**summer internship project**

Kidney Disease Prediction Chronic Using Auto watson AI



**SmartInternz Project**

**Submitted By – Anshu Singh**

|  |  |  |  |
| --- | --- | --- | --- |
| 1 |  | INTRODUCTION | |
|  |  | 1.1 Overview | |
|  |  | 1.2 Purpose | |
| 2 |  | LITERATURE SURVEY | |
|  |  | 2.1 Existing problem | | |
|  |  | 2.2 Proposed solution | | |
| 3 |  | THEORITICAL ANALYSIS | |
|  |  | 3.1 Block diagram | |
|  |  | 3.2 Hardware / Software designing | | |
| 4 |  | EXPERIMENTAL INVESTIGATIONS | | |
| 5 |  | FLOWCHART | |
| 6 |  | RESULT |  |
| 7 |  | ADVANTAGES & DISADVANTAGES | | |
| 8 |  | APPLICATIONS | |
| 9 |  | CONCLUSION | |
| 10 |  | FUTURE SCOPE | |
| 11 |  | BIBILOGRAPHY | |
|  |  | APPENDIX | |

CONTENT

1. Introduction

1.1 Overview

Chronic Kidney Disease (CKD) is a major medical problem and can be cured if only treated it in the early stages. Usually, people are not aware that medical tests, we take for different purposes could contain valuable information concerning kidney diseases.consequetly, attributes of various medical tests are investigated to distinguish which attributes may contain helpful information about the disease.The information says that it helps us to measure theseverity of the problem, the predicted survival of the patient after the illness, the pattern of the disease and work for curing the disease. This project foc-ues on automatic classification of the patients may having chronic kidney Disease symtomps based on their medical records.

1.2 Purpose

We define the task to solve as follows : prediction based on medical reports which gives the classification of any patient as ckd or notckd at a certain moment in time, given the patient’s medical history up to that moment. In order to learn the task automatically from data, we trained amodel on medical records of patients with some valuable attribues for the target to be predicted. We optimized the model architecture and feature set, and tested the performance of several models. Various data mining classification approaches and machine learning algorithms are applied for prediction of chronic diseases. Here we are concerned about Chronic kidney disease (CKD), also known as chronic renal disease, is an abnormal function of kidney or a progressive failure of renal function over a period of months or years. Often, chronic kidney disease is diagnosed as a result of screening of people known to be at risk of kidney problems, such as those with high blood pressure or diabetes and those with a blood relative with CKD.

The following sections describe:

* Collecting dataset;
* the train-validation-test split;
* our methods for creating the input data for the model;
* our methods for determining the model architecture;

our methods for feature selection;

* The evaluation protocol

  2. Literature Survey

2.1 Existing problem

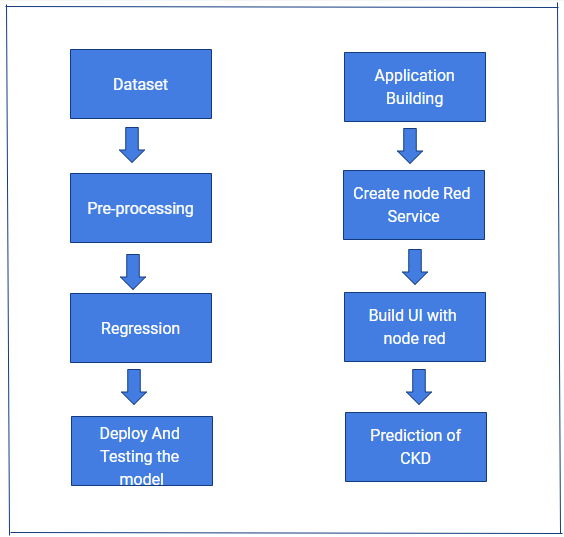
Chronic Kidney Disease (CKD) is a major medical problem and can be cured if only treated it in the early stages. Life expectancy plays an important role when decisions about the final phase of life need to be made. Good prognostication for example helps to determine the course of treatment and helps to anticipate the procurement of health care services and facilities, or more broadly: facilitates Advance Care Planning. Advance Care Planning (ACP) is the process during which patients make decisions about the health care they wish to receive in the future, in case the patient loses the capacity of making decisions or communicating about them .

