**INTRODUCTION**

 1.1  Overview

This is an ML project for predicting life expectancy of a person using provided information. Predicting life expectancy is a difficult task, Researchers have to consider hundreds of parameters like country, age, gender, physical health, psychological health, financial condition, governmental schemes, death rate, possible risk factors for well being of that individual and other specific factors varying from person to person. Larger the size of data difficult it is to run analysis over it using human brains, especially if the task is complex and repetitive. So, the basic idea behind this project is to construct a data set with all the available data we have about current life expectancy of the countries and all the major factors affecting it, and create an ML model for life expectancy prediction.

 1.2  Purpose

Life expectancy is crucial while determining Medical plans for some geographical region, Or Calculating deadliness of a disease/ pandemic/ epidemic, Or Mediclaim policies/Life insurance. Life expectancy says a lot about the health factors of a particular Area, community, family, supporting healthcare system. By programming and deploying this model we take a step towards widening the list of affecting parameters, help digitizing the medical analytic system and automating the complex procedure of life expectancy prediction.

**LITERATURE SURVEY**

 2.1  Existing problem

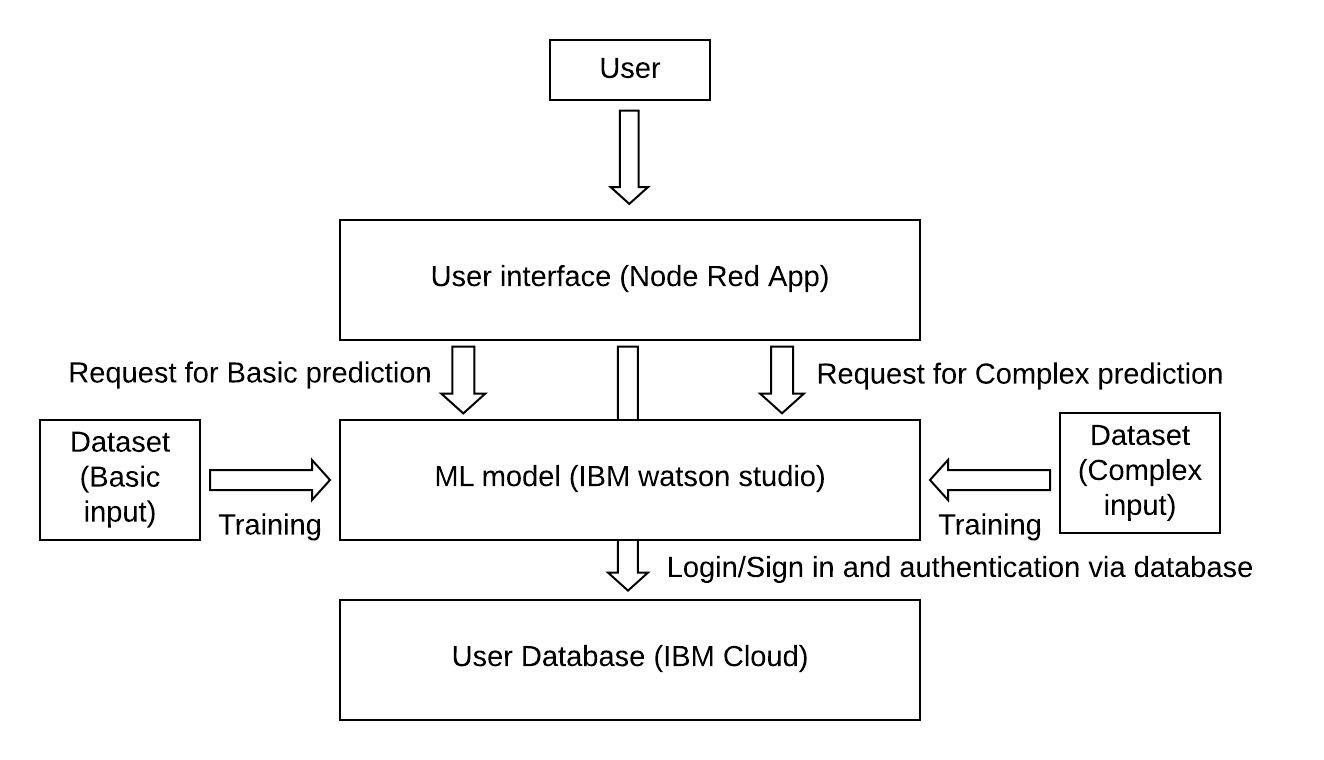
1. Data discontinuity - Prediction of Life expectancy is highly data dependent. The number of factors affecting is huge and thus the data set will be vast. Number of discrepancies increase with the size of the dataset.
2. Varying parameters - Various parameters involved in data are subjected to variation over time. The parameters like Mortality rate, birth rate, GDP, population, etc. does not only change from country to country but also changes from year to year.

 2.2  Proposed solution

1. Through the step of data processing various data discrepancies are removed.
2. Where attribute value is null or undefined, most expected value is put in that place.
3. The mathematical functions like mean, mode and median are used to substitute missing values.
4. Some rows with null and undefined values are dropped so prediction could be more robust.
5. If one column varies proportionally with another then their relation is used to find out likely values in the second column. Linear regression is used in such cases. It requires at least two columns.
6. Algorithms like Random forest and k nearest nodes are also used if there is no relation between columns. These algorithms require only one column(one with null values) to fill in missing values.

**THEORETICAL ANALYSIS**

Block diagram

Hardware / Software designing

1. User Interface:- This part of application is purely designed for User to carry out usage. The UI is designed using Node red app deployed through IBM cloud. UI will have following functions-
2. Sign In page - New User will have to register here with his email id and customized password. This data will be stored in Cloudant user database on IBM cloud.
3. Login page - When an existing user wants to use the website again he has to enter his existing email id and password. Both of them are verified from Cloudant so that user requests can be granted.
4. Basic Prediction - This button takes users to the Basic Life Expectancy prediction form where less number of parameters are required for prediction. When these inputs are delivered to the ML model the prediction is made on it and returned through http request/response.
5. Complex Prediction - This button takes users to the complex life expectancy prediction form where the number of parameters is vast so that predictions made could be more accurate. These predictions are also posted through http request/response.

     2)  ML Model - The Machine learning models are used for making predictions about Life expectancy. It is mentioned in the UI that there are two models involved in application both work on different data inputs. Where one data input is a subset of another.

1. Basic model - This model is trained using a limited part of the complete dataset using Watson Auto AI experiment. As the original dataset is very detailed and thus the model based on requires approx. 15 parameters, this model is simplified to 6 input parameters.
2. Complex model - This model uses complete dataset to make predictions and thus is very accurate as compared to the previous model.

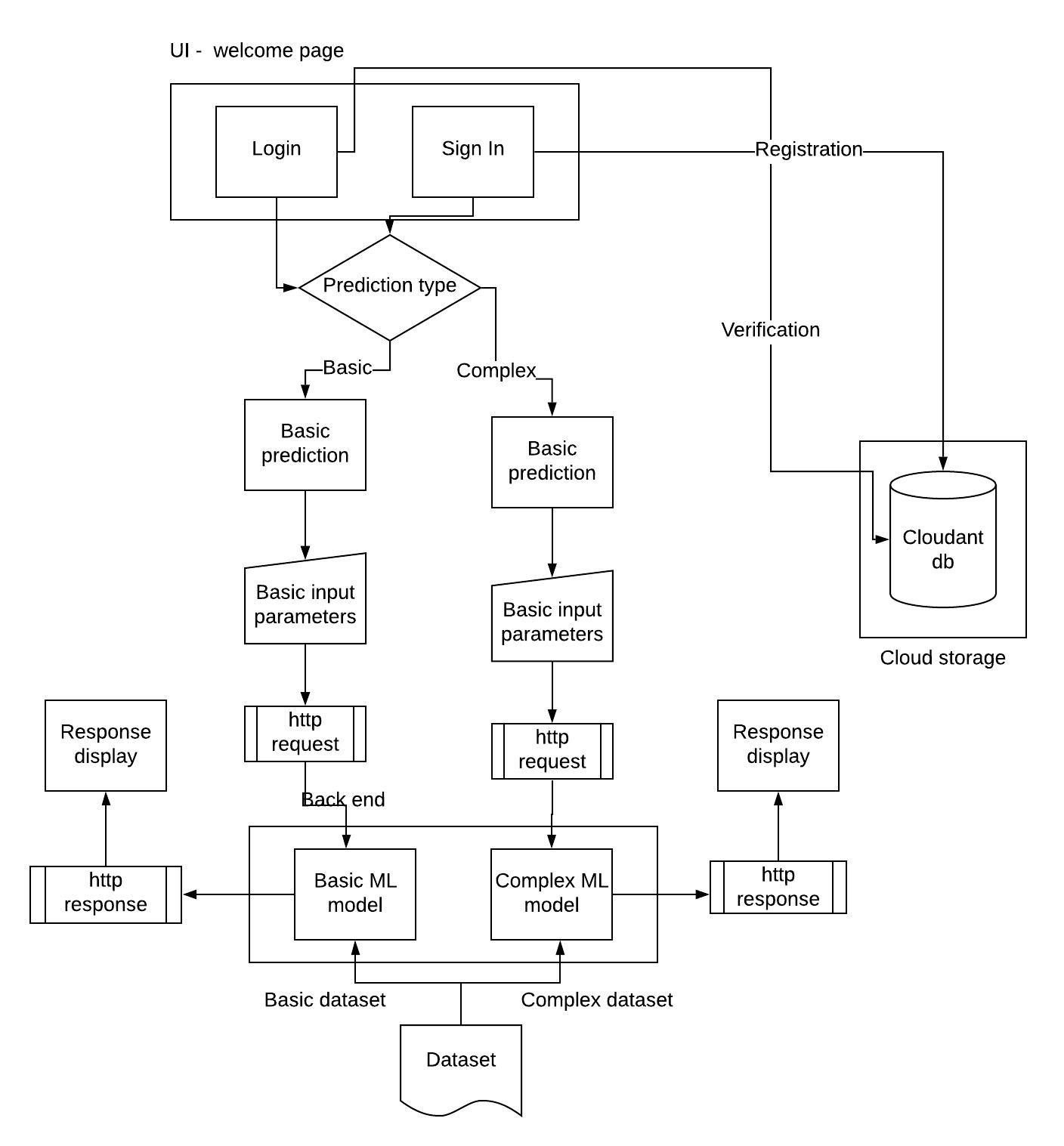
    3) User Database - This layer is used to store the login details of the users like email id and password. Cloudant database stores all the data provided by the user in the form of a json document and retrieves it during the login verification process every time.

    4) Training datasets - This is the dataset used for the training of ML models.

**EXPERIMENTAL INVESTIGATIONS**

1. Dataset - This dataset was obtained from kaggle websites sample datasets for machine learning experiments.
2. Resources -
3. IBM cloud account - This account allows access to all the required services for application.
4. IBM Watson studio - This IBM service provides machine learning, Watson assistant , AI, Auto AI services for construction of the project.
5. Auto AI - This service was used to create the ML model based on the required database.
6. Node red app - This application was used to create UI. Connecting UI with ML models and Cloudant database was also done using Node red app.
7. Cloudant db - This database service provided by IBM cloud was used to store user login information in json format

**FLOWCHART**



**RESULT**

1. This application enabled users to predict his/her life expectancy in two different forms: basic and complex.
2. Basic form requires 5 - 6 inputs, whereas complex form requires 15 inputs.
3. Accuracy of complex prediction is more than basic prediction.

**ADVANTAGES & DISADVANTAGES**

1. Advantages - Involving cloud services reduced the programming stack to its minimum. Node red, Cloudant and Auto AI eased the tough part of ML model construction, UI integration with backend, database management.
2. Disadvantages - When a new dataset arrives, a new model must be constructed in order to make a valid prediction. Although basic model predictions are easy to get, their accuracy is compromised due to redundant data.

