



Learning Progress Review

Data Science Bootcamp Week 14

OMICRON

Omicron Team Members



Muhammad Fikri Fadila

<https://www.linkedin.com/in/muhammad-fikri-fadila-a551161a6/>



Anugrah Yazid Ghani

<https://www.linkedin.com/in/anugrah-yazid-7253bb221/>



Edo M Hadad Gibran

<https://www.linkedin.com/in/edo-gibran-38505a142/>



DigitalSkola

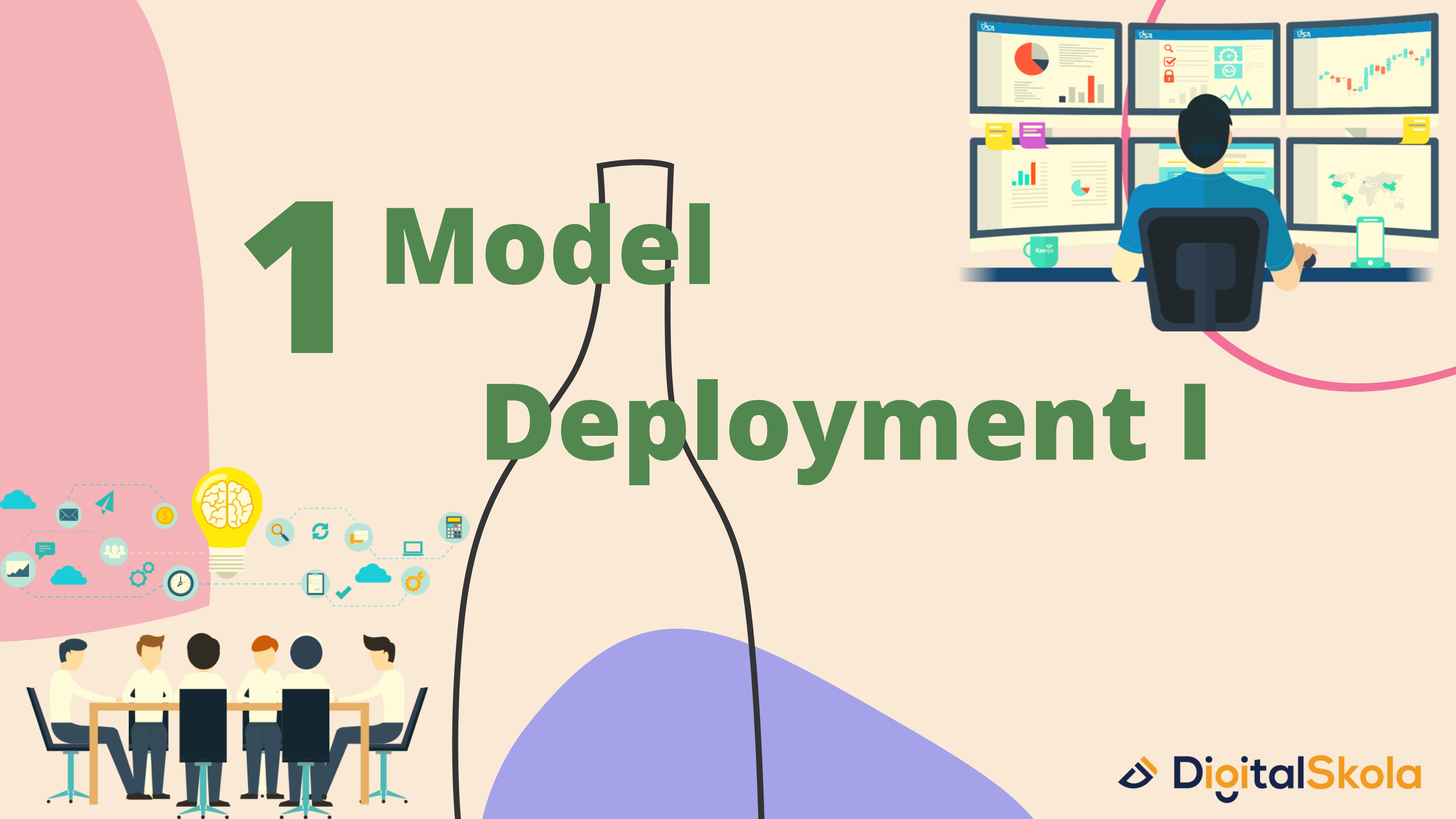
Content

1 Model Deployment I

2 Model Deployment II

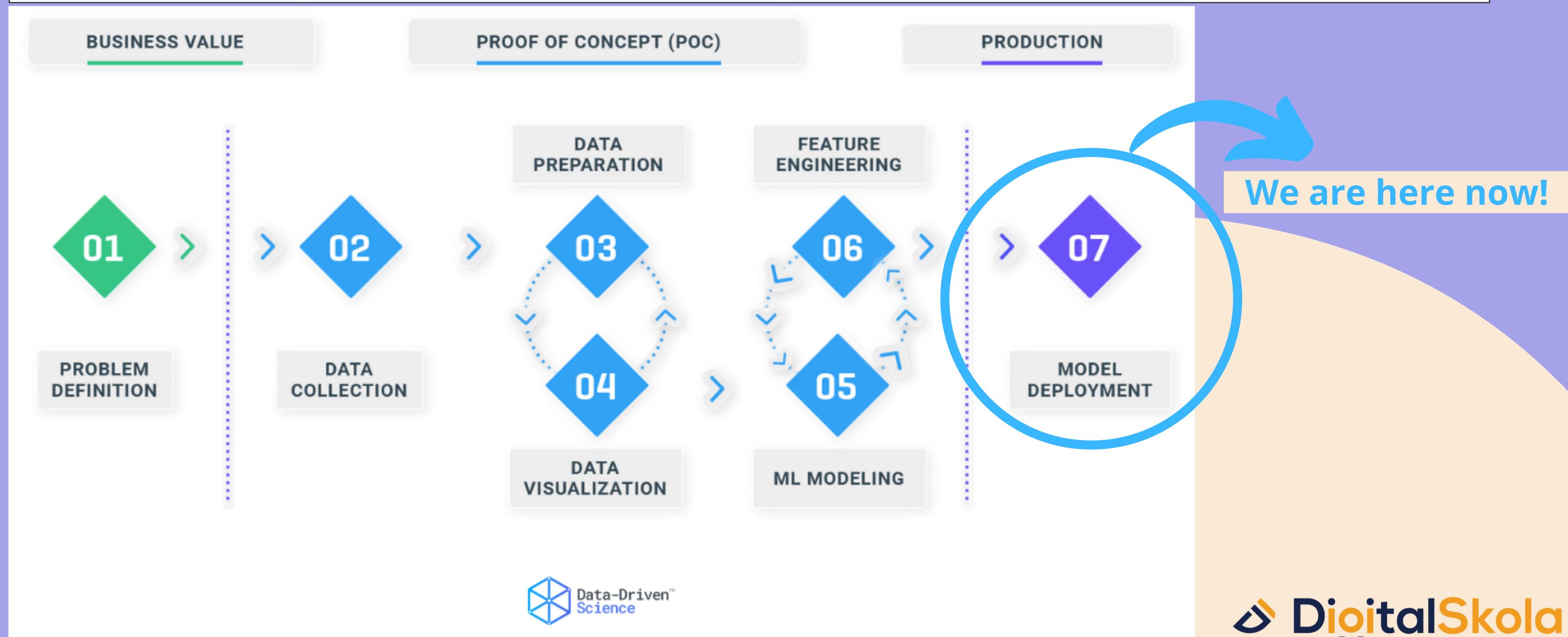
3 Business Intelligence

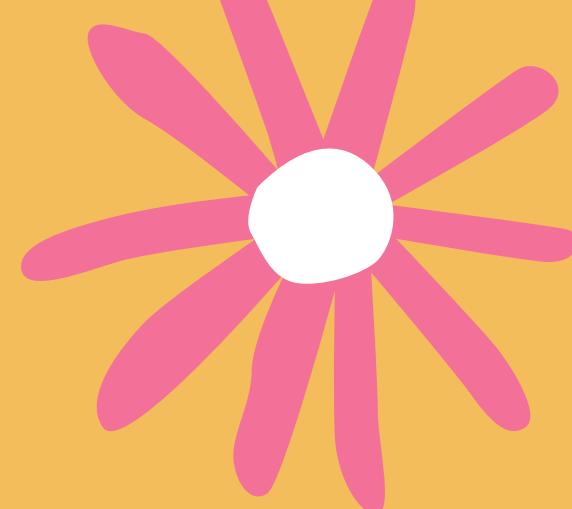
1 Model Deployment



What is Model Deployment?