2.2 Proposed Solution

In this proposed system we are able to identify the patients with disease. Once any person gets kidney disease, they may suffer from the disease which may decrease their working capability as well as living quality. Our aim is to predict patients with chronic kidney failure (ckd) disease and patients who do not (notckd) suffer from the disease. So for that we are building a Machine Learning model to predict the compressive strength of concrete using IBM Watson AutoAI Machine Learning Service. The model is deployed on IBM cloud to get scoring end point which can be used as API in mobile app or web app building. We are developing a web application which is built using node red service. We make use of the scoring end point to give user input values to the deployed model. The model prediction is then showcased on User Interface.

**3. THEORETICAL ANALYSIS**

# 3.1 Block Diagram



**3.2Hardware/Software designing**

3.2.1IBM Cloud Platform

IBM cloud computing is a set of [cloud computing](https://en.wikipedia.org/wiki/Cloud_computing) services for business offered by the information technology company IBM. IBM Cloud includes infrastructure as a service (IaaS), software as a service (SaaS) and platform as a service (PaaS) offered through public, private and hybrid cloud delivery models, in addition to the components that make up those clouds.

IBM offers three hardware platforms for cloud computing.[[1]](https://en.wikipedia.org/wiki/IBM_cloud_computing#cite_note-IBM_hardware-1) These platforms offer built-in support for [virtualization](https://en.wikipedia.org/wiki/Virtualization). For [virtualization](https://en.wikipedia.org/wiki/Virtualization) IBM offers IBM [Websphere](https://en.wikipedia.org/wiki/Websphere) application infrastructure that supports programming models and open standards for virtualization.

The management layer of the IBM cloud framework includes IBM [Tivoli](https://en.wikipedia.org/wiki/Tivoli_Software) middleware. Management tools provide capabilities to regulate images with automated [provisioning](https://en.wikipedia.org/wiki/Provisioning) and de-provisioning, monitor operations and meter usage while tracking costs and allocating billing. The last layer of the framework provides integrated workload tools. Workloads for cloud computing are services or instances of code that can be executed to meet specific business needs. IBM offers tools for cloud based collaboration, development and test, application development, analytics, business-to-business integration, and security.

**3.2.2 IBM Node-Red Service**

Browser-based flow editing

Node-RED provides a browser-based flow editor that makes it easy to wire together flows using the wide range of nodes in the palette. Flows can be then deployed to the runtime in a single-click.

JavaScript functions can be created within the editor using a rich text editor.A built-in library allows you to save useful functions, templates or flows for re-use.

**Built on Node.js**

The light-weight runtime is built on Node.js, taking full advantage of its event-driven, non-blocking model. This makes it ideal to run at the edge of the network on low-cost hardware such as the Raspberry Pi as well as in the cloud.

With over 225,000 modules in Node's package repository, it is easy to extend the range of palette nodes to add new capabilities.

**Social Development**

The flows created in Node-RED are stored using JSON which can be easily imported and exported for sharing with others.

An online flow library allows you to share your best flows with the world.

**3.2.3 IBM Watson Service**

Watson was created as a [question answering](https://en.wikipedia.org/wiki/Question_answering) (QA) computing system that IBM built to apply advanced [natural language processing](https://en.wikipedia.org/wiki/Natural_language_processing), [information retrieval](https://en.wikipedia.org/wiki/Information_retrieval), [knowledge representation](https://en.wikipedia.org/wiki/Knowledge_representation), [automated reasoning](https://en.wikipedia.org/wiki/Automated_reasoning), and [machine learning](https://en.wikipedia.org/wiki/Machine_learning) technologies to the field of [open domain question answering](https://en.wikipedia.org/wiki/Open_domain_question_answering).

