AS226429

Legacy Data Analysis on Web with Forge – Cost Prediction in Initial Design Stage.

TAKAHIRO Oka
NIHONSEKKEI, INC.

Learning Objectives

- Describe effective cost management workflow at initial design stage for BIM project.
- Strategy of Utilization legacy project data with WebAPI of internal systems.
- Picking up the data of on-going project on Web using Forge.
- Data Analysis method for decision-making with “Scikit-learn” that is library of Python.

Description

The class presents cost management strategy using web based system that includes Forge.

In the system, we focused to analyze legacy project data interactively for on-going project with Machine Learning Algorithm In an attempt of 5D BIM.

It supports decision-making of cost control at the initial design stage when Revit model don't have detail object for cost estimation.

By improving quality of cost estimate in initial design stage, it is easy to manage the cost within target budget through project.

We will focus on as billow;

a. How to get past project data interactively through WebAPI of internal system.

b. How to get and share on-going project data with "Data management API " and "Viewer" of Forge.

c. Data analysis techniques for cost estimation of on-going project using "scikit-learn" that is machine learning library in Python.

d. Data cycle system to improve quality of analysis, constantly.

This session will explore the above technique as it has been used for actual building projects.

Your AU Expert

TAKAHIRO Oka is the Software Engineer for the NIHON SEKKEI, Inc[1], where he has developing plug-in with Revit API for Revit, Web API for internal system corporation, analyzed project data for cost estimation.

He studied spatial perception for the exhibition space using the eye tracking and VR device at the Chiba University.

He began using Revit and Dynamo from 2016 after graduation.
Currently, He has developed various tools focusing on how to use legacy project data for on-going projects with web technology and data analysis method in NIHON SEKKEI, Inc.

Forge Introduction

About Forge APIs

Forge[2] is a set of web services has been provided by Autodesk, and It can be used from any application that can have environment to use HTTP protocol.

Currently, provided services are:

- OAuth
- Data Management API
- Design Automation API
- Model Derivative API
- Viewer
- Reality Capture API
- BIM360 API
- Webhooks API

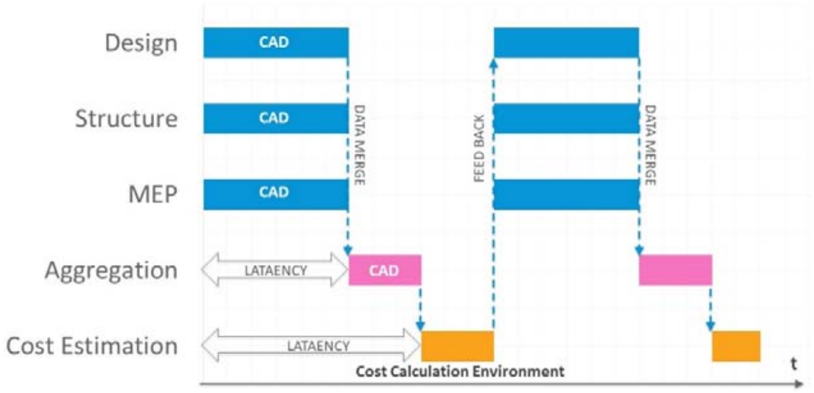
Platform Combination

Data Management API and BIM360 API are enable to access project data saved in Autodesk's PaaS service such as BIM 360 DOCS from whatever programable environment that can use HTTP protocol. It means that project data can be collaborated with other platform's API for application of one's own making and it is generally called "API Economy".

How Changes in Data Management Process by BIM and PaaS ?

Legacy Process of Data Management

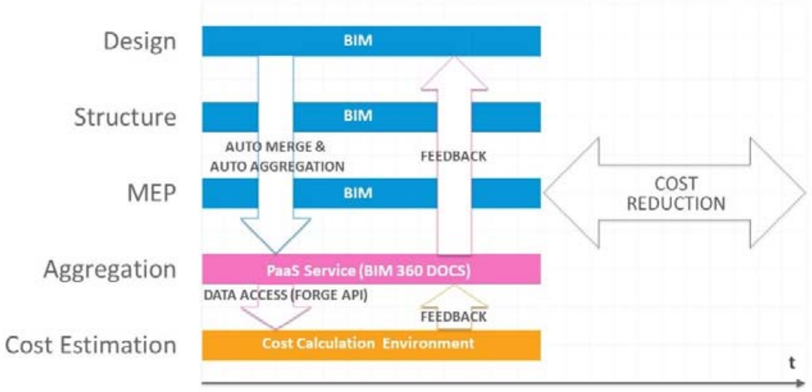
The figure below describe ever process of project data management. In this process, project data of each section such as Design, Structure, MEP, etc. had been only unified at break of each design phases, so adjustment of design beyond section had been secured by communication and strict workflow.
In this process, data are always synchronous, so some process has to wait to use the data until the other process finish using them. As the result, the throughput decrease.



