [1]

- Labor force comprises people ages 15 and older who supply labor for the production of goods and services during a specified period. It includes people who are currently employed and people who are unemployed but seeking work as well as first-time job-seekers. Not everyone who works is included, however. Unpaid workers, family workers, and students are often omitted, and some countries do not count members of the armed forces. Labor force size tends to vary during the year as seasonal workers enter and leave.

1 -https://datacatalog.worldbank.org/labor-force-total-4

Introduction

Machine learning [1]

In general, a learning problem considers a set of n samples of data and then tries to predict properties of unknown data. This problem can be either:

1. supervised learning, in which the data comes with additional attributes that the goal is predicting them, including the prediction models of:

- **classification**: samples belong two or more classes and we want to learn from already labeled data how to predict the class of unlabeled data.
- **regression**: if the desired output consists of one or more continuous variables, then the task is called regression.

1 - https://scikit-learn.org/stable/tutorial/basic/tutorial.html

Introduction

Machine learning

2. nsupervised learning, in which the training data consists of a set of input vectors x without any corresponding target values and the goal in such problems may be:

- **Clustering**: discovering groups of similar examples within the data.
- **Density estimation**: determining the distribution of data within the input space, or to project the data from a high-dimensional space down to two or three dimensions for the purpose of visualization.

- In this research a supervised machine learning algorithm is used.

Dataset

- All three variables including labor force, manufacturing, value added (% of GDP), and exports of goods and services are from 266 countries or regions around the world that their data from 1990 to 2018 was available.
- The data source is https://data.worldbank.org/indicator.
- Data and code are uploaded at https://github.com/Alireza-Ehiaei/Data_Sciences/tree/main/Machine_Learning/Export_value_estimator.

Process

First, we have a block of codes for downloading and preparing data, then a block of codes for the machine learning model, and finally a block of codes for the Flask app. All parts are written in python jupyter notebook that should be run respectively from top to down, so that at the end web app will be executed in a new tab in order to do a prediction.

Preparing data

- Since the data downloaded in separated csv files have information of all years for each country in one row and we need to combine files so that the file would be a time series in which each row has information of the three variables, for a specified country at a specified year, a function (conv) to convert data from their initial format into the desired format is defined. For example, initial format of export data is like:

	Country Name	Country Code	Indicator Name	Indicator Code	1960	1961	1962	1963	1964	1965	...	2011
0	Aruba	ABW	Exports of goods and services (current US$)	NE.EXP.GNFS.CD	NaN	NaN	NaN	NaN	NaN	NaN	...	1.729050e+09
1	Africa Eastern and Southern	AFE	Exports of goods and services (current US$)	NE.EXP.GNFS.CD	NaN	NaN	NaN	NaN	NaN	NaN	...	2.962620e+11
2	Afghanistan	AFG	Exports of goods and services (current US$)	NE.EXP.GNFS.CD	NaN	NaN	2.666668e+07	6.888892e+07	7.111114e+07	1.133333e+08	...	NaN

Preparing data

- After using conv function for each data set and combining data we have our own prepared data format to use in machine learning model as shown below, each row includes the information of a country but the name of country is delete since it is not needed in creating the model.

	year	Labor force, total	Manufacturing, value added (% of GDP)	Exports of goods and services (current US$)
0	1990.0	120457957.0	16.913422	5.044359e+10
1	1991.0	124014761.0	15.832989	4.924190e+10
2	1992.0	127772120.0	15.810120	5.135789e+10
3	1993.0	131626739.0	14.860628	5.391123e+10
4	1994.0	135601191.0	14.314819	5.718522e+10

Machine learning model

In this research a supervised machine learning algorithm called LinearRegression from Scikit-learn is used to create a model predicting the targeted variable. Scikit-learn is a free software machine learning library for the Python programming language.

First, all columns except the last one are assigned to a dataframe called X as input variables, and the last column as y variable that here is the export variable. Then the data by a function called ‘train_test_split’ is split into training data (here 80% of data) and testing data (rest of the data). The training data is the data that the model will learn from and the testing data is used to see the accuracy of the model performance on unseen data.

Machine learning model

Then by function ‘Fit’ the linear regression algorithm is used to train and build the model based on X and y data frames. Finally, the model is saved in order to be called in Flask app using Pickle function.

```
#split dataset into train and test data
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_s

# Building and training the model (linear)
from sklearn import linear_model
from sklearn.metrics import mean_squared_error, r2_score

regressor  = linear_model.LinearRegression()
regressor.fit(X_train, y_train)
y_pred = regressor.predict(X_test)

# Saving model to disk to use later in Flask app
pickle.dump(regressor , open('model.pkl','wb'))
```