The key difference between QA technology and document search is that document search takes a keyword query and returns a list of documents, ranked in order of relevance to the query (often based on popularity and page ranking), while QA technology takes a question expressed in natural language, seeks to understand it in much greater detail, and returns a precise answer to the question.

When created, IBM stated that, "more than 100 different techniques are used to analyze natural language, identify sources, find and generate hypotheses, find and score evidence, and merge and rank hypotheses.

In recent years, the Watson capabilities have been extended and the way in which Watson works has been changed to take advantage of new deployment models (Watson on IBM Cloud) and evolved machine learning capabilities and optimised hardware available to developers and researchers. It is no longer purely a [question answering](https://en.wikipedia.org/wiki/Question_answering) (QA) computing system designed from Q&A pairs but can now 'see', 'hear', 'read', 'talk', 'taste', 'interpret', 'learn' and 'recommend'.

* **Software**

Watson uses IBM's software and the Apache [UIMA](https://en.wikipedia.org/wiki/UIMA) (Unstructured Information Management Architecture) framework implementation. The system was written in various languages, including [Java](https://en.wikipedia.org/wiki/Java_(programming_language)), [C++](https://en.wikipedia.org/wiki/C%2B%2B), and [Prolog](https://en.wikipedia.org/wiki/Prolog), and runs on the [SUSE Linux Enterprise Server](https://en.wikipedia.org/wiki/SUSE_Linux_Enterprise_Server) 11.

* **Hardware**

The system is workload-optimized, integrating massively parallel POWER7 processors and built on IBM’s DeepQA technology, which it uses to generate hypotheses, gather massive evidence, and analyze data. Watson employs a cluster of ninety IBM Power 750 servers, each of which uses a 3.5GHz POWER7 eight-core processor, with four threads per core. In total, the system has 2,880 POWER7 processor threads and 16 terab ytes of RAM.

* **Data**

The sources of information for Watson include encyclopedias, dictionaries, thesauri, newswire articles and literally works. Watson also used databases, taxonomies and ontologies including DBPedia, WordNet and Yago. The IBM team provided Watson with millions of documents, including dictionaries, encyclopedias and other reference material that it could use to build its knowledge.

**3.2.4 IBM Watson Studio**

IBM Watson Studio helps data scientists and analysts prepare data and build models at scale across any cloud. With its open, flexible multicloud architecture, Watson Studio provides capabilities that empower businesses to simplify enterprise data science and AI:

* Automate AI lifecycle management with AutoAI.
* Visually prepare and build models with IBM SPSS Modeler.
* Build models using images with IBM Watson Visual Recognition and texts with IBM Watson Natural Language Classifier.
* Deploy and run models through one-click integration with IBM Watson Machine Learning.
* Manage and monitor models through integration with IBM Watson Open Scale.

Hardware:

* A laptop with at least 4GB RAM
* A laptop with at least 4GB RAM
* A 2GB GPU

**4. Experimental Investigations**

Increasing age is a risk factor for many diseases; therefore developing pharmacological interventions that slow down ageing and consequently postpone the onset of many age-related diseases is highly desirable. In this work we analyse data from the Drug Age database, which contains chemical compounds and their effect on the lifespan of model organisms. Predictive models were built using the machine learning method random forests to predict whether or not a chemical compound will increase Caenorhabditis elegans' lifespan, using as features Gene Ontology (GO) terms annotated for proteins targeted by the compounds and chemical descriptors calculated from each compound's chemical structure. The model with the best predictive accuracy used both biological and chemical features, achieving a prediction accuracy of 80%. The top 20 most important GO terms include those related to mitochondrial processes, to enzymatic and immunological processes, and terms related to metabolic and transport processes. We applied our best model to predict compounds which are more likely to increase C. elegans' lifespan in the DGIdb database, where the effect of the compounds on an organism's lifespan is unknown. The top hit compounds can be broadly divided into four groups: compounds affecting mitochondria, compounds for cancer treatment, anti-inflammatories, and compounds for gonadotropinreleasing hormone therapies.