Legacy Process Pipeline

Current Process of Data Management

The figure below describe current process of project data management. In this process, project data are uploaded to Autodesk cloud service like BIM 360 DOCS every specified time and task. It means that unified project data can be accessed every time for adjustment of design beyond section and You can always do your own analysis on that data by your Forge Application.
In this process. We don't need to wait finishing of the other process, so the throughput increase.



Current Process Pipeline

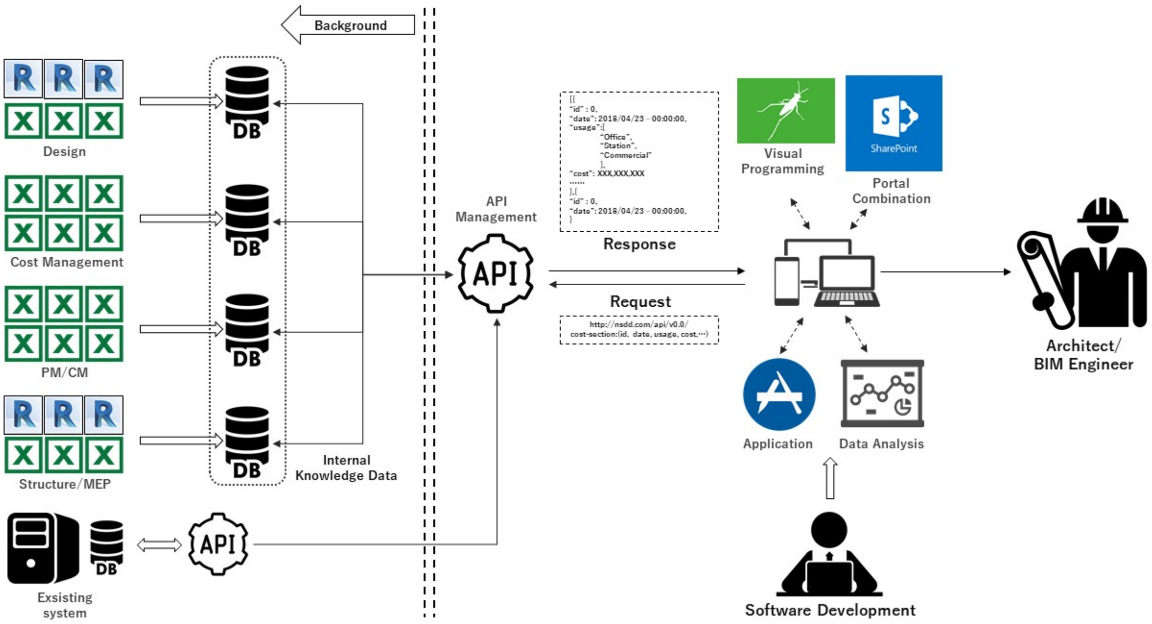
Accumulated Data Demand and How to use them.

Demand

Recently, growth of Machine Learning technology and API Economy accelerate increasing demand of Big Data, and based on the data, various services are publishing. As example, picture that take people were used for Machine Learning to generate analysis model of detecting people face, and the model have been incorporating into camera of smart phone device.

How to Use Accumulated Project Data

NIHONSEKKEI have been building a system to access accumulated project data from various platform easily, which called NS API. We can access accumulated project data with HTTP protocol by NS API which is as Web API like Forge API, and this system accelerate collaboration with each internal systems and application development , data analysis, etc.



```
[{
"id" : 0,
"date": 2018/04/23 - 00:00:00,
"usage":[
        "Office",
        "Station",
        "Commercial"
        ],
"cost": XXX,XXX,XXX
......
},{
"id" : 0,
"date": 2018/04/23 - 00:00:00,
}
```