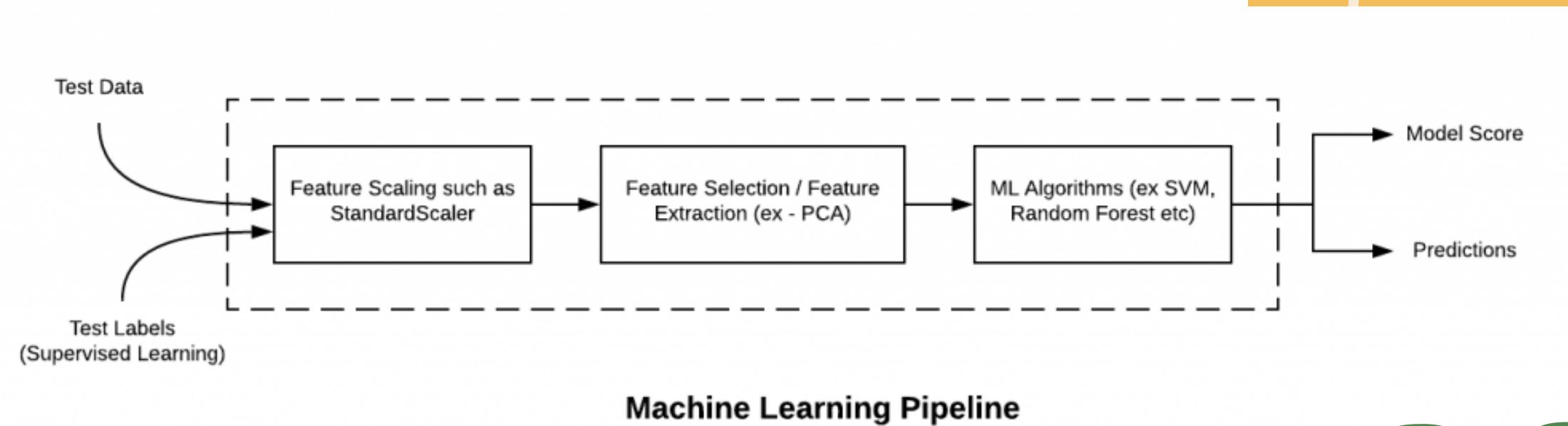
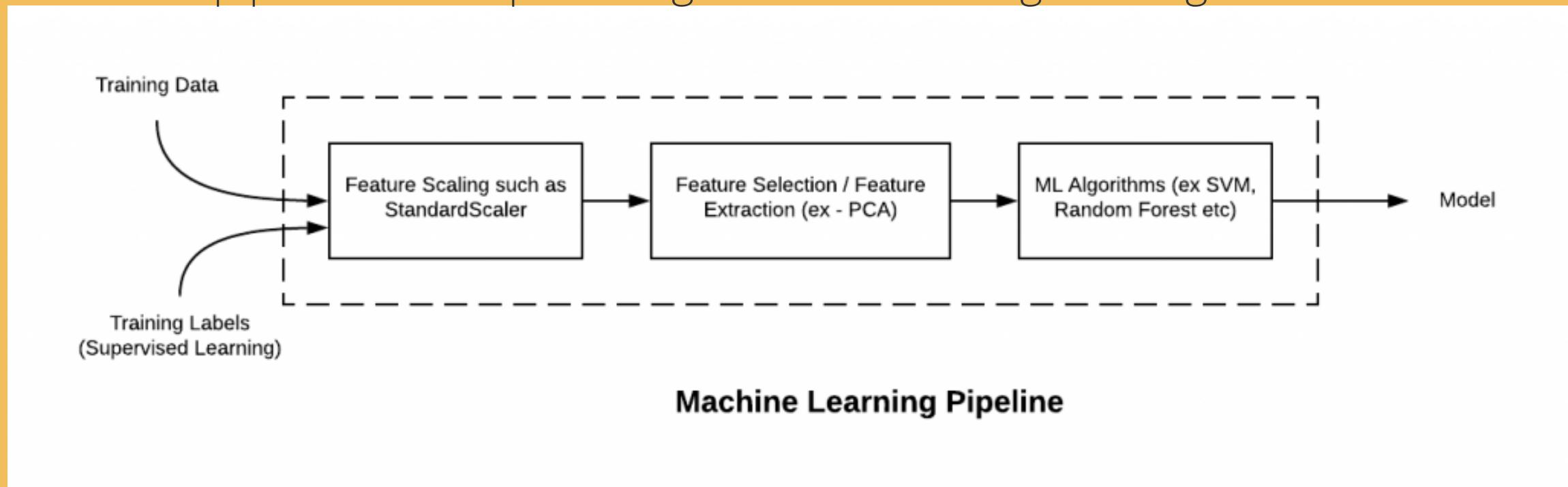
the method by which you integrate a machine learning model into an existing production environment to make practical business decisions based on data.