**APPLICATIONS**

Personal use - If a person is curious to find out his/her life expectancy then he/she can get results by providing data. This application can only be used for fun purposes by curious individuals.

**CONCLUSION**

This project deals with prediction of average life expectancy of an individual. Generally, life expectancy is determined by the country in which a person lives, Geographical conditions, Mortality rate, Medical facilities of the region. This project will consider all of the above mentioned factors and will construct a ML model which will predict the life expectancy of an individual.

**FUTURE SCOPE**

This project can be improved by adding more personalizing parameters like age,medical condition of an individual. By doing so we can enable this application’s use for Corporate companies, small medical facilities.

**BIBLIOGRAPHY**

1. <https://www.allbusinesstemplates.com/download/?filecode=2KBA4&lang=en&iuid=9f9faa69-9fab-40ee-8457-ea0e5df8c8de>
2. <https://www.youtube.com/watch?v=LOCkV-mENq8&feature=youtu.be>
3. <https://github.com/>
4. <https://www.youtube.com/watch?v=w3jLJU7DT5E&feature=youtu.be>
5. <https://www.zoho.com/writer/help/working-with-text.html>
6. <https://my15.digitalexperience.ibm.com/b73a5759-c6a6-4033-ab6b-d9d4f9a6d65b/dxsites/151914d1-03d2-48fe-97d9-d21166848e65/>
7. <https://www.ibm.com/cloud/get-started>
8. <https://developer.ibm.com/tutorials/how-to-create-a-node-red-starter-application/>
9. <https://github.com/watson-developer-cloud/node-red-labs>
10. <https://www.youtube.com/watch?v=s7wmiS2mSXY&feature=youtu.be>
11. <https://www.w3schools.com/howto/howto_make_a_website.asp>
12. <https://developer.ibm.com/technologies/machine-learning/series/learning-path-machine-learning-for-developers/>
13. <https://www.youtube.com/watch?v=NmdjtezQMSM>
14. <https://bookdown.org/caoying4work/watsonstudio-workshop/jn.html>
15. <https://developer.ibm.com/tutorials/watson-studio-auto-ai/>
16. <https://www.youtube.com/watch?v=IDKCmC1fCiU>
17. <https://www.kaggle.com/kumarajarshi/life-expectancy-who>
18. <https://www.youtube.com/watch?v=Jtej3Y6uUng>
19. <https://bookdown.org/caoying4work/watsonstudio-workshop/jn.html#deploy-model-as-web-service>

**APPENDIX**

**A. Source code**

**Node red flow**

[

    {

        "id": "816e7f0a.56a38",

        "type": "tab",

        "label": "Main website",

        "disabled": false,

        "info": ""

    },

    {

        "id": "85950ec1.e5d64",

        "type": "ui\_button",

        "z": "816e7f0a.56a38",

        "name": "",

        "group": "149f63d2.98e83c",

        "order": 5,

        "width": 0,

        "height": 0,

        "passthru": false,

        "label": "Sign In",

        "tooltip": "",

        "color": "",

        "bgcolor": "",

        "icon": "",

        "payload": "true",

        "payloadType": "bool",

        "topic": "",

        "x": 120,

        "y": 440,

        "wires": [

            [

                "4587432f.b2009c"

            ]

        ]

    },

    {

        "id": "c381bc51.d3ae1",

        "type": "ui\_ui\_control",

        "z": "816e7f0a.56a38",

        "name": "",

        "events": "all",

        "x": 400,

        "y": 440,

        "wires": [

            []

        ]

    },

    {

        "id": "4587432f.b2009c",

        "type": "function",

        "z": "816e7f0a.56a38",

        "name": "",

        "func": "msg.payload = {\n    'tab': 'Sign In'\n}\nreturn msg;",

        "outputs": 1,

        "noerr": 0,

        "x": 260,

        "y": 440,

        "wires": [

            [

                "c381bc51.d3ae1"

            ]

        ]

    },

    {

        "id": "52780d4a.7a1a34",

        "type": "ui\_button",

        "z": "816e7f0a.56a38",

        "name": "",

        "group": "149f63d2.98e83c",

        "order": 0,

        "width": 0,

        "height": 0,

        "passthru": false,

        "label": "Log In",

        "tooltip": "",

        "color": "",

        "bgcolor": "",

        "icon": "",

        "payload": "true",

        "payloadType": "bool",

        "topic": "",

        "x": 120,

        "y": 500,

        "wires": [

            [

                "53c6d297.4ae62c"

            ]

        ]

    },

    {

        "id": "53c6d297.4ae62c",

        "type": "function",

        "z": "816e7f0a.56a38",

        "name": "",

        "func": "msg.payload = {\n    'tab':'Log in'\n}\nreturn msg;",

        "outputs": 1,

        "noerr": 0,

        "x": 280,

        "y": 500,

        "wires": [

            [

                "bdee6d90.b6bd2"

            ]

        ]

    },

    {

        "id": "bdee6d90.b6bd2",

        "type": "ui\_ui\_control",

        "z": "816e7f0a.56a38",

        "name": "",

        "events": "all",

        "x": 440,

        "y": 500,

        "wires": [

            []

        ]

    },

    {

        "id": "95d2ea03.23f6b8",

        "type": "ui\_form",

        "z": "816e7f0a.56a38",

        "name": "",

        "label": "",

        "group": "7787cf17.7454a",

        "order": 0,

        "width": 0,

        "height": 0,

        "options": [

            {

                "label": "Your Name",

                "value": "name",

                "type": "text",

                "required": true,

                "rows": null

            },

            {

                "label": "Your Organization",

                "value": "or\_name",

                "type": "text",

                "required": true,

                "rows": null

            },

            {

                "label": "Your Email",

                "value": "email",

                "type": "email",

                "required": true,

                "rows": null

            },

            {

                "label": "Your Password",

                "value": "pwd",

                "type": "password",

                "required": true,

                "rows": null

            }

        ],

        "formValue": {

            "name": "",

            "or\_name": "",

            "email": "",

            "pwd": ""

        },

        "payload": "",

        "submit": "submit",

        "cancel": "cancel",

        "topic": "",

        "x": 170,

        "y": 600,

        "wires": [

            [

                "17a2096c.814137"

            ]

        ]

    },

    {

        "id": "37bcc3f5.45bf4c",

        "type": "cloudant out",

        "z": "816e7f0a.56a38",

        "name": "",

        "cloudant": "",

        "database": "members",

        "service": "node-red-euipf-cloudant-1590482242688-84752",

        "payonly": true,

        "operation": "insert",

        "x": 480,

        "y": 660,

        "wires": []

    },

    {

        "id": "17a2096c.814137",

        "type": "function",

        "z": "816e7f0a.56a38",

        "name": "",

        "func": "msg.payload = {\n    \"\_id\":msg.payload.email,\n    \"email\":msg.payload.email,\n    \"pwd\":msg.payload.pwd,\n    \"name\":msg.payload.name,\n    \"or\_name\":msg.payload.or\_name\n}\nreturn msg;",

        "outputs": 1,

        "noerr": 0,

        "x": 300,

        "y": 600,

        "wires": [

            [

                "37bcc3f5.45bf4c",

                "d549c49c.263998"

            ]

        ]

    },

    {

        "id": "848ff149.9dc7c",

        "type": "comment",

        "z": "816e7f0a.56a38",

        "name": "sign in page",

        "info": "",

        "x": 120,

        "y": 560,

        "wires": []

    },

    {

        "id": "a25341fc.88092",

        "type": "ui\_ui\_control",

        "z": "816e7f0a.56a38",

        "name": "",

        "events": "change",

        "x": 580,

        "y": 600,

        "wires": [

            []

        ]

    },

    {

        "id": "3ae70881.c0a208",

        "type": "ui\_text",

        "z": "816e7f0a.56a38",

        "group": "6d92483e.53d708",

        "order": 1,

        "width": 0,

        "height": 0,

        "name": "",

        "label": "",

        "format": "{{msg.payload}}",

        "layout": "col-center",

        "x": 330,

        "y": 1220,

        "wires": []

    },

    {

        "id": "d549c49c.263998",

        "type": "function",

        "z": "816e7f0a.56a38",

        "name": "",

        "func": "msg.payload={\n    'tab':'Select Input Complexity'\n}\nreturn msg;",

        "outputs": 1,

        "noerr": 0,

        "x": 450,

        "y": 600,

        "wires": [

            [

                "a25341fc.88092"

            ]

        ]

    },

    {

        "id": "c8d4ec30.beea5",

        "type": "ui\_form",

        "z": "816e7f0a.56a38",

        "name": "",

        "label": "",

        "group": "acc05384.74e84",

        "order": 1,

        "width": 0,

        "height": 0,

        "options": [

            {

                "label": "Enter Your Email ",

                "value": "email",

                "type": "email",

                "required": true,

                "rows": null

            },

            {

                "label": "Enter Your Password",

                "value": "pwd",

                "type": "password",

                "required": true,

                "rows": null

            }

        ],

        "formValue": {

            "email": "",

            "pwd": ""

        },

        "payload": "",

        "submit": "submit",

        "cancel": "cancel",

        "topic": "",

        "x": 90,

        "y": 780,

        "wires": [

            [

                "f4de92c3.c5f9"

            ]

        ]

    },

    {

        "id": "5dd75fa9.59e1",

        "type": "cloudant in",

        "z": "816e7f0a.56a38",

        "name": "",

        "cloudant": "",

        "database": "members",

        "service": "node-red-euipf-cloudant-1590482242688-84752",

        "search": "\_id\_",

        "design": "",

        "index": "",

        "x": 500,

        "y": 780,

        "wires": [

            [

                "7d910ea5.7df31"

            ]

        ]

    },

    {

        "id": "f4de92c3.c5f9",

        "type": "function",

        "z": "816e7f0a.56a38",

        "name": "",

        "func": "global.set(\"email\",msg.payload.email)\nglobal.set(\"pwd\",msg.payload.pwd)\nreturn msg;",

        "outputs": 1,

        "noerr": 0,

        "x": 230,

        "y": 780,

        "wires": [

            [

                "ce5b88ce.d5c6a8"

            ]

        ]

    },

    {

        "id": "ce5b88ce.d5c6a8",

        "type": "function",

        "z": "816e7f0a.56a38",

        "name": "",

        "func": "var email = global.get('email')\nmsg.payload = {\n    \"\_id\" : email\n}\nreturn msg;",

        "outputs": 1,

        "noerr": 0,

        "x": 350,

        "y": 780,

        "wires": [

            [

                "5dd75fa9.59e1"

            ]

        ]

    },

    {

        "id": "7d910ea5.7df31",

        "type": "function",

        "z": "816e7f0a.56a38",

        "name": "",

        "func": "var pwd = global.get('pwd')\nif(pwd == msg.payload.pwd)\n    msg.payload = {'tab':'Select Input Complexity'}\nelse\n    msg.payload = 'Login failed'\nreturn msg;",

        "outputs": 1,

        "noerr": 0,

        "x": 650,

        "y": 780,

        "wires": [

            [

                "4b6323af.3efe3c",

                "bffc61c9.db00a"

            ]

        ]

    },

    {

        "id": "4b6323af.3efe3c",

        "type": "ui\_text",

        "z": "816e7f0a.56a38",

        "group": "acc05384.74e84",

        "order": 2,

        "width": 0,

        "height": 0,

        "name": "",

        "label": "",

        "format": "",

        "layout": "row-spread",

        "x": 810,

        "y": 840,

        "wires": []