**4.2Prediction using AutoAI/ Without Python**

In this project we do prediction by using AutoAI it is the app in IBM Cloud service which is used to define prediction by many algorithm within a step without writing a code in it.

Two types of algorithm applied in it :-

* Extra Trees Regression
* Decision Tree Regression

AutoAI have the ability to choose best algorithm for data by visualization.

It can chose, Linear Regression and Logistic Regression etc.

In this we used Node-Red application also, it is used for the make Flowgraph and calculate the prediction easily if your Flowgraph is correct.

* **AutoAI**

**Automated Artificial Intelligence (AutoAI)** is a variation of the [automated machine learning](https://en.wikipedia.org/wiki/Automated_machine_learning), or AutoML, technology, which extends the automation of model building towards automation of the full life cycle of a machine learning model. It applies intelligent automation to the task of building [predictive](https://en.wikipedia.org/wiki/Predictive_modelling) machine learning models by preparing data for training, identifying the best type of model for the given data, then choosing the features, or columns of data, that best support the problem the model is solving. Finally, automation tests a variety of tuning options to reach the best result as it generates, then ranks, model-candidate pipelines. The best performing pipelines can be put into production to process new data, and deliver predictions based on the model training. Automated artificial intelligence can also be applied to making sure the model does not have inherent bias and automating the tasks for continuous improvement of the model.

The Automated Machine Learning and Data Science Team (AMLDS),[[2]](https://en.wikipedia.org/wiki/AutoAI#cite_note-2) a small team within IBM Research, which was formed to “apply techniques from [Artificial Intelligence](https://en.wikipedia.org/wiki/Artificial_intelligence) (AI), [Machine Learning](https://en.wikipedia.org/wiki/Machine_learning) (ML), and [data management](https://en.wikipedia.org/wiki/Data_management) to accelerate and optimize the creation of machine learning and data science workflows,” is credited with advancing the development of AutoAI.

**Usecase of AutoAI:-**

A typical use case for AutoAI would be training a model to predict how customers might respond to a sales incentive. The model is first trained with actual data on how customers responded to the promotion. Presented with new data, the model can provide a prediction of how a new customer might respond, with a confidence score for the prediction. Prior to AutoML, data scientists had to build these predictive models by hand, testing various combinations of algorithms, then testing to see how predictions compared to actual results. Where AutoML automated some of the process of preparing the data for training, applying algorithms to process the data and then further optimizing the results, AutoAI provides greater intelligent automation that allows for testing significantly more combinations of factors to generate model candidate pipelines that more accurately reflect and address the problem being solved. Once built, a model

can be tested for bias and updated to improve performance.

The AutoAI have do their prediction in 5 steps:-

1. Provide data in .csv file.

2. Prepare data.

* Featured type detection.
* Missing values imputation.
* Feature encoding and scaling.

3. Select model type.

Selection of best algorithm for the data.

4. Generate and Rank model pipelines.

* Hyper-Parameter optimization(HPO).
* Optimized feature engineering.

5. Save and Deploy a model.

* **Node-Red**

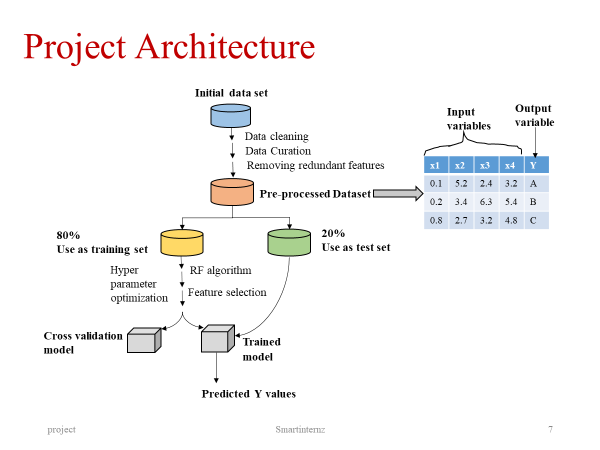
 Node-Red is used to make Flowgraph.