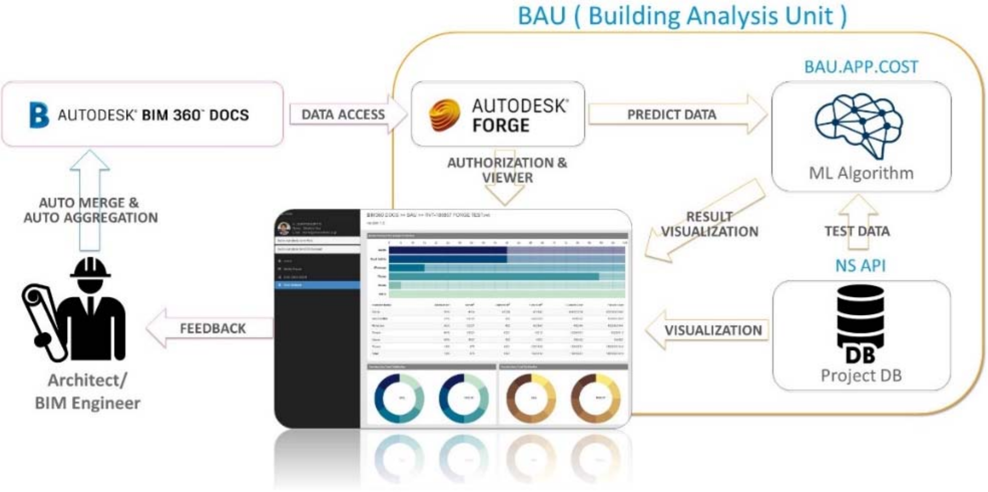
```
http://nsdd.com/api/v0.0/
cost-section:(id, date, usage, cost,…)
```

NS API Overview

About Our Application by Forge

System Overview

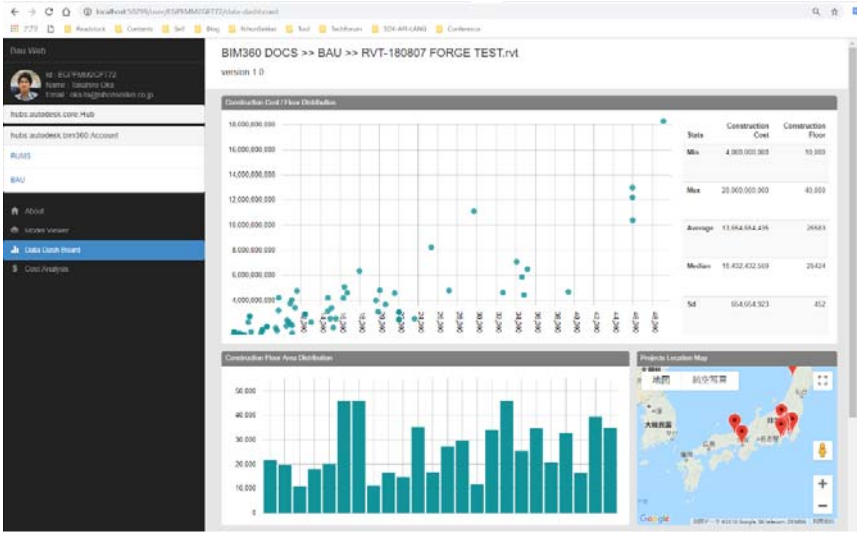
This system is called BAU : Building Analysis Unit.
BAU is provided as internal system. In BAU.WEB, Forge API and NS API is unified as one Web Application. BAU.APP.COST is Application Server to analyze construction cost of ongoing project by machine learning (hereinafter, this is called ML) and ML model are created by learning project data of NS API. BAU support decision-making by to predict construction cost of ongoing project whenever you want.



System Overview

Function 1 – Project Data DashBord

In Dashboard tool, past project data be read from NS API are statistically visualized, and support to make decisions for ongoing project by searching and comparing to reference projects.