    },

    {

        "id": "30a3aa41.98ced6",

        "type": "comment",

        "z": "816e7f0a.56a38",

        "name": "log in page",

        "info": "",

        "x": 100,

        "y": 720,

        "wires": []

    },

    {

        "id": "bffc61c9.db00a",

        "type": "ui\_ui\_control",

        "z": "816e7f0a.56a38",

        "name": "",

        "events": "all",

        "x": 800,

        "y": 780,

        "wires": [

            []

        ]

    },

    {

        "id": "e092494b.b9a498",

        "type": "comment",

        "z": "816e7f0a.56a38",

        "name": "Calculate life expectancy",

        "info": "Country,Year,Status,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\n\nCountry\nYear\nStatus",

        "x": 230,

        "y": 920,

        "wires": []

    },

    {

        "id": "c16b537e.f819b",

        "type": "ui\_form",

        "z": "816e7f0a.56a38",

        "name": "",

        "label": "Input data",

        "group": "a6ab71a7.9d05b",

        "order": 2,

        "width": "0",

        "height": 0,

        "options": [

            {

                "label": "Your Conuntry",

                "value": "Country",

                "type": "text",

                "required": true,

                "rows": null

            },

            {

                "label": "Current Year",

                "value": "Year",

                "type": "number",

                "required": true,

                "rows": null

            },

            {

                "label": "Select Country Status",

                "value": "Status",

                "type": "text",

                "required": true,

                "rows": null

            },

            {

                "label": "Adult Mortality Rate",

                "value": "Adult\_Mortality",

                "type": "number",

                "required": true,

                "rows": null

            },

            {

                "label": "Infant deaths",

                "value": "infant\_deaths",

                "type": "number",

                "required": true,

                "rows": null

            },

            {

                "label": "Avg. Alcohol Consumption",

                "value": "Alcohol",

                "type": "number",

                "required": true,

                "rows": null

            },

            {

                "label": "Percentage Expenditure",

                "value": "percentage\_expenditure",

                "type": "number",

                "required": true,

                "rows": null

            },

            {

                "label": "Country's Hepatitis B rate",

                "value": "Hepatitis\_B",

                "type": "number",

                "required": true,

                "rows": null

            },

            {

                "label": "Country's Measles Rate",

                "value": "Measles",

                "type": "number",

                "required": true,

                "rows": null

            },

            {

                "label": "BMI",

                "value": "BMI",

                "type": "number",

                "required": true,

                "rows": null

            },

            {

                "label": "Under-five death Rate",

                "value": "under\_five\_deaths",

                "type": "number",

                "required": true,

                "rows": null

            },

            {

                "label": "Polio Rates",

                "value": "Polio",

                "type": "number",

                "required": true,

                "rows": null

            },

            {

                "label": "Total Expenditure",

                "value": "Total\_expenditure",

                "type": "number",

                "required": true,

                "rows": null

            },

            {

                "label": "HIV/AIDS Rate",

                "value": "HIV\_AIDS",

                "type": "number",

                "required": true,

                "rows": null

            },

            {

                "label": "Country's GDP",

                "value": "GDP",

                "type": "number",

                "required": true,

                "rows": null

            },

            {

                "label": "Diphtheria Rate",

                "value": "Diphtheria",

                "type": "number",

                "required": true,

                "rows": null

            },

            {

                "label": "Population",

                "value": "Population",

                "type": "number",

                "required": true,

                "rows": null

            },

            {

                "label": "thinness (1-19) years rate",

                "value": "thinnes \_1\_19\_years",

                "type": "number",

                "required": true,

                "rows": null

            },

            {

                "label": "thinnes (5-9) years",

                "value": "thinness\_5\_9\_years",

                "type": "number",

                "required": true,

                "rows": null

            },

            {

                "label": "Income composition of resources",

                "value": "Income\_composition\_of\_resources",

                "type": "number",

                "required": true,

                "rows": null

            },

            {

                "label": "Schooling",

                "value": "Schooling",

                "type": "number",

                "required": true,

                "rows": null

            }

        ],

        "formValue": {

            "Country": "",

            "Year": "",

            "Status": "",

            "Adult\_Mortality": "",

            "infant\_deaths": "",

            "Alcohol": "",

            "percentage\_expenditure": "",

            "Hepatitis\_B": "",

            "Measles": "",

            "BMI": "",

            "under\_five\_deaths": "",

            "Polio": "",

            "Total\_expenditure": "",

            "HIV\_AIDS": "",

            "GDP": "",

            "Diphtheria": "",

            "Population": "",

            "thinnes \_1\_19\_years": "",

            "thinness\_5\_9\_years": "",

            "Income\_composition\_of\_resources": "",

            "Schooling": ""

        },

        "payload": "",

        "submit": "submit",

        "cancel": "cancel",

        "topic": "",

        "x": 150,

        "y": 980,

        "wires": [

            [

                "9983db92.8d58e8"

            ]

        ]

    },

    {

        "id": "9983db92.8d58e8",

        "type": "function",

        "z": "816e7f0a.56a38",

        "name": "PreToken",

        "func": "global.set(\"Country\",msg.payload.Country)\nglobal.set(\"Year\",msg.payload.Year)\nglobal.set(\"Status\",msg.payload.Status)\nglobal.set(\"Adult\_Mortality\",msg.payload.Adult\_Mortality)\nglobal.set(\"infant\_deaths\",msg.payload.infant\_deaths)\nglobal.set(\"Alcohol\",msg.payload.Alcohol)\nglobal.set(\"percentage\_expenditure\",msg.payload.percentage\_expenditure)\nglobal.set(\"Hepatitis\_B\",msg.payload.Hepatitis\_B)\nglobal.set(\"Measles\",msg.payload.Measles)\nglobal.set(\"BMI\",msg.payload.BMI)\nglobal.set(\"under\_five\_deaths\",msg.payload.under\_five\_deaths)\nglobal.set(\"Polio\",msg.payload.Polio)\nglobal.set(\"Total\_expenditure\",msg.payload.Total\_expenditure)\nglobal.set(\"Diphtheria\",msg.payload.Diphtheria)\nglobal.set(\"HIV\_AIDS\",msg.payload.HIV\_AIDS)\nglobal.set(\"GDP\",msg.payload.GDP)\nglobal.set(\"Population\",msg.payload.Population)\nglobal.set(\"thinness\_1\_19\_years\",msg.payload.thinness\_1\_19\_years)\nglobal.set(\"thinness\_5\_9\_years\",msg.payload.thinness\_5\_9\_years)\nglobal.set(\"Income\_composition\_of\_resources\",msg.payload.Income\_composition\_of\_resources)\nglobal.set(\"Schooling\",msg.payload.Schooling)\n\nvar apikey=\"Enter API key here\";\nmsg.headers={\"content-type\":\"application/x-www-form-urlencoded\"}\nmsg.payload=msg.payload={\"grant\_type\":\"urn:ibm:params:oauth:grant-type:apikey\",\"apikey\":apikey}\nreturn msg;\n",

        "outputs": 1,

        "noerr": 0,

        "x": 300,

        "y": 980,

        "wires": [

            [

                "a1949c5b.115f7"

            ]

        ]

    },

    {

        "id": "a1949c5b.115f7",

        "type": "http request",

        "z": "816e7f0a.56a38",

        "name": "",

        "method": "POST",

        "ret": "obj",

        "paytoqs": false,

        "url": "https://iam.cloud.ibm.com/identity/token",

        "tls": "",

        "persist": false,

        "proxy": "",

        "authType": "",

        "x": 470,

        "y": 980,

        "wires": [

            [

                "ed41d3c7.e7403"

            ]

        ]

    },

    {

        "id": "5bd7f35f.230f7c",

        "type": "debug",

        "z": "816e7f0a.56a38",

        "name": "abc",

        "active": true,

        "tosidebar": true,

        "console": false,

        "tostatus": false,

        "complete": "payload",

        "targetType": "msg",

        "x": 1210,

        "y": 1300,

        "wires": []

    },

    {

        "id": "ed41d3c7.e7403",

        "type": "function",

        "z": "816e7f0a.56a38",

        "name": "Pre Prediction",

        "func": "var Country=global.get(\"Country\",msg.payload.Country)\nvar Year=global.get(\"Year\",msg.payload.Year)\nvar Status=global.get(\"Status\",msg.payload.Status)\nvar Adult\_Mortality=global.get(\"Adult\_Mortality\",msg.payload.Adult\_Mortality)\nvar infant\_deaths=global.get(\"infant\_deaths\",msg.payload.infant\_deaths)\nvar Alcohol=global.get(\"Alcohol\",msg.payload.Alcohol)\nvar percentage\_expenditure=global.get(\"percentage\_expenditure\",msg.payload.percentage\_expenditure)\nvar Hepatitis\_B=global.get(\"Hepatitis\_B\",msg.payload.Hepatitis\_B)\nvar Measles=global.get(\"Measles\",msg.payload.Measles)\nvar BMI=global.get(\"BMI\",msg.payload.BMI)\nvar under\_five\_deaths=global.get(\"under\_five\_deaths\",msg.payload.under\_five\_deaths)\nvar Polio=global.get(\"Polio\",msg.payload.Polio)\nvar Total\_expenditure=global.get(\"Total\_expenditure\",msg.payload.Total\_expenditure)\nvar Diphtheria=global.get(\"Diphtheria\",msg.payload.Diphtheria)\nvar HIV\_AIDS=global.get(\"HIV\_AIDS\",msg.payload.HIV\_AIDS)\nvar GDP=global.get(\"GDP\",msg.payload.GDP)\nvar Population=global.get(\"Population\",msg.payload.Population)\nvar thinness\_1\_19\_years=global.get(\"thinness\_1\_19\_years\",msg.payload.thinness\_1\_19\_years)\nvar thinness\_5\_9\_years=global.get(\"thinness\_5\_9\_years\",msg.payload.thinness\_5\_9\_years)\nvar Income\_composition\_of\_resources=global.get(\"Income\_composition\_of\_resources\",msg.payload.Income\_composition\_of\_resources)\nvar Schooling=global.get(\"Schooling\",msg.payload.Schooling)\n\narray=[Country,Year,Status,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]\n\nvar token=msg.payload.access\_token\nvar instance\_id=\"afb5a4cd-139e-4e89-8714-a1d7921ada59\"\nmsg.headers={'Content-Type': 'application/json',\"Authorization\":\"Bearer \"+token,\"ML-Instance-ID\":instance\_id}\nmsg.payload = {\"input\_data\" :[{\"fields\": [\"Country\", \"Year\", \"Status\", \"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\"], \"values\": [array]}]}\nreturn msg;",

        "outputs": 1,

        "noerr": 0,

        "x": 660,

        "y": 980,

        "wires": [

            [

                "19568960.688ed7"

            ]

        ],

        "info": "array=[\"Country\",\"Year\",\"Status\",\"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\"]\narray=[Country,Year,Status,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]\n"

    },

    {

        "id": "19568960.688ed7",

        "type": "http request",

        "z": "816e7f0a.56a38",

        "name": "",

        "method": "POST",

        "ret": "obj",

        "paytoqs": false,

        "url": "https://eu-gb.ml.cloud.ibm.com/v4/deployments/d2f3d213-aaab-4011-96d6-8a27b124b203/predictions",

        "tls": "",

        "persist": false,

        "proxy": "",

        "authType": "",

        "x": 850,

        "y": 980,

        "wires": [

            [

                "1e01ae8b.a7d881"

            ]

        ]

    },

    {

        "id": "1e01ae8b.a7d881",

        "type": "function",

        "z": "816e7f0a.56a38",

        "name": "",

        "func": "global.set('Expec\_c',msg.payload.predictions[0].values[0])\nmsg.payload={\n    'tab':'Life Expectancy'\n}\nreturn msg;",