Machine learning model

Now the model is built and we can have a looak at the coefficients, intercept and accuracy of the model that here is 0.66%, and test it for estimating the export value of a country. For example, with 6000000 labor force, manufacturing 10% of its GDP at 2025 the export value is estimated 4195527:

```
    # example: loading model to compare the results here
    model = pickle.load(open('model.pkl','rb'))
    print('Estimation of export value is :', model.predict([[ 2025, 6000000, 10 ]]))

    # print model features
    print('Coefficients:', regressor.coef_)
    print('Intercept:', regressor.intercept_)
    print('Mean squared error (MSE): %.2f'
          % mean_squared_error(y_test, y_pred))
    print('Coefficient of determination (R^2): %.2f'
          % r2_score(y_test, y_pred))
    return(regressor.coef_)
```

```
ML_linear(data_merged)
```

```
stimation of export value is : [4195527.64583124]
oefficients: [47.51789574  0.68326705  0.15612796]
ntercept: -299.93389151821793
ean squared error (MSE): 927507.63
oefficient of determination (R^2): 0.66
```

Deployment the model on Flask

Flask is the prototype used to create instances of web application. After importing Flask, we created an instance of the *Flask* class for our web app. Then we call the ML model built in previous part by using pickle function. Then the *home* function will run and it will return its output on the webpage. Home function executes codes written in index.html file that is saved in folder named templates inside the folder that the code of Flask code existed in.

```
import numpy as np
from flask import Flask, request, jsonify, render_template
import pickle

app1 = Flask(__name__,template_folder='template')
model = pickle.load(open('model.pkl', 'rb'))

@app1.route('/')
def home():
    return render_template('index.html')
```

Deployment the model on Flask

index.html handles the text that should be written in a webpage and the structure of getting the user inputs and showing the result. For example, by executing the app the headers and text ‘This is a machine learning estimator…’ and three blank boxes to get user inputs and the button of prediction will be shown. Here is the related part of index.html file:

```
<h1>About app</h1>
<p>This is a machine learning estimator using linear regression model is used to predict the export value of a country based on the variables; year, t

<p> For estimation export value of a country please enter the information below and see the result. for example; number of labor force 12000000, manu
</p><br>

<!-- Main Input For Receiving Query to our ML -->
<form action="{{ url_for('predict')}}"method="post">
    <input type="text" name="Labor force" placeholder="Labor force" required="required" /><br><br>
    <input type="text" name="Manufacturing,(% of GDP)" placeholder="Manufacturing,(% of GDP)" required="required" /><br><br>
    <input type="text" name="year" placeholder="year" required="required" /><br><br>


    <button type="submit" class="btn btn-primary btn-block btn-large">Predict</button>
</form>
```