BIM360 DOCS >> BAU >> RVT-180807 FORGE TEST.rvt

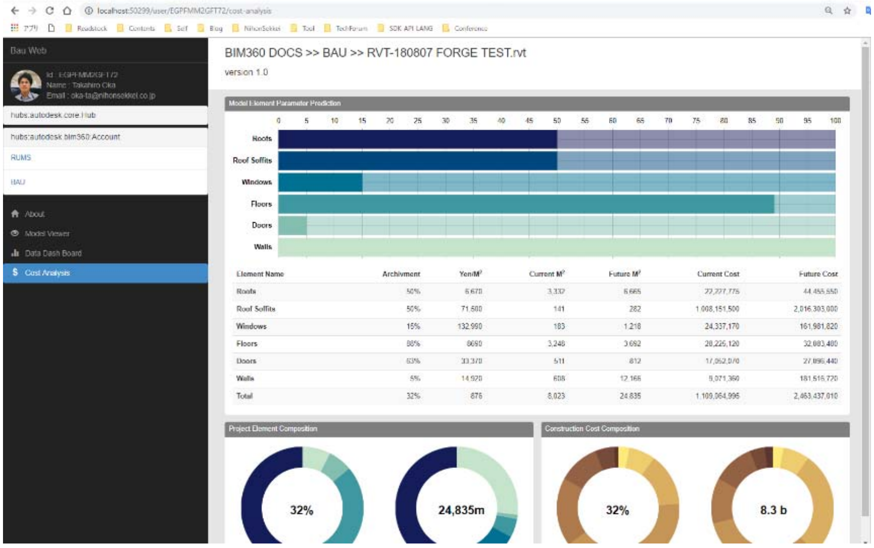
version 1.0

Stats	Construction Cost	Construction Floor
Min	4,000,000,000	10,000
Max	20,000,000,000	40,000
Average	11,654,654,435	26683
Median	10,432,432,569	26424
Sd	654,654,923	452

Dashboard Window Image

Function 2 - Cost Analysis with ML

In this tool, You can check predicted construction cost of ongoing project by posting project data, that is read from Autodesk cloud service by Forge API, with HTTP protocol to BAU.APP.COST. It predict current and future construction cost from project's state by ML.



BIM360 DOCS >> BAU >> RVT-180807 FORGE TEST.rvt

version 1.0

Element Name	Archivment	Yen/M²	Current M²	Future M²	Current Cost	Future Cost
Roofs	50%	6,670	3,332	6,665	22,227,775	44,455,550
Roof Soffits	50%	71,500	141	282	1,008,151,500	2,016,303,000
Windows	15%	132,990	183	1,218	24,337,170	161,981,820
Floors	88%	8690	3,248	3,692	28,225,120	32,083,480
Doors	63%	33,370	511	812	17,052,070	27,096,440
Walls	5%	14,920	608	12,166	9,071,360	181,516,720
Total	32%	876	8,023	24,835	1,109,064,996	2,463,437,010

.

Cost Analysis Window Image

About Front-End Design

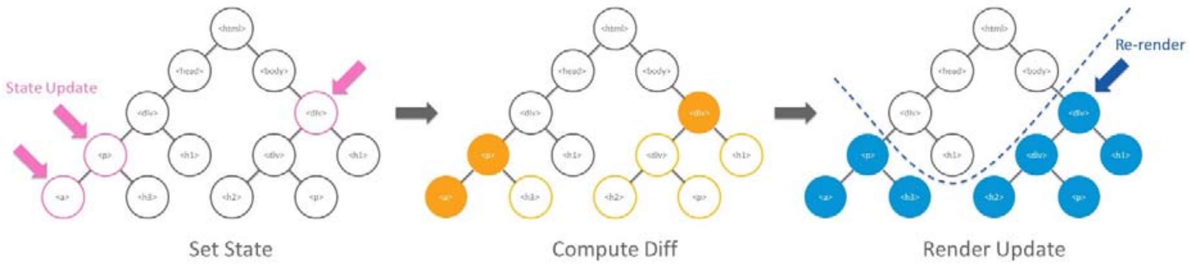
Adapted Flamework and Library

The following technologies are mainly adopted on BAU.WEB.

- **ASP.NET Core**[3] - Open Source Framework to design Cloud Base Application.
- **WebPack**[4] - Module Handler to manage javascript (hereinafter, this is called JS) Library.
- **React**[5] - JS library for building user interface as SPA.
- **Redux**[6] - Redux is a predictable state container for JavaScript apps.
- **React-hot loader**[7] - Tweak React components in real time for development.
- **Bootstrap**[8] - Free and open-source front-end framework for designing web applications.
- **Babel**[9] - Convert ES 2015+ code into a backwards compatible version of JavaScript.
- **Axios**[10] - Promise based HTTP communication library.