        "outputs": 1,

        "noerr": 0,

        "x": 1010,

        "y": 980,

        "wires": [

            [

                "a8568e56.ed0f5"

            ]

        ]

    },

    {

        "id": "5f858608.043e68",

        "type": "debug",

        "z": "816e7f0a.56a38",

        "name": "dmnd",

        "active": true,

        "tosidebar": true,

        "console": false,

        "tostatus": false,

        "complete": "payload",

        "targetType": "msg",

        "x": 330,

        "y": 1100,

        "wires": []

    },

    {

        "id": "d9ae83c.f244d8",

        "type": "comment",

        "z": "816e7f0a.56a38",

        "name": "Result",

        "info": "",

        "x": 130,

        "y": 1080,

        "wires": []

    },

    {

        "id": "a8568e56.ed0f5",

        "type": "ui\_ui\_control",

        "z": "816e7f0a.56a38",

        "name": "",

        "events": "all",

        "x": 1140,

        "y": 980,

        "wires": [

            [

                "6989581e.8799a8"

            ]

        ]

    },

    {

        "id": "52636452.2645fc",

        "type": "function",

        "z": "816e7f0a.56a38",

        "name": "",

        "func": "var Expec\_c = global.get('Expec\_c')\nvar Expec\_b = global.get('Expec\_b')\nif(Expec\_c===undefined)\n    msg.payload= 'Your life expectancy is '+Expec\_b\nelse msg.payload= 'Your life expectancy is '+Expec\_c\nreturn msg;",

        "outputs": 1,

        "noerr": 0,

        "x": 170,

        "y": 1160,

        "wires": [

            [

                "3ae70881.c0a208",

                "bd994db0.b965c",

                "5f858608.043e68"

            ]

        ]

    },

    {

        "id": "bd994db0.b965c",

        "type": "ui\_button",

        "z": "816e7f0a.56a38",

        "name": "",

        "group": "6d92483e.53d708",

        "order": 3,

        "width": 5,

        "height": 1,

        "passthru": false,

        "label": "Make another prediction",

        "tooltip": "",

        "color": "",

        "bgcolor": "",

        "icon": "",

        "payload": "true",

        "payloadType": "bool",

        "topic": "",

        "x": 390,

        "y": 1160,

        "wires": [

            [

                "5ea3fbe2.b4ad74"

            ]

        ]

    },

    {

        "id": "2c077275.f3494e",

        "type": "ui\_ui\_control",

        "z": "816e7f0a.56a38",

        "name": "",

        "events": "all",

        "x": 720,

        "y": 1160,

        "wires": [

            []

        ]

    },

    {

        "id": "5ea3fbe2.b4ad74",

        "type": "function",

        "z": "816e7f0a.56a38",

        "name": "",

        "func": "msg.payload={\n    'tab':'Select Input Complexity'\n}\nreturn msg;",

        "outputs": 1,

        "noerr": 0,

        "x": 570,

        "y": 1160,

        "wires": [

            [

                "2c077275.f3494e"

            ]

        ]

    },

    {

        "id": "b6d55784.0efcf8",

        "type": "ui\_form",

        "z": "816e7f0a.56a38",

        "name": "",

        "label": "Input data",

        "group": "e6bbb703.a47b68",

        "order": 2,

        "width": "0",

        "height": 0,

        "options": [

            {

                "label": "Your Conuntry",

                "value": "Country",

                "type": "text",

                "required": true,

                "rows": null

            },

            {

                "label": "Current Year",

                "value": "Year",

                "type": "number",

                "required": true,

                "rows": null

            },

            {

                "label": "Select Country Status",

                "value": "Status",

                "type": "text",

                "required": true,

                "rows": null

            },

            {

                "label": "BMI",

                "value": "BMI",

                "type": "number",

                "required": true,

                "rows": null

            },

            {

                "label": "Country's GDP",

                "value": "GDP",

                "type": "number",

                "required": true,

                "rows": null

            },

            {

                "label": "Population",

                "value": "Population",

                "type": "number",

                "required": true,

                "rows": null

            }

        ],

        "formValue": {

            "Country": "",

            "Year": "",

            "Status": "",

            "BMI": "",

            "GDP": "",

            "Population": ""

        },

        "payload": "",

        "submit": "submit",

        "cancel": "cancel",

        "topic": "",

        "x": 160,

        "y": 1380,

        "wires": [

            [

                "42727d2d.1c33e4"

            ]

        ]

    },

    {

        "id": "42727d2d.1c33e4",

        "type": "function",

        "z": "816e7f0a.56a38",

        "name": "PreToken",

        "func": "global.set(\"Country\",msg.payload.Country)\nglobal.set(\"Year\",msg.payload.Year)\nglobal.set(\"Status\",msg.payload.Status)\nglobal.set(\"BMI\",msg.payload.BMI)\nglobal.set(\"GDP\",msg.payload.GDP)\nglobal.set(\"Population\",msg.payload.Population)\n\nvar apikey=\"Enter API key here\";\nmsg.headers={\"content-type\":\"application/x-www-form-urlencoded\"}\nmsg.payload=msg.payload={\"grant\_type\":\"urn:ibm:params:oauth:grant-type:apikey\",\"apikey\":apikey}\nreturn msg;\n",

        "outputs": 1,

        "noerr": 0,

        "x": 310,

        "y": 1380,

        "wires": [

            [

                "c97cde7b.eda04"

            ]

        ]

    },

    {

        "id": "c97cde7b.eda04",

        "type": "http request",

        "z": "816e7f0a.56a38",

        "name": "",

        "method": "POST",

        "ret": "obj",

        "paytoqs": false,

        "url": "https://iam.cloud.ibm.com/identity/token",

        "tls": "",

        "persist": false,

        "proxy": "",

        "authType": "",

        "x": 480,

        "y": 1380,

        "wires": [

            [

                "af8603e6.d8d1c"

            ]

        ]

    },

    {

        "id": "af8603e6.d8d1c",

        "type": "function",

        "z": "816e7f0a.56a38",

        "name": "Pre Prediction",

        "func": "var Country=global.get(\"Country\",msg.payload.Country)\nvar Year=global.get(\"Year\",msg.payload.Year)\nvar Status=global.get(\"Status\",msg.payload.Status)\nvar BMI=global.get(\"BMI\",msg.payload.BMI)\nvar GDP=global.get(\"GDP\",msg.payload.GDP)\nvar Population=global.get(\"Population\",msg.payload.Population)\n\narray=[Country,Year,Status,BMI,GDP,Population]\n\nvar token=msg.payload.access\_token\nvar instance\_id=\"afb5a4cd-139e-4e89-8714-a1d7921ada59\"\nmsg.headers={'Content-Type': 'application/json',\"Authorization\":\"Bearer \"+token,\"ML-Instance-ID\":instance\_id}\nmsg.payload= {\"input\_data\": [{\"fields\": [\"Country\", \"Year\", \"Status\", \" BMI \", \"GDP\", \"Population\"], \"values\": [array]}]}\nreturn msg;",

        "outputs": 1,

        "noerr": 0,

        "x": 670,

        "y": 1380,

        "wires": [

            [

                "62fe28b2.2ade88"

            ]

        ],

        "info": "array=[\"Country\",\"Year\",\"Status\",\"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\"]\narray=[Country,Year,Status,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]\n"

    },

    {

        "id": "62fe28b2.2ade88",

        "type": "http request",

        "z": "816e7f0a.56a38",

        "name": "",

        "method": "POST",

        "ret": "obj",

        "paytoqs": false,

        "url": "https://eu-gb.ml.cloud.ibm.com/v4/deployments/367b3aee-94d7-447e-aed1-9b8db390a501/predictions",

        "tls": "",

        "persist": false,

        "proxy": "",

        "authType": "",

        "x": 860,

        "y": 1380,

        "wires": [

            [

                "10d031bc.40285e",

                "a6175113.f284a"

            ]

        ]

    },

    {

        "id": "10d031bc.40285e",

        "type": "function",

        "z": "816e7f0a.56a38",

        "name": "",

        "func": "global.set('Expec\_b',msg.payload.predictions[0].values[0])\nmsg.payload={\n    'tab':'Life Expectancy'\n}\nreturn msg;",

        "outputs": 1,

        "noerr": 0,

        "x": 1020,

        "y": 1380,

        "wires": [

            [

                "a269e1c5.3c1a7",

                "5bd7f35f.230f7c"

            ]

        ]

    },

    {

        "id": "a269e1c5.3c1a7",

        "type": "ui\_ui\_control",

        "z": "816e7f0a.56a38",

        "name": "",

        "events": "all",

        "x": 1150,

        "y": 1380,

        "wires": [

            [

                "b93112b2.b5c8f"

            ]

        ]

    },

    {

        "id": "972c8e9.e77417",

        "type": "comment",

        "z": "816e7f0a.56a38",

        "name": "life expectancy basic",

        "info": "",

        "x": 170,

        "y": 1320,

        "wires": []

    },

    {

        "id": "6e333f8f.992a8",

        "type": "ui\_button",

        "z": "816e7f0a.56a38",

        "name": "",

        "group": "a9ad0f97.44215",

        "order": 2,

        "width": 4,

        "height": 1,

        "passthru": false,

        "label": "Complex Input Format",

        "tooltip": "",

        "color": "",

        "bgcolor": "",

        "icon": "",

        "payload": "true",

        "payloadType": "bool",

        "topic": "",

        "x": 270,

        "y": 1520,

        "wires": [

            [

                "17588b66.5d6105"

            ]

        ]

    },

    {

        "id": "7bf3eb9e.4d2614",

        "type": "ui\_button",

        "z": "816e7f0a.56a38",

        "name": "",

        "group": "a9ad0f97.44215",

        "order": 5,

        "width": 4,

        "height": 1,

        "passthru": false,

        "label": "Basic Input Format",

        "tooltip": "",

        "color": "",

        "bgcolor": "",

        "icon": "",

        "payload": "true",

        "payloadType": "bool",

        "topic": "",

        "x": 260,

        "y": 1580,

        "wires": [

            [

                "c6d65c8f.0aa0d"

            ]

        ]

    },

    {

        "id": "47dee13c.325ac",

        "type": "ui\_ui\_control",

        "z": "816e7f0a.56a38",

        "name": "",

        "events": "all",

        "x": 620,

        "y": 1540,

        "wires": [

            []

        ]

    },

    {

        "id": "17588b66.5d6105",

        "type": "function",

        "z": "816e7f0a.56a38",

        "name": "",

        "func": "msg.payload={\n    'tab':'Calculate Life Expectancy'\n}\nreturn msg;",

        "outputs": 1,

        "noerr": 0,

        "x": 460,

        "y": 1520,

        "wires": [

            [

                "47dee13c.325ac"

            ]

        ]

    },

    {

        "id": "c6d65c8f.0aa0d",

        "type": "function",

        "z": "816e7f0a.56a38",

        "name": "",

        "func": "msg.payload={\n    'tab':'Life Expectancy from Basic Data'\n}\nreturn msg;",

        "outputs": 1,

        "noerr": 0,

        "x": 450,

        "y": 1580,

        "wires": [

            [

                "47dee13c.325ac"

            ]

        ]

    },

    {

        "id": "d1bb7658.098528",

        "type": "comment",

        "z": "816e7f0a.56a38",

        "name": "select input complexity",

        "info": "",

        "x": 140,

        "y": 1480,

        "wires": []

    },

    {

        "id": "7a82f448.cd926c",

        "type": "comment",

        "z": "816e7f0a.56a38",

        "name": "Welcome page",

        "info": "",

        "x": 120,

        "y": 400,

        "wires": []

    },

    {

        "id": "b93112b2.b5c8f",

        "type": "link out",

        "z": "816e7f0a.56a38",

        "name": "",

        "links": [

            "3eeed506.9aa97a"