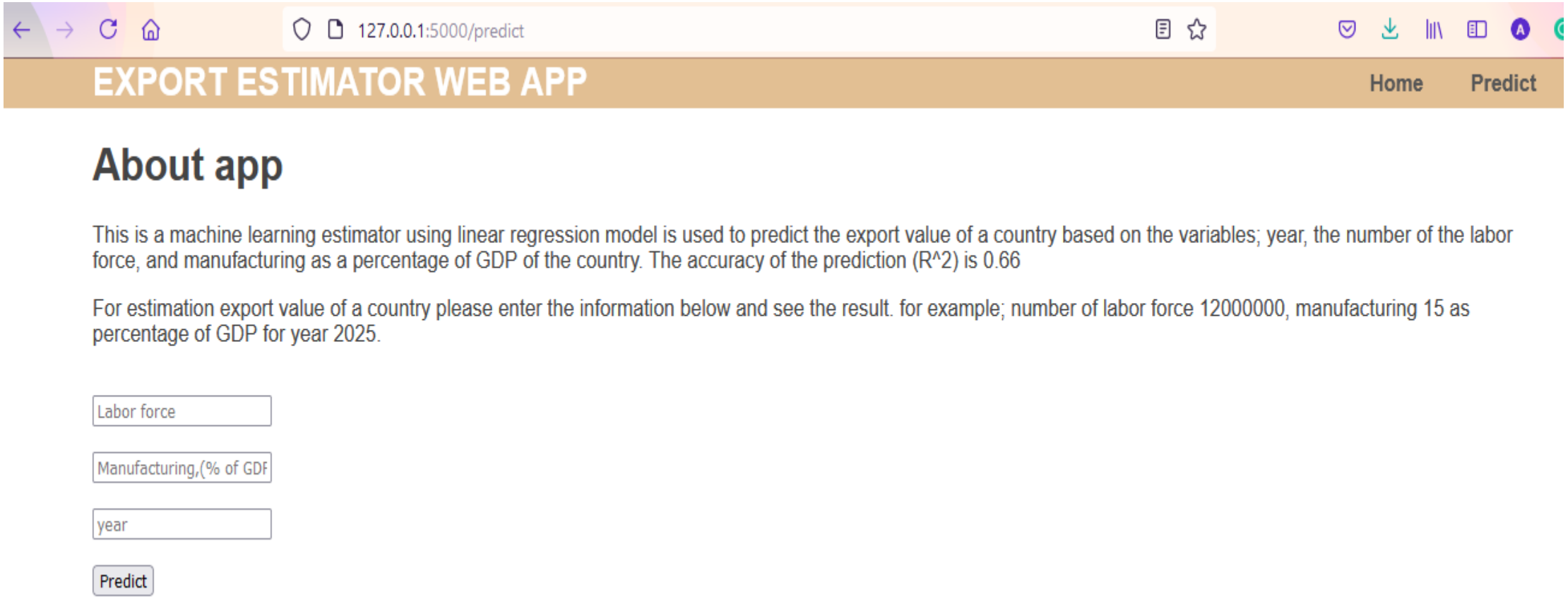
Deployment the model on Flask

Now, the predict function, by using the POST method, receives input data sent by users, and stores them in the array named int_features as float values, and maps the machine learning model on inputs, and sends the result into the index.HTML function to be shown as the estimation of the targeted variable.

```
@app1.route('/predict',methods=['POST'])
def predict():
    '''
    For rendering results on HTML GUI
    '''
    int_features = [float(x) for x in request.form.values()]
    final_features = np.array([[int_features[2], int_features[0], int_features[1]]])
    prediction = model.predict(final_features)

    output = round(prediction[0], 2)

    return render_template('index.html', prediction_text='Estimation of exports of goods and services \
                           for your inputs is (US$) {}'.format(output))

if __name__ == "__main__":
    app1.run(debug=True, use_reloader=False)
```

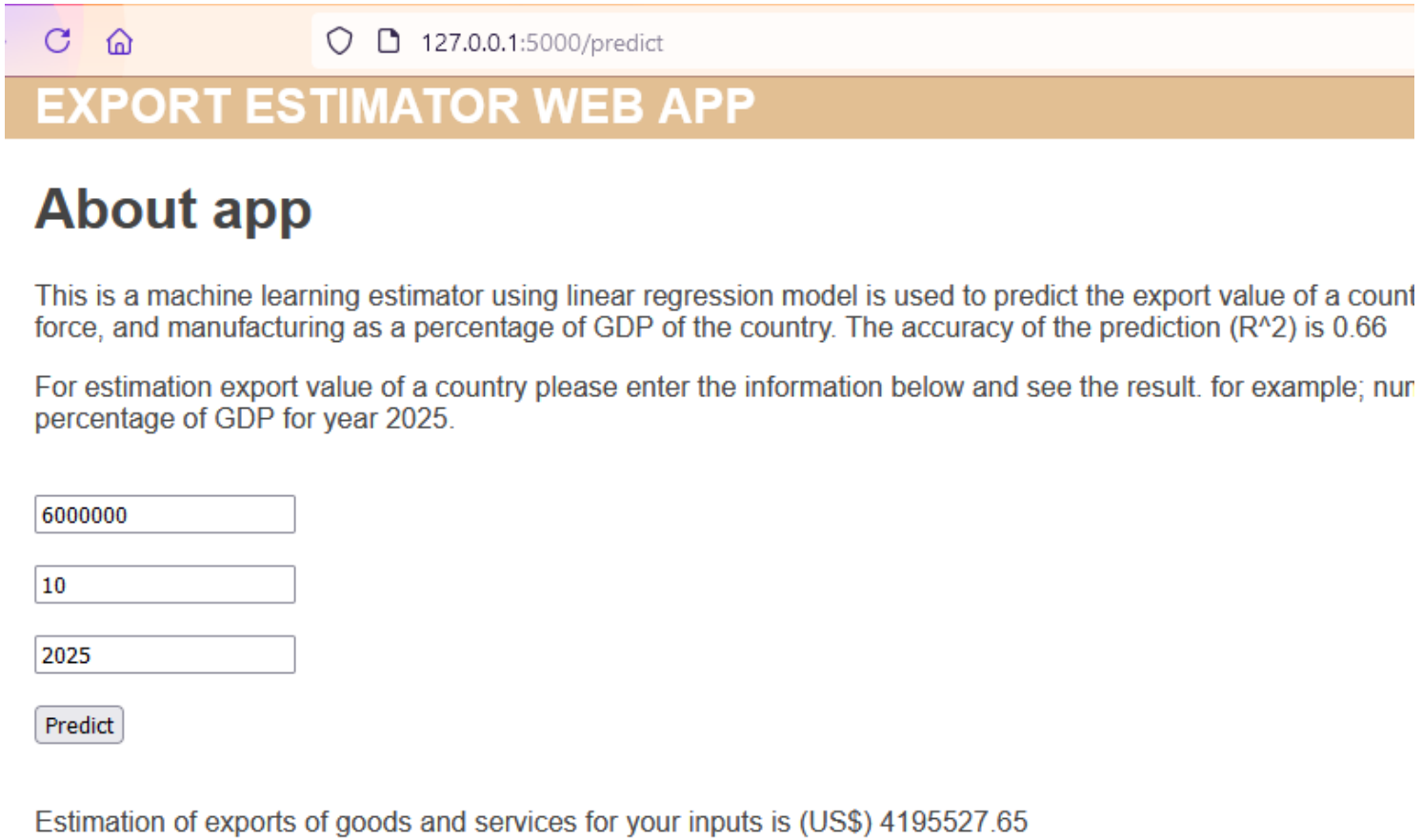
Deployment the model on Flask

The web app looks like this:



Deployment the model on Flask

For example, if user inputs the above information of a hypothetical country with 6000000 labor force, manufacturing 10% of its GDP at 1990, the same result shown as the result of the machine learning model itself will be shown and the export value will be estimated at 4195527:



Thank You