**Node-RED** is a [flow-based](https://en.wikipedia.org/wiki/Flow-based_programming) development tool for [visual programming](https://en.wikipedia.org/wiki/Visual_programming_language) developed originally by [IBM](https://en.wikipedia.org/wiki/IBM) for wiring together hardware devices, [APIs](https://en.wikipedia.org/wiki/Application_programming_interface) and [online services](https://en.wikipedia.org/wiki/Online_services) as part of the [Internet of Things](https://en.wikipedia.org/wiki/Internet_of_Things).

Node-RED provides a [web browser](https://en.wikipedia.org/wiki/Web_browser)-based flow editor, which can be used to create [JavaScript](https://en.wikipedia.org/wiki/JavaScript) functions. Elements of applications can be saved or shared for re-use. The runtime is built on [Node.js](https://en.wikipedia.org/wiki/Node.js). The flows created in Node-RED are stored using JSON. MQTT nodes can make properly configures TLS connections.

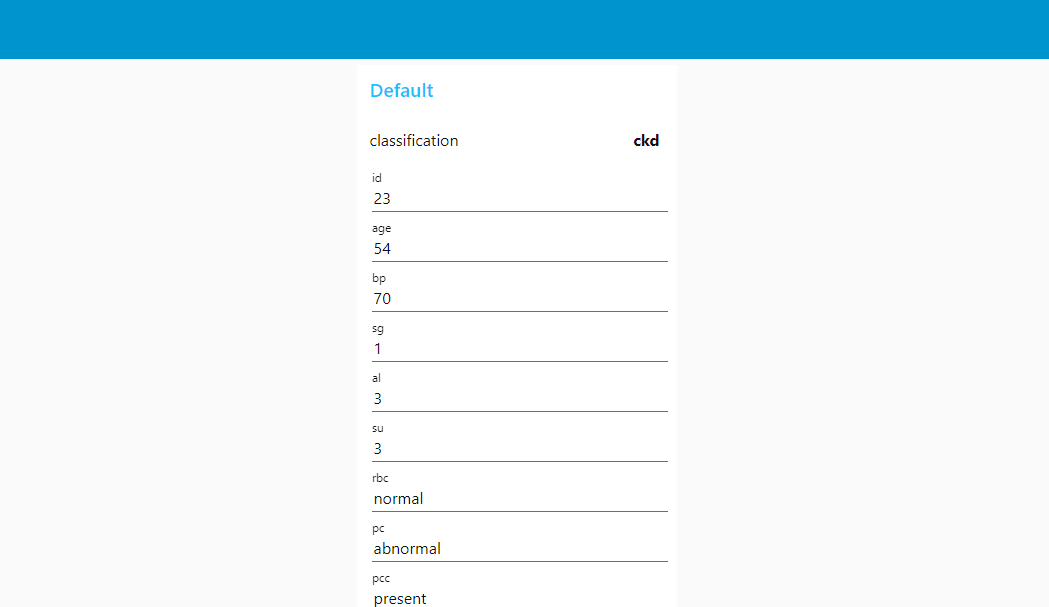
Node-Red is a Visual tool for wiring the internet of Things.

**5. Flowchart**



**6. Result**

The user should enter the credentials required for prediction . The prediction of Chronic kidney Disease patient will be displayed on the screen.