        ],

        "x": 1275,

        "y": 1380,

        "wires": []

    },

    {

        "id": "3eeed506.9aa97a",

        "type": "link in",

        "z": "816e7f0a.56a38",

        "name": "",

        "links": [

            "b93112b2.b5c8f",

            "6989581e.8799a8"

        ],

        "x": 75,

        "y": 1160,

        "wires": [

            [

                "52636452.2645fc"

            ]

        ]

    },

    {

        "id": "a6175113.f284a",

        "type": "debug",

        "z": "816e7f0a.56a38",

        "name": "",

        "active": true,

        "tosidebar": true,

        "console": false,

        "tostatus": false,

        "complete": "false",

        "x": 1010,

        "y": 1440,

        "wires": []

    },

    {

        "id": "6989581e.8799a8",

        "type": "link out",

        "z": "816e7f0a.56a38",

        "name": "",

        "links": [

            "3eeed506.9aa97a"

        ],

        "x": 1255,

        "y": 980,

        "wires": []

    },

    {

        "id": "149f63d2.98e83c",

        "type": "ui\_group",

        "z": "",

        "name": "Welcome",

        "tab": "388f326e.99d5de",

        "order": 1,

        "disp": true,

        "width": "6",

        "collapse": false

    },

    {

        "id": "7787cf17.7454a",

        "type": "ui\_group",

        "z": "",

        "name": "Sign In",

        "tab": "12e436a7.bbdfb9",

        "order": 1,

        "disp": true,

        "width": "6",

        "collapse": false

    },

    {

        "id": "6d92483e.53d708",

        "type": "ui\_group",

        "z": "",

        "name": "Life Expectancy",

        "tab": "c27ec4c9.638dd8",

        "order": 2,

        "disp": true,

        "width": "9",

        "collapse": false

    },

    {

        "id": "acc05384.74e84",

        "type": "ui\_group",

        "z": "",

        "name": "log in",

        "tab": "565531b9.89fe7",

        "order": 1,

        "disp": true,

        "width": "6",

        "collapse": false

    },

    {

        "id": "a6ab71a7.9d05b",

        "type": "ui\_group",

        "z": "",

        "name": "Calculate Life Expectancy",

        "tab": "5298d197.4b7ab",

        "order": 2,

        "disp": true,

        "width": "9",

        "collapse": false

    },

    {

        "id": "e6bbb703.a47b68",

        "type": "ui\_group",

        "z": "",

        "name": "Life Expectancy from Basic Data",

        "tab": "68377363.398e5c",

        "order": 1,

        "disp": true,

        "width": "6",

        "collapse": false

    },

    {

        "id": "a9ad0f97.44215",

        "type": "ui\_group",

        "z": "",

        "name": "Select Input Complexity",

        "tab": "6b61dc7a.e166b4",

        "order": 3,

        "disp": true,

        "width": 6,

        "collapse": false

    },

    {

        "id": "388f326e.99d5de",

        "type": "ui\_tab",

        "z": "",

        "name": "Welcome",

        "icon": "dashboard",

        "order": 8,

        "disabled": false,

        "hidden": false

    },

    {

        "id": "12e436a7.bbdfb9",

        "type": "ui\_tab",

        "z": "",

        "name": "Sign In",

        "icon": "dashboard",

        "order": 4,

        "disabled": false,

        "hidden": false

    },

    {

        "id": "c27ec4c9.638dd8",

        "type": "ui\_tab",

        "z": "",

        "name": "Life Expectancy",

        "icon": "dashboard",

        "disabled": false,

        "hidden": true

    },

    {

        "id": "565531b9.89fe7",

        "type": "ui\_tab",

        "z": "",

        "name": "Log in",

        "icon": "dashboard",

        "disabled": false,

        "hidden": false

    },

    {

        "id": "5298d197.4b7ab",

        "type": "ui\_tab",

        "z": "",

        "name": "Calculate Life Expectancy",

        "icon": "dashboard",

        "disabled": false,

        "hidden": true

    },

    {

        "id": "68377363.398e5c",

        "type": "ui\_tab",

        "z": "",

        "name": "Life Expectancy from Basic Data",

        "icon": "dashboard",

        "disabled": false,

        "hidden": true

    },

    {

        "id": "6b61dc7a.e166b4",

        "type": "ui\_tab",

        "z": "",

        "name": "Select Input Complexity",

        "icon": "dashboard",

        "disabled": false,

        "hidden": true

    }

]

**ML model 1**

{

    "cells": [

        {

            "cell\_type": "markdown",

            "metadata": {},

            "source": "### IBM AutoAI-SDK Auto-Generated Notebook v1.11.12\n\n\*\*Note:\*\* Notebook code generated using AutoAI will execute successfully. If code is modified or reordered,   \nthere is no guarantee it will successfully execute. This pipeline is optimized for the original dataset.  \nThe pipeline may fail or produce sub-optimium results if used with different data. For different data,  \nplease consider returning to AutoAI Experiments to generate a new pipeline. Please read our documentation   \nfor more information:   \n<a href=\"https://dataplatform.cloud.ibm.com/docs/content/wsj/analyze-data/autoai-notebook.html\">Cloud Platform</a>  \n\n\nBefore modifying the pipeline or trying to re-fit the pipeline, consider:   \nThe notebook converts dataframes to numpy arrays before fitting the pipeline   \n(a current restriction of the preprocessor pipeline). The known\_values\_list is passed by reference   \nand populated with categorical values during fit of the preprocessing pipeline. Delete its members before re-fitting."

        },

        {

            "cell\_type": "markdown",

            "metadata": {},

            "source": "<a id=\"content\"></a>\n## Notebook content\n\nThis notebook contains steps and code to demonstrate AutoAI pipeline. This notebook introduces commands for getting data,  \npipeline model, model inspection and testing.\n\nSome familiarity with Python is helpful. This notebook uses Python 3."

        },

        {

            "cell\_type": "markdown",

            "metadata": {

                "pycharm": {

                    "name": "#%% md\n"

                }

            },

            "source": "## Notebook goals\n\n-  inspection of trained pipeline via graphical vizualization and source code preview.\n-  pipeline evaluation.\n-  pipeline deployment and webservice scoring\n\n## Contents\n\nThis notebook contains the following parts:\n\n1.\t[Setup](#setup)     \n    a.  [Connection to WML](#wml\_connection)        \n    b.  [Defined variables from experiment](#variables\_definition)      \n2.\t[Pipeline inspection](#inspection)      \n    a.  [Get historical optimizer instance and training details](#get\_hist\_and\_train)      \n    b.  [Get pipeline](#get\_pipeline)      \n    c.  [Preview pipeline model to python code representation](#preview\_model\_to\_python\_code)      \n    d.  [Visualize pipeline](#visualize\_pipeline)      \n    e.  [Reading training and holdout](#train\_holdout\_read)      \n    f.  [Test pipeline model locally](#test\_model)      \n3.\t[Pipeline refinery](#refinery)      \n    a.  [Pipeline definition source code](#pipeline\_definition)      \n    b.  [Lale library](#lale\_library)      \n4.\t[Deployment and scoring](#scoring)      \n    a.  [Deployment creation](#deployment)      \n    b.  [Scoring of webservice](#online\_scoring)      \n    c.  [Deleting deployment](#delete\_deployment)      \n5.  [Authors](#authors)      "

        },

        {

            "cell\_type": "markdown",

            "metadata": {},

            "source": "<a id=\"setup\"></a>\n# Setup\n\nBefore you use the sample code in this notebook, you must perform the following setup tasks:\n - `watson-machine-learning-client` uninstallation of the old client\n - `watson-machine-learning-client-V4` installation\n - `autoai-libs` installation/upgrade\n - `lightgbm` installation/downgrade \n - `xgboost` installation/downgrade"

        },

        {

            "cell\_type": "code",

            "execution\_count": null,

            "metadata": {

                "pycharm": {

                    "name": "#%%\n"

                }

            },

            "outputs": [],

            "source": "!pip uninstall watson-machine-learning-client -y"

        },

        {

            "cell\_type": "code",

            "execution\_count": null,

            "metadata": {

                "pycharm": {

                    "is\_executing": false,

                    "name": "#%%\n"

                }

            },

            "outputs": [],

            "source": "!pip install -U watson-machine-learning-client-V4"

        },

        {

            "cell\_type": "code",

            "execution\_count": null,

            "metadata": {

                "pycharm": {

                    "name": "#%%\n"

                }

            },

            "outputs": [],

            "source": "!pip install -U autoai-libs"

        },

        {

            "cell\_type": "code",

            "execution\_count": null,

            "metadata": {

                "pycharm": {

                    "name": "#%%\n"

                }

            },

            "outputs": [],

            "source": "!pip install -U lightgbm==2.2.3"

        },

        {

            "cell\_type": "code",

            "execution\_count": null,

            "metadata": {

                "pycharm": {

                    "name": "#%%\n"

                }

            },

            "outputs": [],

            "source": "!pip install -U xgboost==0.90"

        },

        {

            "cell\_type": "markdown",

            "metadata": {},

            "source": "<a id=\"wml\_connection\"></a>\n### Connect to WML\n\*\*Action:\*\* Next you will need credentials for Watson Machine Learning:\n - go to [Cloud catalog resources list](https://cloud.ibm.com/resources)\n - click on Services and chose Machine Learning service. Once you are there\n - click the \*\*Service Credentials\*\* link on the left side of the screen\n - click to expand specific credentials name.\n - copy and paste your WML credentials into the cell below\n\n\*Take in mind that WML Service instance should be the same as used to generate this notebook.\*"

        },

        {

            "cell\_type": "code",

            "execution\_count": null,

            "metadata": {

                "pycharm": {

                    "is\_executing": false

                },

                "scrolled": true

            },

            "outputs": [],

            "source": "wml\_credentials = {\n  \"apikey\": \"\",\n  \"iam\_apikey\_description\": \"\",\n  \"iam\_apikey\_name\": \"\",\n  \"iam\_role\_crn\": \"r\",\n  \"iam\_serviceid\_crn\": \"\",\n  \"instance\_id\": \"\",\n  \"url\": \"\" \n}"

        },

        {

            "cell\_type": "markdown",

            "metadata": {},

            "source": "<a id=\"variables\_definition\"></a>\n### Defined variables from experiment\n\nThis cell contains input parameters provided to run the AutoAI experiment in Watson Studio."

        },

        {

            "cell\_type": "code",

            "execution\_count": null,

            "metadata": {

                "pycharm": {

                    "name": "#%%\n"

                }

            },

            "outputs": [],

            "source": "run\_id = '1029cd99-3de2-4043-ad2c-71e284d5fefb'\npipeline\_name = 'Pipeline\_3'\nscorer = 'neg\_root\_mean\_squared\_error'\ntarget\_label\_name = 'Life expectancy '"

        },

        {

            "cell\_type": "markdown",

            "metadata": {

                "pycharm": {

                    "name": "#%% md\n"

                }

            },

            "source": "<a id=\"inspection\"></a>\n## Pipeline inspection\nIn this section you will get the trained pipeline model from the AutoAI experiment and inspect it.  \nYou will see pipeline as a pythone code, graphically visualized and at the end, you will perform a local test.\n"

        },

        {

            "cell\_type": "markdown",

            "metadata": {

                "pycharm": {

                    "name": "#%% md\n"

                }

            },

            "source": "<a id=\"get\_hist\_and\_train\"></a>\n### Get historical optimizer instance and training details\n\nThe next cell contains code for retrieving fitted optimizer."