With React and Redux for Single Page Application(SPA)

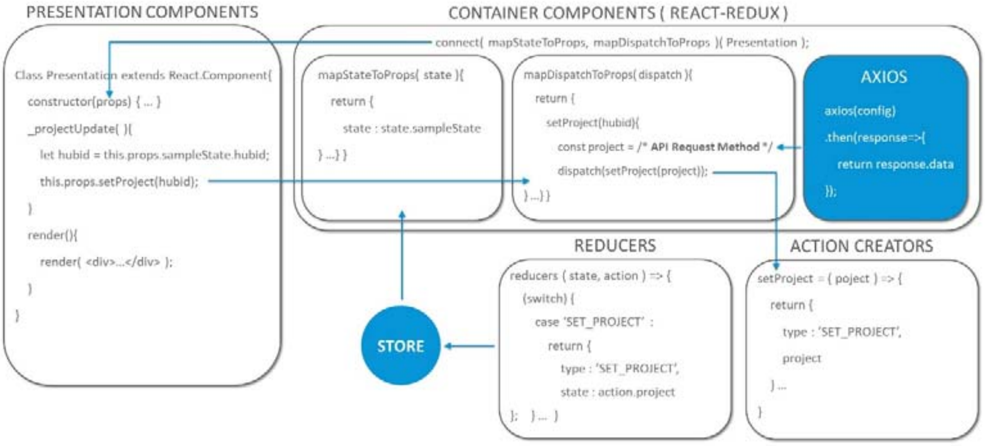
In Many interapplication communication by Forge API, to use React is good for improving user experience because it adapts Virtual-DOM architecture which optimize DOM tree rendering. And Redux is good for manage the complex state of User Interface on Single Page Application.



Virtual DOM

React-Redux Life Cycle Design for API request with Axios

React-Redux is the official React binding for Redux. It lets each React components read data from a Redux store, and dispatch actions to the store to update data. It is good for readability that each React Component and application data is loosely coupled. When you want to communicate to API, I recommend you to write HTTP request code on react-redux’s function like “mapDispachToProps” because to keep readability of React component code and Redux code.



```
connect( mapStateToProps, mapDispatchToProps )( Presentation );
```

```
Class Presentation extends React.Component{
  constructor(props) { ... }
  _projectUpdate( ){
    let hubid = this.props.sampleState.hubid;
    this.props.setProject(hubid);
  }
  render(){
    render( <div>...</div> );
  }
}
```

```
mapStateToProps( state ){
  return {
    state : state.sampleState
} ...} }
```

```
mapDispatchToProps( dispatch ){
  return {
    setProject(hubid){
      const project = /* API Request Method */
      dispatch(setProject(project));
} ...} }
```

```
axios(config)
.then(response=>{
  return response.data
});
```

```
reducers ( state, action ) => {
  (switch) {
    case 'SET_PROJECT' :
      return {
        type : 'SET_PROJECT',
        state : action.project
};  } ... }
```

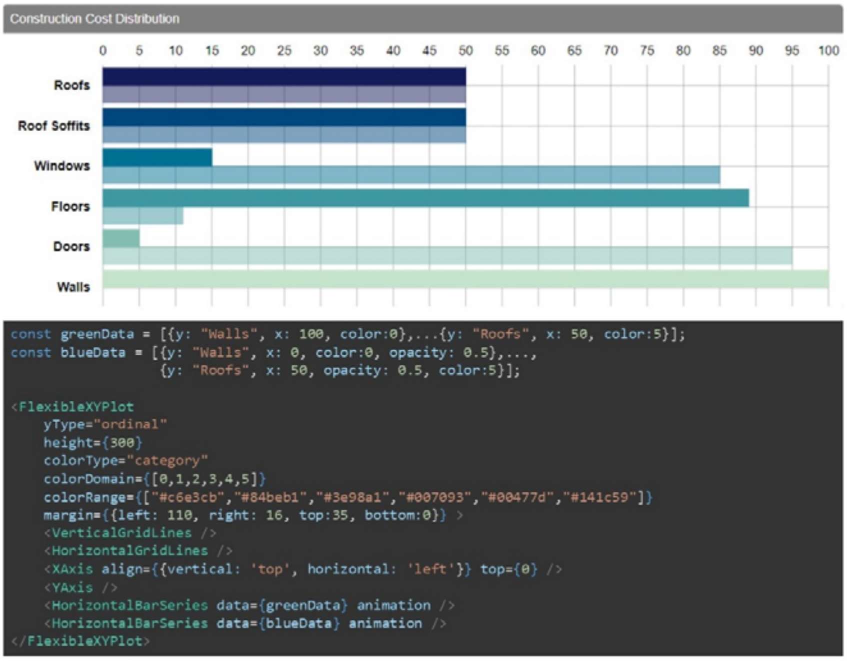
```
setProject = ( poject ) => {
  return {
    type : 'SET_PROJECT',
    project
  } ...
}
```

.