**7. Advantages & Disadvantages**

**Advantages of Machine learning**

1. Easily identifies trends and patterns

Machine Learning can review large volumes of data and discover specific trends and patterns that would not be apparent to humans. For instance, for an e-commerce website like Amazon, it serves to understand the browsing behaviours and purchase histories of its users to help cater to the right products, deals, and reminders relevant to them. It uses the results to reveal relevant advertisements to them.

**2. No human intervention needed (automation)**

With ML, you don’t need to babysit your project every step of the way. Since it means giving machines the ability to learn, it lets them make predictions and also improve the algorithms on their own. A common example of this is anti-virus softwires; they learn to filter new threats as they are recognized. ML is also good at recognizing spam.

**3. Continuous Improvement**

As ml models  gain experience, they keep improving in accuracy and efficiency. This lets them make better decisions. Say you need to make a weather forecast model. As the amount of data you have keeps growing, your algorithms learn to make more accurate predictions faster.

**4. Handling multi-dimensional and multi-variety data---**Machine Learning algorithms are good at handling data that are multi-dimensional and multi-variety, and they can do this in dynamic or uncertain environments.

**5. Wide Applications**

You could be an e-tailer or a healthcare provider and make ML work for you. Where it does apply, it holds the capability to help deliver a much more personal experience to customers while also targeting the right customers.

**Disadvantages of Machine Learning**

With all those advantages to its powerfulness and popularity, Machine Learning isn’t perfect. The following factors serve to limit it:

**1. Data Acquisition**

Machine Learning requires massive data sets to train on, and these should be inclusive/unbiased, and of good quality. There

can also be times where they must wait for new data to be generated.

**2. Time and Resources**

ML needs enough time to let the algorithms learn and develop enough to fulfill their purpose with a considerable amount of accuracy and relevancy. It also needs massive resources to function. This can mean additional requirements of computer power for you.

**3. Interpretation of Results**

Another major challenge is the ability to accurately interpret results generated by the algorithms. You must also carefully choose the algorithms for your purpose.

**4. High error-susceptibility**

**ML** is autonomous but highly susceptible to errors. Suppose you train an algorithm with data sets small enough to not be inclusive. You end up with biased predictions coming from a biased training set. This leads to irrelevant advertisements being displayed to customers. In the case of ML, such blunders can set off a chain of errors that can go undetected for long periods of time. And when they do get noticed, it takes quite some time to recognize the source of the issue, and even longer to correct it.

**8. Applications**

Existing health mobile app can integrate PLE feature as additional services (b) as both functions require the same physiological data to transfer to a monitoring center for PLE calculation. Applications Along with existing heath

applications such as fitness tracking, chronic disease monitoring and real-time patient monitoring, the PLE application can be useful for users to improve their lifestyle and exercise by planning goals on a short and long-term basis. For example, the current PLE outcome of 85 years will be adjusted when the user changes their attributes such as smoking cessation, reducing alcohol consumption, commencing regular exercise, or modifying dietary plans.

**9.Conclusion**

Prognostication of life expectancy is difficult for humans. Our research shows that machine learning and natural language processing techniques offer a feasible and promising approach to predicting life expectancy. The research has potential for real-life applications, such as supporting timely recognition of the right moment to start Advance Care Planning. After, the code we got the life ecpectancy prediction nearly accuracy with some error.

**10.Future Scope**

Few works have been done to provide an individually customized life expectancy prediction. We have reviewed existing works and techniques in the prediction of human LE, and reached a conclusion that it is feasible to predict a PLE for individuals using evolving technologies and devices such as big data, AI, machine learning techniques, and PHDs, wearables and mobile health monitoring devices. We also identified that the collection of data will be a huge challenge due to the privacy and government policy considerations, which will require collaboration of various bodies in the health industry. The interworking of a heterogeneous health network is also a challenge for data collection. Despite these challenges, we showed a possibility of a PLE prediction by proposing an approach of data collection and application by smartphone, with which users can enter their information to access the cloud server to obtain their own PLE. No attempt has been made to create this novel idea of using smartphone integrating cloud servers for real-time data entry. We investigated obstacles and barriers that can be resolved by future works described below. Previous works have described a five year LE prediction, however it is not oriented as a personalized prediction but rather utilizes a median model-predicted probability of 5-year survival of patients who are either sick or healthy. It is proposed that this can be extended to a lifetime prediction by using big data to generate a generic data, which can be used to create a PLE based