        },

        {

            "cell\_type": "code",

            "execution\_count": null,

            "metadata": {

                "pycharm": {

                    "is\_executing": false,

                    "name": "#%%\n"

                }

            },

            "outputs": [],

            "source": "from watson\_machine\_learning\_client.experiment import AutoAI\n\nexperiment = AutoAI(wml\_credentials)\noptimizer = experiment.runs.get\_optimizer(run\_id)"

        },

        {

            "cell\_type": "markdown",

            "metadata": {

                "pycharm": {

                    "name": "#%% md\n"

                }

            },

            "source": "<a id=\"get\_pipeline\"></a>\n### Get pipeline\n\nThe following cell loads selected AutoAI pipeline model. If you want to get pure scikit-learn pipeline specify `as\_type='sklearn'` parameter. By default enriched scikit-learn pipeline is returned `as\_type='lale'`."

        },

        {

            "cell\_type": "code",

            "execution\_count": null,

            "metadata": {

                "pycharm": {

                    "is\_executing": false,

                    "name": "#%%\n"

                }

            },

            "outputs": [],

            "source": "pipeline\_model = optimizer.get\_pipeline(pipeline\_name=pipeline\_name)"

        },

        {

            "cell\_type": "markdown",

            "metadata": {

                "pycharm": {

                    "name": "#%% md\n"

                }

            },

            "source": "<a id=\"preview\_model\_to\_python\_code\"></a>\n### Preview pipeline model as python code\nIn the next cell, downloaded pipeline model could be previewed as a python code.  \nYou will be able to see what exact steps are involved in model creation."

        },

        {

            "cell\_type": "code",

            "execution\_count": null,

            "metadata": {

                "pycharm": {

                    "name": "#%%\n"

                }

            },

            "outputs": [],

            "source": "pipeline\_model.pretty\_print(combinators=False, ipython\_display=True)"

        },

        {

            "cell\_type": "markdown",

            "metadata": {

                "pycharm": {

                    "name": "#%% md\n"

                }

            },

            "source": "<a id=\"visualize\_pipeline\"></a>\n### Visualize pipeline\n\nPreview pipeline model stages as graph. Each node's name links to detailed description of the stage.\n"

        },

        {

            "cell\_type": "code",

            "execution\_count": null,

            "metadata": {

                "pycharm": {

                    "is\_executing": false,

                    "name": "#%%\n"

                }

            },

            "outputs": [],

            "source": "pipeline\_model.visualize()"

        },

        {

            "cell\_type": "markdown",

            "metadata": {

                "pycharm": {

                    "name": "#%% md\n"

                }

            },

            "source": "<a id=\"train\_holdout\_read\"></a>\n### Reading training and holdout data\n\nRetrieve training dataset from AutoAI experiment as pandas DataFrame."

        },

        {

            "cell\_type": "code",

            "execution\_count": null,

            "metadata": {

                "pycharm": {

                    "name": "#%%\n"

                }

            },

            "outputs": [],

            "source": "training\_df, holdout\_df = optimizer.get\_data\_connections()[0].read(with\_holdout\_split=True)\n\ntrain\_X = training\_df.drop([target\_label\_name], axis=1).values\ntrain\_y = training\_df[target\_label\_name].values\n\ntest\_X = holdout\_df.drop([target\_label\_name], axis=1).values\ny\_true = holdout\_df[target\_label\_name].values"

        },

        {

            "cell\_type": "markdown",

            "metadata": {

                "pycharm": {

                    "name": "#%% md\n"

                }

            },

            "source": "<a id=\"test\_model\"></a>\n### Test pipeline model locally\n\*\*Note\*\*: you can chose the metric to evaluate the model by your own, this example contains only a basic scenario."

        },

        {

            "cell\_type": "code",

            "execution\_count": null,

            "metadata": {

                "pycharm": {

                    "name": "#%%\n"

                }

            },

            "outputs": [],

            "source": "from sklearn.metrics import r2\_score\n\npredictions = pipeline\_model.predict(test\_X)\nscore = r2\_score(y\_true=y\_true, y\_pred=predictions)\nprint('r2\_score: ', score)"

        },

        {

            "cell\_type": "markdown",

            "metadata": {

                "pycharm": {

                    "name": "#%% md\n"

                }

            },

            "source": "<a id=\"refinery\"></a>\n## Pipeline refinery and testing (optional)\n\nIn this section you will learn how to refine and retrain the best pipeline returned by AutoAI.\nIt can be performed by:\n - modifying pipeline definition source code\n - using [lale](https://lale.readthedocs.io/en/latest/) library for semi-automated data science\n\n\*\*Note\*\*: In order to run this section change following cells to 'code' cell."

        },

        {

            "cell\_type": "markdown",

            "metadata": {},

            "source": "<a id=\"pipeline\_definition\"></a>\n### Pipeline definition source code\nFollowing cell lets you experiment with pipeline definition in python, e.g. change steps parameters.\n\nIt will inject pipeline definition to the next cell."

        },

        {

            "cell\_type": "raw",

            "metadata": {

                "pycharm": {

                    "name": "#%%\n"

                }

            },

            "source": "pipeline\_model.pretty\_print(combinators=False, ipython\_display='input')"

        },

        {

            "cell\_type": "markdown",

            "metadata": {

                "pycharm": {

                    "name": "#%% md\n"

                }

            },

            "source": "<a id=\"lale\_library\"></a>\n### Lale library\n\n\*\*Note\*\*: This is only an exemplary usage of lale package. You can import more different estimators to refine downloaded pipeline model."

        },

        {

            "cell\_type": "markdown",

            "metadata": {},

            "source": "#### Import estimators"

        },

        {

            "cell\_type": "raw",

            "metadata": {

                "pycharm": {

                    "is\_executing": false,

                    "name": "#%%\n"

                }

            },

            "source": "from sklearn.linear\_model import LinearRegression as E1\nfrom sklearn.tree import DecisionTreeRegressor as E2\nfrom sklearn.neighbors import KNeighborsRegressor as E3\nfrom lale.lib.lale import Hyperopt\nfrom lale.operators import TrainedPipeline\nfrom lale import wrap\_imported\_operators\nfrom lale.helpers import import\_from\_sklearn\_pipeline\nwrap\_imported\_operators()"

        },

        {

            "cell\_type": "markdown",

            "metadata": {

                "pycharm": {

                    "name": "#%% md\n"

                }

            },

            "source": "<a id=\"decomposition\_definition\"></a>\n#### Pipeline decomposition and new definition\nIn this step the last stage from pipeline is removed."

        },

        {

            "cell\_type": "raw",

            "metadata": {

                "pycharm": {

                    "name": "#%%\n"

                }

            },

            "source": "prefix = pipeline\_model.remove\_last().freeze\_trainable()\nprefix.visualize()"

        },

        {

            "cell\_type": "raw",

            "metadata": {

                "pycharm": {

                    "is\_executing": false,

                    "name": "#%%\n"

                }

            },

            "source": "new\_pipeline = prefix >> (E1 | E2 | E3)\nnew\_pipeline.visualize()"

        },

        {

            "cell\_type": "markdown",

            "metadata": {

                "pycharm": {

                    "name": "#%% md\n"

                }

            },

            "source": "<a id=\"new\_optimizer\"></a>\n#### New optimizer `hyperopt` configuration and training\n\nThis section can introduce other results than the original one and it should be used\nby more advanced users.\n\nNew pipeline is re-trained by passing train data to it and calling `fit` method."

        },

        {

            "cell\_type": "raw",

            "metadata": {

                "pycharm": {

                    "name": "#%%\n"

                }

            },

            "source": "hyperopt = Hyperopt(estimator=new\_pipeline, cv=3, max\_evals=20, scoring='r2')\nfitted\_hyperopt = hyperopt.fit(train\_X, train\_y)"

        },

        {

            "cell\_type": "raw",

            "metadata": {

                "pycharm": {

                    "name": "#%%\n"

                }

            },

            "source": "hyperopt\_pipeline = fitted\_hyperopt.get\_pipeline()\nnew\_pipeline = hyperopt\_pipeline.export\_to\_sklearn\_pipeline()"

        },

        {

            "cell\_type": "raw",

            "metadata": {

                "pycharm": {

                    "name": "#%%\n"

                }

            },

            "source": "predictions = new\_pipeline.predict(train\_X)"

        },

        {

            "cell\_type": "raw",

            "metadata": {

                "pycharm": {

                    "name": "#%%\n"

                }

            },

            "source": "predictions = new\_pipeline.predict(test\_X)\nrefined\_score = r2\_score(y\_true=y\_true, y\_pred=predictions)\nprint('r2\_score: ', score)\nprint('refined\_r2\_score: ', refined\_score)"

        },

        {

            "cell\_type": "markdown",

            "metadata": {

                "pycharm": {

                    "name": "#%% md\n"

                }

            },

            "source": "<a id=\"scoring\"></a>\n## Deploy and Score\n\nIn this section you will learn how to deploy and score pipeline model as webservice using WML instance."

        },

        {

            "cell\_type": "markdown",

            "metadata": {},

            "source": "<a id=\"deployment\"></a>\n### Deployment creation\n\n\*\*Action\*\*: Please change:\n - the pipeline\_model to new\_pipeline in order to deploy refined pipeline\n - the deployment name"

        },

        {

            "cell\_type": "code",

            "execution\_count": null,

            "metadata": {

                "pycharm": {

                    "name": "#%%\n"

                }

            },

            "outputs": [],

            "source": "from watson\_machine\_learning\_client.deployment import WebService\n\nservice = WebService(wml\_credentials)\n\nservice.create(\n    experiment\_run\_id=run\_id,\n    model=pipeline\_model, \n    deployment\_name=f'{pipeline\_name}\_webservice'\n    )"

        },

        {

            "cell\_type": "markdown",

            "metadata": {},

            "source": "Deployment object could be printed to show basic information:"

        },

        {

            "cell\_type": "code",

            "execution\_count": null,

            "metadata": {

                "pycharm": {

                    "name": "#%%\n"

                }

            },

            "outputs": [],

            "source": "print(service)"

        },

        {

            "cell\_type": "markdown",

            "metadata": {},

            "source": "To be able to show all available information about deployment use `.get\_params()` method:"

        },

        {

            "cell\_type": "code",

            "execution\_count": null,

            "metadata": {

                "pycharm": {

                    "name": "#%%\n"

                }

            },

            "outputs": [],

            "source": "service.get\_params()"

        },

        {

            "cell\_type": "markdown",

            "metadata": {

                "pycharm": {

                    "name": "#%% md\n"

                }

            },

            "source": "<a id=\"online\_scoring\"></a>\n### Scoring of webservice\nYou can make scoring request by calling `score()` on deployed pipeline."

        },

        {

            "cell\_type": "code",

            "execution\_count": null,

            "metadata": {

                "pycharm": {

                    "name": "#%%\n"

                }

            },

            "outputs": [],

            "source": "predictions = service.score(payload=holdout\_df.drop([target\_label\_name], axis=1).iloc[:10])\npredictions"

        },

        {

            "cell\_type": "markdown",

            "metadata": {},

            "source": "If you want to work with the webservice in external Python application you can retrieve the service object by:\n - initialize service by `service = WebService(wml\_credentials)`\n - get deployment\_id by `service.list()` method\n - get webservice object by `service.get('deployment\_id')` method\n\nAfter that you can call `service.score()` method."

        },

        {

            "cell\_type": "markdown",

            "metadata": {

                "pycharm": {

                    "name": "#%% md\n"

                }

            },

            "source": "<a id=\"delete\_deployment\"></a>\n### Deleting deployment\n\nYou can delete an existing deployment by calling `service.delete()`."