React-Redux Lifecycle with API Request

React-vis for Responsible Data Visualization

To realize responsible data visualization on React app, I have been using “react-vis”[11] which are designed to work just like other React components. It have various type diagrams and can be displayed them easily.

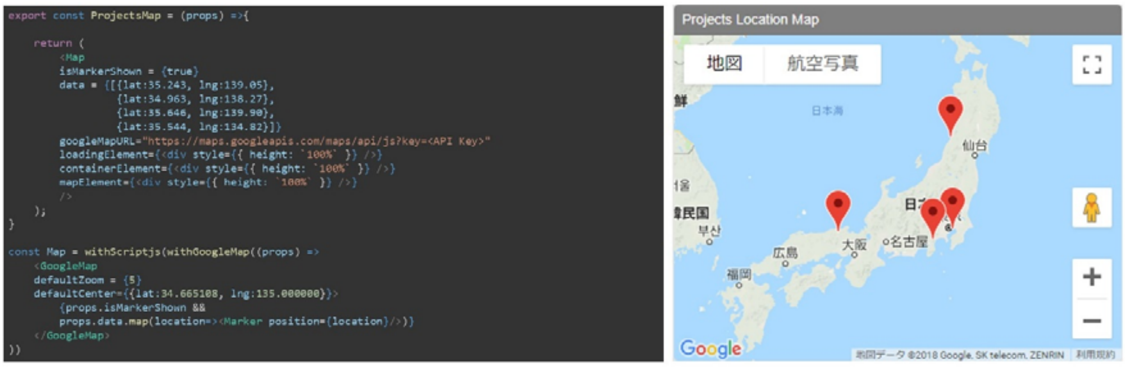


```
const greenData = [{y: "Walls", x: 100, color:0},...{y: "Roofs", x: 50, color:5}];
const blueData = [{y: "Walls", x: 0, color:0, opacity: 0.5},...,
                  {y: "Roofs", x: 50, opacity: 0.5, color:5}];

<FlexibleXYPlot
    yType="ordinal"
    height={300}
    colorType="category"
    colorDomain={[0,1,2,3,4,5]}
    colorRange={["#c6e3cb","#84beb1","#3e98a1","#007093","#00477d","#141c59"]}
    margin={{left: 110, right: 16, top:35, bottom:0}} >
    <VerticalGridLines />
    <HorizontalGridLines />
    <XAxis align={{vertical: 'top', horizontal: 'left'}} top={0} />
    <YAxis />
    <HorizontalBarSeries data={greenData} animation />
    <HorizontalBarSeries data={blueData} animation />
</FlexibleXYPlot>
```

Sample Code of ‘react-vis’ with jsx

React-google-maps for Combination of React and Google Maps API

To map projects location for displaying on Web browser, we always use “google-map”. “React-google-maps”[[12] is wrapper of “Google Maps javascript API”[13] for React App. It also work just like other React Component.



```
export const ProjectsMap = (props) =>{

    return (
        <Map
        isMarkerShown = {true}
        data = {[{lat:35.243, lng:139.05},
                 {lat:34.963, lng:138.27},
                 {lat:35.646, lng:139.90},
                 {lat:35.544, lng:134.82}]}
        googleMapURL="https://maps.googleapis.com/maps/api/js?key=<API Key>"
        loadingElement={<div style={{ height: `100%` }} />}
        containerElement={<div style={{ height: `100%` }} />}
        mapElement={<div style={{ height: `100%` }} />}
        />
    );
}

const Map = withScriptjs(withGoogleMap((props) =>
    <GoogleMap
    defaultZoom = {5}
    defaultCenter={{lat:34.665108, lng:135.000000}}>
        {props.isMarkerShown &&
        props.data.map(location=><Marker position={location}/>)}
    </GoogleMap>
))
```