on training data as a future solution. Building a generic database will take a considerable amount of time for data collection and analysis, taken from birth to death for various demographic groups to be useful and accurate in representing each attribute classifications. Whilst current applications attempt to show PLE for smartphone users, they are complicated and difficult to collect technical data requested by the questionnaire, as users are unlikely to be able to provide these data themselves. This can be resolved by connecting the app to the central cloud server with the mHealth networks which provide other health related applications. A centralized cloud server plays a key role in collecting, processing, and creating meaningful value using big data, which forms the input of the solution as well as creates generic data against each user’s PLE requirements. Service providers shall envisage challenges and hurdles to obtain consent of personal health ‘of heterogeneous health networks across developed countries. This will lead to classification of data based on processing big data and each group’s traits, which can be used as personalized threshold ranges; When this has been completed in a cloud, it can be connected to a smart device app that can provide questionnaires developed by health specialists and collect answers to customize the user’s PLE; Optimization of the generic groups’ data is done by developing an algorithm using machine learning for continuously building and optimizing the user’s generic data. As the proposed solution requires processing and transmitting health information of users, information security is a key aspect to consider such as privacy as well as ethical requirements recommenced by regulation bodies, such as the Australian national health and medical research . The scope of security and ethical requirements need to be clearly defined and specified for future work as challenges are expected to build a centralized database with incorporation of health networks. For example, North America, Asia, and Europe may have their own unique requirements to satisfy in dealing with health data with different health research guidelines.

10. Bibilography

* https://smartinternz.com/Student/workspace/2932
* https://www.kaggle.com/mansoordaku/ckdisease
* https://cloud.ibm.com/login
* https://www.ibm.com/cloud/get-started
* https://developer.ibm.com/tutorials/how-to-create-a-node-red-starter-application/
* https://eu-gb.dataplatform.cloud.ibm.com/
* https://developer.ibm.com/technologies/machine-learning/series/learning-path-machine-learning-for-developers/
* https://bookdown.org/caoying4work/watsonstudio-workshop/jn.html
* https://developer.ibm.com/tutorials/watson-studio-auto-ai/
* https://bookdown.org/caoying4work/watsonstudio-workshop/auto.html#add-asset-as-auto-ai
* https://github.com/SmartPracticeschool/llSPS-INT-2932-Chronic-kidney-disease-prediction-using-Watson-Auto-AI
* https://node-red-efrxp.eu-gb.mybluemix.net/red/#flow/8dce98dd.a01768

Sample Code

1.AutoAI experiment metadata

This cell contains input parameters provided to run the AutoAI experiment in Watson Studio and COS credentials required to retrieve AutoAI pipeline.

**from** **watson\_machine\_learning\_client.helpers** **import** DataConnection, S3Connection, S3Location

experiment\_metadata = dict(

prediction\_type='classification',

prediction\_column='classification',

test\_size=0.1,

scoring='accuracy',

max\_number\_of\_estimators=2,

training\_data\_reference = [DataConnection(

connection=S3Connection(

api\_key='vAQF-rcDWOzc8PJR9A69L5ekZXo63Dsi5haobRriYIxg',

auth\_endpoint='https://iam.bluemix.net/oidc/token/',

endpoint\_url='https://s3.eu-geo.objectstorage.softlayer.net'

),

location=S3Location(

bucket='chronickidneydiseaseprediction-donotdelete-pr-fhdoszggrjnq8d',

path='datasets\_1111\_2005\_kidney\_disease.csv'