        },

        {

            "cell\_type": "markdown",

            "metadata": {

                "pycharm": {

                    "name": "#%% md\n"

                }

            },

            "source": "<a id=\"authors\"></a>\n### Authors\n\nLicensed Materials - Copyright (C) 2020 IBM. This notebook and its source code are released under the terms of the ILAN License.\nUse, duplication disclosure restricted by GSA ADP Schedule Contract with IBM Corp.\n\n\*\*Note:\*\* The auto-generated notebooks are subject to the International License Agreement for Non-Warranted Programs  \n(or equivalent) and License Information document for Watson Studio Auto-generated Notebook (License Terms),  \nsuch agreements located in the link below. Specifically, the Source Components and Sample Materials clause  \nincluded in the License Information document for Watson Studio Auto-generated Notebook applies to the auto-generated notebooks.  \n\nBy downloading, copying, accessing, or otherwise using the materials, you agree to the <a href=\"http://www14.software.ibm.com/cgi-bin/weblap/lap.pl?li\_formnum=L-AMCU-BHU2B7&title=IBM%20Watson%20Studio%20Auto-generated%20Notebook%20V2.1\">License Terms</a>  \n\n\_\_\_"

        }

    ],

    "metadata": {

        "kernelspec": {

            "display\_name": "Python 3",

            "language": "python",

            "name": "python3"

        },

        "language\_info": {

            "codemirror\_mode": {

                "name": "ipython",

                "version": 3

            },

            "file\_extension": ".py",

            "mimetype": "text/x-python",

            "name": "python",

            "nbconvert\_exporter": "python",

            "pygments\_lexer": "ipython3",

            "version": "3.6.8"

        }

    },

    "nbformat": 4,

    "nbformat\_minor": 2

}

**ML model 2**

{

    "cells": [

        {

            "cell\_type": "markdown",

            "metadata": {},

            "source": "### IBM AutoAI-SDK Auto-Generated Notebook v1.12.2\n\n\*\*Note:\*\* Notebook code generated using AutoAI will execute successfully. If code is modified or reordered,   \nthere is no guarantee it will successfully execute. This pipeline is optimized for the original dataset.  \nThe pipeline may fail or produce sub-optimium results if used with different data. For different data,  \nplease consider returning to AutoAI Experiments to generate a new pipeline. Please read our documentation   \nfor more information:   \n<a href=\"https://dataplatform.cloud.ibm.com/docs/content/wsj/analyze-data/autoai-notebook.html\">Cloud Platform</a>  \n\n\nBefore modifying the pipeline or trying to re-fit the pipeline, consider:   \nThe notebook converts dataframes to numpy arrays before fitting the pipeline   \n(a current restriction of the preprocessor pipeline). The known\_values\_list is passed by reference   \nand populated with categorical values during fit of the preprocessing pipeline. Delete its members before re-fitting."

        },

        {

            "cell\_type": "markdown",

            "metadata": {},

            "source": "<a id=\"content\"></a>\n## Notebook content\n\nThis notebook contains steps and code to demonstrate AutoAI pipeline. This notebook introduces commands for getting data,  \npipeline model, model inspection and testing.\n\nSome familiarity with Python is helpful. This notebook uses Python 3."

        },

        {

            "cell\_type": "markdown",

            "metadata": {

                "pycharm": {

                    "name": "#%% md\n"

                }

            },

            "source": "## Notebook goals\n\n-  inspection of trained pipeline via graphical vizualization and source code preview.\n-  pipeline evaluation.\n-  pipeline deployment and webservice scoring\n\n## Contents\n\nThis notebook contains the following parts:\n\n1.\t[Setup](#setup)         \n    a.  [AutoAI experiment metadata](#variables\_definition)      \n2.\t[Pipeline inspection](#inspection)      \n    a.  [Get historical optimizer instance](#get\_hist\_and\_train)      \n    b.  [Get pipeline model](#get\_pipeline)      \n    c.  [Preview pipeline model as python code](#preview\_model\_to\_python\_code)      \n    d.  [Visualize pipeline model](#visualize\_pipeline)      \n    e.  [Read training and holdout data](#train\_holdout\_read)        \n    f.  [Test pipeline model locally](#test\_model)       \n3.\t[Pipeline refinery](#refinery)       \n    a.  [Pipeline definition source code](#pipeline\_definition)      \n    b.  [Lale library](#lale\_library)      \n4.\t[Deploy and score](#scoring)       \n    a.  [Insert WML credentials](#wml\_credentials)   \n    b.  [Create deployment](#deployment)      \n    c.  [Score webservice](#online\_scoring)        \n    d.  [Delete deployment](#delete\_deployment)       \n5.  [Authors](#authors)      "

        },

        {

            "cell\_type": "markdown",

            "metadata": {},

            "source": "<a id=\"setup\"></a>\n# Setup\n\nBefore you use the sample code in this notebook, you must perform the following setup tasks:\n - `watson-machine-learning-client` uninstallation of the old client\n - `watson-machine-learning-client-V4` installation\n - `autoai-libs` installation/upgrade\n - `lightgbm` or `xgboost` installation/downgrade if they are needed"

        },

        {

            "cell\_type": "code",

            "execution\_count": null,

            "metadata": {

                "pycharm": {

                    "name": "#%%\n"

                }

            },

            "outputs": [],

            "source": "!pip uninstall watson-machine-learning-client -y"

        },

        {

            "cell\_type": "code",

            "execution\_count": null,

            "metadata": {

                "pycharm": {

                    "is\_executing": false,

                    "name": "#%%\n"

                }

            },

            "outputs": [],

            "source": "!pip install -U watson-machine-learning-client-V4"

        },

        {

            "cell\_type": "code",

            "execution\_count": null,

            "metadata": {

                "pycharm": {

                    "name": "#%%\n"

                }

            },

            "outputs": [],

            "source": "!pip install -U autoai-libs"

        },

        {

            "cell\_type": "markdown",

            "metadata": {},

            "source": "<a id=\"variables\_definition\"></a>\n### AutoAI experiment metadata\n\nThis cell contains input parameters provided to run the AutoAI experiment in Watson Studio and COS credentials required to retrieve AutoAI pipeline."

        },

        {

            "cell\_type": "code",

            "execution\_count": null,

            "metadata": {

                "pycharm": {

                    "name": "#%%\n"

                }

            },

            "outputs": [],

            "source": "from watson\_machine\_learning\_client.helpers import DataConnection, S3Connection, S3Location\n\nexperiment\_metadata = dict(\n   prediction\_type='regression',\n   prediction\_column='Life expectancy ',\n   test\_size=0.1,\n   scoring='neg\_root\_mean\_squared\_error',\n   max\_number\_of\_estimators=2,\n   training\_data\_reference = [DataConnection(\n        connection=S3Connection(\n            api\_key='08nCUw6w-wVOwgjc6rIfemyMf4BlKFLCPzp4o-5dxAXm',\n            auth\_endpoint='https://iam.bluemix.net/oidc/token/',\n            endpoint\_url='https://s3.eu-geo.objectstorage.softlayer.net'\n        ),\n            location=S3Location(\n            bucket='mymltestproject1-donotdelete-pr-nwhm4oupt4yvim',\n            path='Life Expectancy Data Basic.csv'\n        ))\n    ],\n    training\_result\_reference = DataConnection(\n        connection=S3Connection(\n            api\_key='08nCUw6w-wVOwgjc6rIfemyMf4BlKFLCPzp4o-5dxAXm',\n            auth\_endpoint='https://iam.bluemix.net/oidc/token/',\n            endpoint\_url='https://s3.eu-geo.objectstorage.softlayer.net'\n        ),\n        location=S3Location(\n            bucket='mymltestproject1-donotdelete-pr-nwhm4oupt4yvim',\n            path='auto\_ml/f82fac0c-a310-4166-a31e-cb418ccd5f33/wml\_data/cb7f5d32-030d-4417-bc8f-59e6e2eef730/data/automl',\n            model\_location='auto\_ml/f82fac0c-a310-4166-a31e-cb418ccd5f33/wml\_data/cb7f5d32-030d-4417-bc8f-59e6e2eef730/data/automl/cognito\_output/Pipeline1/model.pickle',\n            training\_status='auto\_ml/f82fac0c-a310-4166-a31e-cb418ccd5f33/wml\_data/cb7f5d32-030d-4417-bc8f-59e6e2eef730/training-status.json'\n        )\n    ))\n\npipeline\_name='Pipeline\_3'"

        },

        {

            "cell\_type": "markdown",

            "metadata": {

                "pycharm": {

                    "name": "#%% md\n"

                }

            },

            "source": "<a id=\"inspection\"></a>\n## Pipeline inspection\nIn this section you will get the trained pipeline model from the AutoAI experiment and inspect it.  \nYou will see pipeline as a pythone code, graphically visualized and at the end, you will perform a local test.\n"

        },

        {

            "cell\_type": "markdown",

            "metadata": {

                "pycharm": {

                    "name": "#%% md\n"

                }

            },

            "source": "<a id=\"get\_hist\_and\_train\"></a>\n### Get historical optimizer instance\n\nThe next cell contains code for retrieving fitted optimizer."

        },

        {

            "cell\_type": "code",

            "execution\_count": null,

            "metadata": {

                "pycharm": {

                    "is\_executing": false,

                    "name": "#%%\n"

                }

            },

            "outputs": [],

            "source": "from watson\_machine\_learning\_client.experiment import AutoAI\n\noptimizer = AutoAI().runs.get\_optimizer(metadata=experiment\_metadata)"

        },

        {

            "cell\_type": "markdown",

            "metadata": {

                "pycharm": {

                    "name": "#%% md\n"

                }

            },

            "source": "<a id=\"get\_pipeline\"></a>\n### Get pipeline model\n\nThe following cell loads selected AutoAI pipeline model. If you want to get pure scikit-learn pipeline specify `as\_type='sklearn'` parameter. By default enriched scikit-learn pipeline is returned `as\_type='lale'`."

        },

        {

            "cell\_type": "code",

            "execution\_count": null,

            "metadata": {

                "pycharm": {

                    "is\_executing": false,

                    "name": "#%%\n"

                }

            },

            "outputs": [],

            "source": "pipeline\_model = optimizer.get\_pipeline(pipeline\_name=pipeline\_name)"

        },

        {

            "cell\_type": "markdown",

            "metadata": {

                "pycharm": {

                    "name": "#%% md\n"

                }

            },

            "source": "<a id=\"preview\_model\_to\_python\_code\"></a>\n### Preview pipeline model as python code\nIn the next cell, downloaded pipeline model could be previewed as a python code.  \nYou will be able to see what exact steps are involved in model creation."

        },

        {

            "cell\_type": "code",

            "execution\_count": null,

            "metadata": {

                "pycharm": {

                    "name": "#%%\n"

                }

            },

            "outputs": [],

            "source": "pipeline\_model.pretty\_print(combinators=False, ipython\_display=True)"

        },

        {

            "cell\_type": "markdown",

            "metadata": {

                "pycharm": {

                    "name": "#%% md\n"

                }

            },

            "source": "<a id=\"visualize\_pipeline\"></a>\n### Visualize pipeline model\n\nPreview pipeline model stages as graph. Each node's name links to detailed description of the stage.\n"

        },

        {

            "cell\_type": "code",

            "execution\_count": null,

            "metadata": {

                "pycharm": {

                    "is\_executing": false,

                    "name": "#%%\n"

                }

            },

            "outputs": [],

            "source": "pipeline\_model.visualize()"

        },

        {

            "cell\_type": "markdown",

            "metadata": {

                "pycharm": {

                    "name": "#%% md\n"

                }

            },

            "source": "<a id=\"train\_holdout\_read\"></a>\n### Read training and holdout data\n\nRetrieve training dataset from AutoAI experiment as pandas DataFrame."