Sample Code of ‘react-google-maps’ with jsx

About Back-End Design for Data Analysis

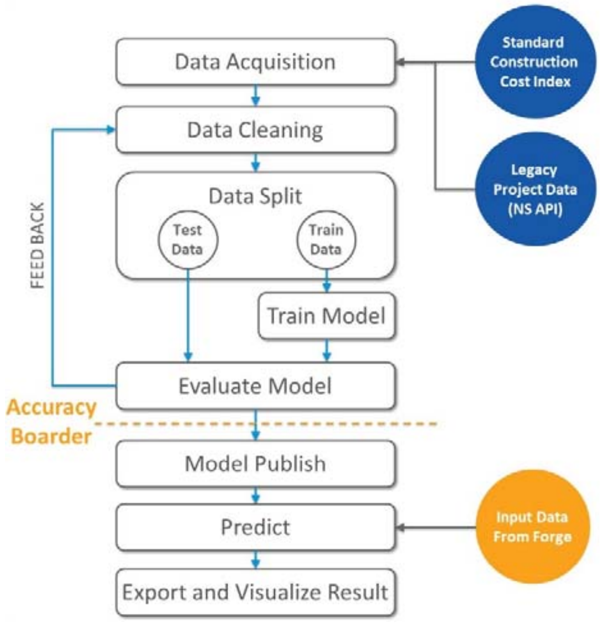
Adapted Flamework and Library

Python have many library for data analysis and machine learning. I have been using Flask, which is server framework , as application server for using these library. The following technologies are adopted on BAU.APP.COST.

- **Flask**[14] - Flask is a micro web framework written in Python.
- **scikit-learn**[15] - It provides simple and efficient tools for data mining and data anlysis.
- **Jupyter Notebook**[16] - It is an open-source web application that allows you to create and share documents that contain live code, equations, visualizations and narrative text.
- **Matplotlib**[17] - It is a Python 2D plotting library.

Process of ML Training, Testing and Leveraging.

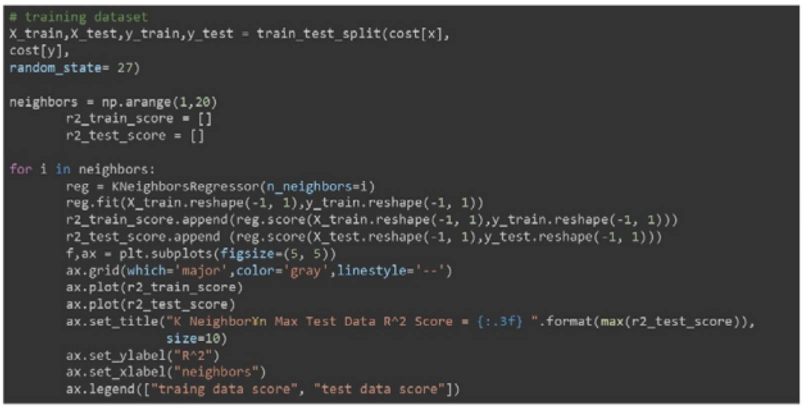
For making cost prediction model with ML algorithm such as "scikit-learn", traing data and test data are needed for adjusting those parameter. In this system, data are past project data, which requested from NS API, and Standard Construction Cost Index that reported from *KensetsuKogyoKeieiKenkyuka*i, which is research institute about construction cost. Data are cleaned to improve accuracy of model, and used as training data and test data. created model is incorporating into application when model's accuracy tested by test data and training data are over some value.



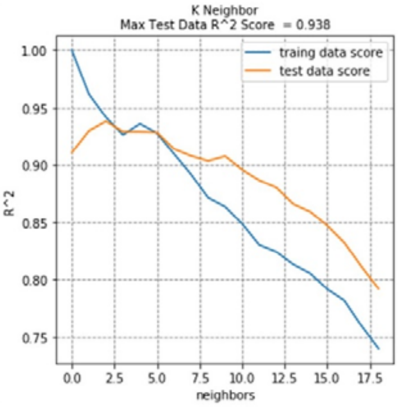
Process for using ML model

Scikit-learn for Machine Learning

This application have been using “scikit-learn” to analysis project data. it provides some algorithm class : Generalized Linear Models, Support Vector Machines, Nearest Neighbors, etc. By using these algorithm class, you can create analytical model and evaluate their accuracy easily. These process is done on jupyter Notebook which is open source application to support management of code and document for data analysis.