))

],

training\_result\_reference = DataConnection(

connection=S3Connection(

api\_key='vAQF-rcDWOzc8PJR9A69L5ekZXo63Dsi5haobRriYIxg',

auth\_endpoint='https://iam.bluemix.net/oidc/token/',

endpoint\_url='https://s3.eu-geo.objectstorage.softlayer.net'

),

location=S3Location(

bucket='chronickidneydiseaseprediction-donotdelete-pr-fhdoszggrjnq8d',

path='auto\_ml/034f1b5e-1c4e-4caa-8da4-ff6905f76f23/wml\_data/5d4068b6-485a-4272-924d-2cbbd95a58d5/data/automl',

model\_location='auto\_ml/034f1b5e-1c4e-4caa-8da4-ff6905f76f23/wml\_data/5d4068b6-485a-4272-924d-2cbbd95a58d5/data/automl/pre\_hpo\_d\_output/Pipeline5/model.pickle',

training\_status='auto\_ml/034f1b5e-1c4e-4caa-8da4-ff6905f76f23/wml\_data/5d4068b6-485a-4272-924d-2cbbd95a58d5/training-status.json'

)

))

pipeline\_name='Pipeline\_5'

1. **Pipeline inspection**

In this section you will get the trained pipeline model from the AutoAI experiment and inspect it.  
You will see pipeline as a pythone code, graphically visualized and at the end, you will perform a local test.

1. **Get historical optimizer instance**

The next cell contains code for retrieving fitted optimizer.

In [ ]: **from** **watson\_machine\_learning\_client.experiment** **import** AutoAI

optimizer = AutoAI().runs.get\_optimizer(metadata=experiment\_metadata)

1. **Get pipeline model**

The 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'.

In [ ]pipeline\_mode=optimizer.get\_pipeline(pipeline\_name = pipeline\_name)

pipeline\_model.pretty\_print(combinators=**False**, ipython\_display=**True**)

pipeline\_model.visualize()

**Read training and holdout data**

Retrieve training dataset from AutoAI experiment as pandas DataFrame.

training\_df, holdout\_df = optimizer.get\_data\_connections()[0].read(with\_holdout\_split=**True**)

train\_X = training\_df.drop([experiment\_metadata['prediction\_column']], axis=1).values

train\_y = training\_df[experiment\_metadata['prediction\_column']].values

test\_X = holdout\_df.drop([experiment\_metadata['prediction\_column']], axis=1).values

y\_true = holdout\_df[experiment\_metadata['prediction\_column']].values

1. **Test pipeline model locally**

**Note**: you can chose the metric to evaluate the model by your own, this example contains only a basic scenario.

**from** **sklearn.metrics** **import** accuracy\_score

predictions = pipeline\_model.predict(test\_X)

score = accuracy\_score(y\_true=y\_true, y\_pred=predictions)

print('accuracy\_score: ', score)

## Deploy and Score

### Connect to WML client in order to create deployment[¶](https://eu-gb.dataplatform.cloud.ibm.com/data/notebooks/converter/assets/feb7b39c-7690-43ef-ba1f-60104b3a560c?project=29a14871-15d1-453f-b5a6-c7cd346a163e#Connect-to-WML-client-in-order-to-create-deployment)

wml\_credentials = {

"apikey": "",

"iam\_apikey\_description": "",

"iam\_apikey\_name": "",

"iam\_role\_crn": "r",

"iam\_serviceid\_crn": "",

"instance\_id": "",

"url": ""

}

Create deployment

**from** **watson\_machine\_learning\_client.deployment** **import** WebService

service = WebService(wml\_credentials)

service.create(

model=pipeline\_model,

metadata=experiment\_metadata,

deployment\_name=f'**{**pipeline\_name**}**\_webservice'

)

print(service)

service.get\_params()

### Score webservice

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

predictions