        },

        {

            "cell\_type": "code",

            "execution\_count": null,

            "metadata": {

                "pycharm": {

                    "name": "#%%\n"

                }

            },

            "outputs": [],

            "source": "training\_df, holdout\_df = optimizer.get\_data\_connections()[0].read(with\_holdout\_split=True)\n\ntrain\_X = training\_df.drop([experiment\_metadata['prediction\_column']], axis=1).values\ntrain\_y = training\_df[experiment\_metadata['prediction\_column']].values\n\ntest\_X = holdout\_df.drop([experiment\_metadata['prediction\_column']], axis=1).values\ny\_true = holdout\_df[experiment\_metadata['prediction\_column']].values"

        },

        {

            "cell\_type": "markdown",

            "metadata": {

                "pycharm": {

                    "name": "#%% md\n"

                }

            },

            "source": "<a id=\"test\_model\"></a>\n### Test pipeline model locally\n\*\*Note\*\*: you can chose the metric to evaluate the model by your own, this example contains only a basic scenario."

        },

        {

            "cell\_type": "code",

            "execution\_count": null,

            "metadata": {

                "pycharm": {

                    "name": "#%%\n"

                }

            },

            "outputs": [],

            "source": "from sklearn.metrics import r2\_score\n\npredictions = pipeline\_model.predict(test\_X)\nscore = r2\_score(y\_true=y\_true, y\_pred=predictions)\nprint('r2\_score: ', score)"

        },

        {

            "cell\_type": "markdown",

            "metadata": {

                "pycharm": {

                    "name": "#%% md\n"

                }

            },

            "source": "<a id=\"refinery\"></a>\n## Pipeline refinery and testing (optional)\n\nIn this section you will learn how to refine and retrain the best pipeline returned by AutoAI.\nIt can be performed by:\n - modifying pipeline definition source code\n - using [lale](https://lale.readthedocs.io/en/latest/) library for semi-automated data science\n\n\*\*Note\*\*: In order to run this section change following cells to 'code' cell."

        },

        {

            "cell\_type": "markdown",

            "metadata": {},

            "source": "<a id=\"pipeline\_definition\"></a>\n### Pipeline definition source code\nFollowing cell lets you experiment with pipeline definition in python, e.g. change steps parameters.\n\nIt will inject pipeline definition to the next cell."

        },

        {

            "cell\_type": "raw",

            "metadata": {

                "pycharm": {

                    "name": "#%%\n"

                }

            },

            "source": "pipeline\_model.pretty\_print(combinators=False, ipython\_display='input')"

        },

        {

            "cell\_type": "markdown",

            "metadata": {

                "pycharm": {

                    "name": "#%% md\n"

                }

            },

            "source": "<a id=\"lale\_library\"></a>\n### Lale library\n\n\*\*Note\*\*: This is only an exemplary usage of lale package. You can import more different estimators to refine downloaded pipeline model."

        },

        {

            "cell\_type": "markdown",

            "metadata": {},

            "source": "#### Import estimators"

        },

        {

            "cell\_type": "raw",

            "metadata": {

                "pycharm": {

                    "is\_executing": false,

                    "name": "#%%\n"

                }

            },

            "source": "from sklearn.linear\_model import LinearRegression as E1\nfrom sklearn.tree import DecisionTreeRegressor as E2\nfrom sklearn.neighbors import KNeighborsRegressor as E3\nfrom lale.lib.lale import Hyperopt\nfrom lale.operators import TrainedPipeline\nfrom lale import wrap\_imported\_operators\nfrom lale.helpers import import\_from\_sklearn\_pipeline\nwrap\_imported\_operators()"

        },

        {

            "cell\_type": "markdown",

            "metadata": {

                "pycharm": {

                    "name": "#%% md\n"

                }

            },

            "source": "<a id=\"decomposition\_definition\"></a>\n#### Pipeline decomposition and new definition\nIn this step the last stage from pipeline is removed."

        },

        {

            "cell\_type": "raw",

            "metadata": {

                "pycharm": {

                    "name": "#%%\n"

                }

            },

            "source": "prefix = pipeline\_model.remove\_last().freeze\_trainable()\nprefix.visualize()"

        },

        {

            "cell\_type": "raw",

            "metadata": {

                "pycharm": {

                    "is\_executing": false,

                    "name": "#%%\n"

                }

            },

            "source": "new\_pipeline = prefix >> (E1 | E2 | E3)\nnew\_pipeline.visualize()"

        },

        {

            "cell\_type": "markdown",

            "metadata": {

                "pycharm": {

                    "name": "#%% md\n"

                }

            },

            "source": "<a id=\"new\_optimizer\"></a>\n#### New optimizer `hyperopt` configuration and training\n\nThis section can introduce other results than the original one and it should be used\nby more advanced users.\n\nNew pipeline is re-trained by passing train data to it and calling `fit` method."

        },

        {

            "cell\_type": "raw",

            "metadata": {

                "pycharm": {

                    "name": "#%%\n"

                }

            },

            "source": "hyperopt = Hyperopt(estimator=new\_pipeline, cv=3, max\_evals=20, scoring='r2')\nfitted\_hyperopt = hyperopt.fit(train\_X, train\_y)"

        },

        {

            "cell\_type": "raw",

            "metadata": {

                "pycharm": {

                    "name": "#%%\n"

                }

            },

            "source": "hyperopt\_pipeline = fitted\_hyperopt.get\_pipeline()\nnew\_pipeline = hyperopt\_pipeline.export\_to\_sklearn\_pipeline()"

        },

        {

            "cell\_type": "raw",

            "metadata": {

                "pycharm": {

                    "name": "#%%\n"

                }

            },

            "source": "predictions = new\_pipeline.predict(train\_X)"

        },

        {

            "cell\_type": "raw",

            "metadata": {

                "pycharm": {

                    "name": "#%%\n"

                }

            },

            "source": "predictions = new\_pipeline.predict(test\_X)\nrefined\_score = r2\_score(y\_true=y\_true, y\_pred=predictions)\nprint('r2\_score: ', score)\nprint('refined\_r2\_score: ', refined\_score)"

        },

        {

            "cell\_type": "markdown",

            "metadata": {

                "pycharm": {

                    "name": "#%% md\n"

                }

            },

            "source": "<a id=\"scoring\"></a>\n## Deploy and Score\n\nIn this section you will learn how to deploy and score pipeline model as webservice using WML instance."

        },

        {

            "cell\_type": "markdown",

            "metadata": {},

            "source": "<a id=\"wml\_credentials\"></a>\n### Connect to WML client in order to create deployment\n\*\*Action:\*\* Next you will need credentials for Watson Machine Learning and training run\_id:\n - go to [Cloud catalog resources list](https://cloud.ibm.com/resources)\n - click on Services and chose Machine Learning service. Once you are there\n - click the \*\*Service Credentials\*\* link on the left side of the screen\n - click to expand specific credentials name.\n - copy and paste your WML credentials into the cell below\n\n\*Take in mind that WML Service instance should be the same as used to generate this notebook.\*"

        },

        {

            "cell\_type": "code",

            "execution\_count": null,

            "metadata": {

                "pycharm": {

                    "name": "#%%\n"

                }

            },

            "outputs": [],

            "source": "wml\_credentials = {\n  \"apikey\": \"\",\n  \"iam\_apikey\_description\": \"\",\n  \"iam\_apikey\_name\": \"\",\n  \"iam\_role\_crn\": \"r\",\n  \"iam\_serviceid\_crn\": \"\",\n  \"instance\_id\": \"\",\n  \"url\": \"\"\n}"

        },

        {

            "cell\_type": "markdown",

            "metadata": {},

            "source": "<a id=\"deployment\"></a>\n### Create deployment\n\n\*\*Action\*\*: If you want to deploy refined pipeline please change the `pipeline\_model` to `new\_pipeline`.\n \nIf you prefer you can also change the `deployment\_name`."

        },

        {

            "cell\_type": "code",

            "execution\_count": null,

            "metadata": {

                "pycharm": {

                    "name": "#%%\n"

                }

            },

            "outputs": [],

            "source": "from watson\_machine\_learning\_client.deployment import WebService\n\nservice = WebService(wml\_credentials)\n\nservice.create(\n    model=pipeline\_model,\n    metadata=experiment\_metadata,\n    deployment\_name=f'{pipeline\_name}\_webservice'\n)"

        },

        {

            "cell\_type": "markdown",

            "metadata": {},

            "source": "Deployment object could be printed to show basic information:"

        },

        {

            "cell\_type": "code",

            "execution\_count": null,

            "metadata": {

                "pycharm": {

                    "name": "#%%\n"

                }

            },

            "outputs": [],

            "source": "print(service)"

        },

        {

            "cell\_type": "markdown",

            "metadata": {},

            "source": "To be able to show all available information about deployment use `.get\_params()` method:"

        },

        {

            "cell\_type": "code",

            "execution\_count": null,

            "metadata": {

                "pycharm": {

                    "name": "#%%\n"

                }

            },

            "outputs": [],

            "source": "service.get\_params()"

        },

        {

            "cell\_type": "markdown",

            "metadata": {

                "pycharm": {

                    "name": "#%% md\n"

                }

            },

            "source": "<a id=\"online\_scoring\"></a>\n### Score webservice\nYou can make scoring request by calling `score()` on deployed pipeline."

        },

        {

            "cell\_type": "code",

            "execution\_count": null,

            "metadata": {

                "pycharm": {

                    "name": "#%%\n"

                }

            },

            "outputs": [],

            "source": "predictions = service.score(payload=holdout\_df.drop([experiment\_metadata['prediction\_column']], axis=1).iloc[:10])\npredictions"

        },

        {

            "cell\_type": "markdown",

            "metadata": {},

            "source": "If you want to work with the webservice in external Python application you can retrieve the service object by:\n - initialize service by `service = WebService(wml\_credentials)`\n - get deployment\_id by `service.list()` method\n - get webservice object by `service.get('deployment\_id')` method\n\nAfter that you can call `service.score()` method."

        },

        {

            "cell\_type": "markdown",

            "metadata": {

                "pycharm": {

                    "name": "#%% md\n"

                }

            },

            "source": "<a id=\"delete\_deployment\"></a>\n### Delete deployment\n\nYou can delete an existing deployment by calling `service.delete()`."

        },

        {

            "cell\_type": "markdown",

            "metadata": {

                "pycharm": {

                    "name": "#%% md\n"

                }

            },

            "source": "<a id=\"authors\"></a>\n### Authors\n\nLicensed Materials - Copyright \u00a9 2020 IBM. This notebook and its source code are released under the terms of the ILAN License.\nUse, duplication disclosure restricted by GSA ADP Schedule Contract with IBM Corp.\n\n\*\*Note:\*\* The auto-generated notebooks are subject to the International License Agreement for Non-Warranted Programs  \n(or equivalent) and License Information document for Watson Studio Auto-generated Notebook (License Terms),  \nsuch agreements located in the link below. Specifically, the Source Components and Sample Materials clause  \nincluded in the License Information document for Watson Studio Auto-generated Notebook applies to the auto-generated notebooks.  \n\nBy downloading, copying, accessing, or otherwise using the materials, you agree to the <a href=\"http://www14.software.ibm.com/cgi-bin/weblap/lap.pl?li\_formnum=L-AMCU-BHU2B7&title=IBM%20Watson%20Studio%20Auto-generated%20Notebook%20V2.1\">License Terms</a>  \n\n\_\_\_"

        }

    ],

    "metadata": {

        "kernelspec": {

            "display\_name": "Python 3",

            "language": "python",

            "name": "python3"

        },

        "language\_info": {

            "codemirror\_mode": {

                "name": "ipython",

                "version": 3

            },

            "file\_extension": ".py",

            "mimetype": "text/x-python",

            "name": "python",

            "nbconvert\_exporter": "python",

            "pygments\_lexer": "ipython3",

            "version": "3.6.8"

        }

    },

    "nbformat": 4,

    "nbformat\_minor": 2

}