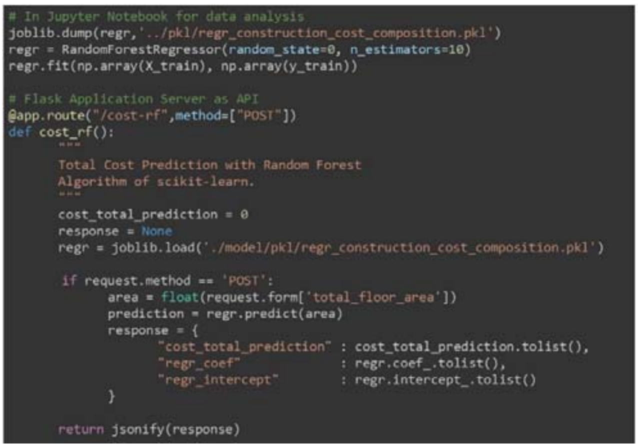
```python
# training dataset
X_train,X_test,y_train,y_test = train_test_split(cost[x],
cost[y],
random_state= 27)

neighbors = np.arange(1,20)
        r2_train_score = []
        r2_test_score = []

for i in neighbors:
        reg = KNeighborsRegressor(n_neighbors=i)
        reg.fit(X_train.reshape(-1, 1),y_train.reshape(-1, 1))
        r2_train_score.append(reg.score(X_train.reshape(-1, 1),y_train.reshape(-1, 1)))
        r2_test_score.append (reg.score(X_test.reshape(-1, 1),y_test.reshape(-1, 1)))
        f,ax = plt.subplots(figsize=(5, 5))
        ax.grid(which='major',color='gray',linestyle='--')
        ax.plot(r2_train_score)
        ax.plot(r2_test_score)
        ax.set_title("K Neighbor¥n Max Test Data R^2 Score = {:.3f} ".format(max(r2_test_score)),
                    size=10)
        ax.set_ylabel("R^2")
        ax.set_xlabel("neighbors")
        ax.legend(["traing data score", "test data score"])
```



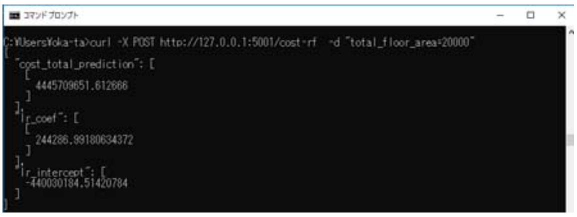
Sample Code of Machine Learning with Python in Jupyter Notebook

Back-End Application Server for Data Analysis

We export the analytical model, which created on Jupyter Notebook , to any directory as ".pkl" files, for we use them on other programs. The files are imported application server and covert to analytical model again.



```python
# In Jupyter Notebook for data analysis
joblib.dump(regr,'../pkl/regr_construction_cost_composition.pkl')
regr = RandomForestRegressor(random_state=0, n_estimators=10)
regr.fit(np.array(X_train), np.array(y_train))

# Flask Application Server as API
@app.route("/cost-rf",method=["POST"])
def cost_rf():
      """
      Total Cost Prediction with Random Forest
      Algorithm of scikit-learn.
      """
      cost_total_prediction = 0
      response = None
      regr = joblib.load('./model/pkl/regr_construction_cost_composition.pkl')

       if request.method == 'POST':
            area = float(request.form['total_floor_area'])
            prediction = regr.predict(area)
            response = {
                  "cost_total_prediction" : cost_total_prediction.tolist(),
                  "regr_coef"             : regr.coef_.tolist(),
                  "regr_intercept"        : regr.intercept_.tolist()
            }

      return jsonify(response)
```



```
C:¥Users¥oka-ta>curl -X POST http://127.0.0.1:5001/cost-rf  -d "total_floor_area=20000"
{
  "cost_total_prediction": [
    [
      4445709651.612666
    ]
  ],
  "r_coef": [
    [
      244286.99180634372
    ]
  ],
  "r_intercept": [
    -440030184.51420784
  ]
}
```

Sample Code of Flask API and scikit-learn algorithm

LINKS

1. https://www.nihonsekkei.co.jp/
2. https://forge.autodesk.com/
3. https://docs.microsoft.com/en-us/aspnet/core/?view=aspnetcore-2.1
4. https://webpack.js.org/
5. https://reactjs.org/
6. https://redux.js.org/
7. https://github.com/gaearon/react-hot-loader
8. https://getbootstrap.com/
9. https://babeljs.io/
10. https://github.com/axios/axios
11. https://uber.github.io/react-vis/
12. https://github.com/tomchentw/react-google-maps
13. https://developers.google.com/maps/documentation/javascript/tutorial
14. http://flask.pocoo.org/
15. http://scikit-learn.org/stable/
16. http://jupyter.org/
17. https://matplotlib.org/