Machine Learning Pipeline

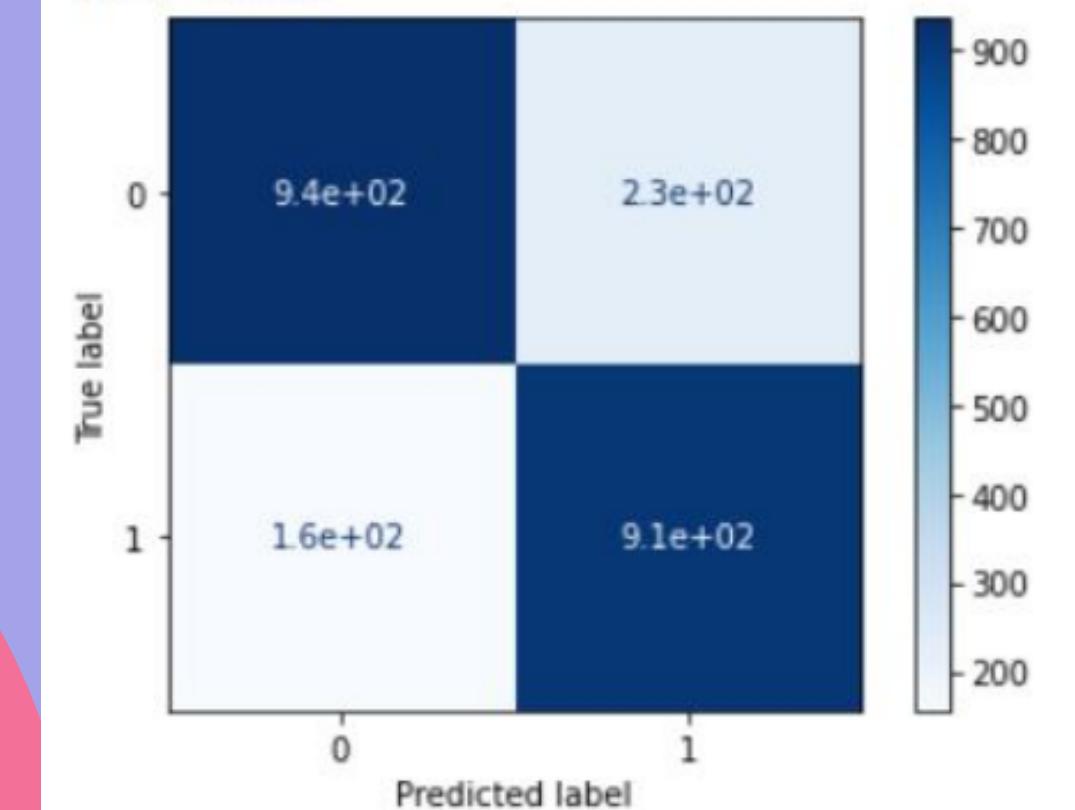
- Machine Learning (ML) pipeline = cara meng-otomatisasi alur kerja yang diperlukan untuk menghasilkan model machine learning.
- Sebelumnya kita men-transformasi data untuk training dan testing set secara terpisah.
- Dengan menggunakan Sklearn.pipeline kita dapat meng-otomatiskan langkah-langkah ini.



How to Use Pipeline

```
x_train, X_test, y_train, y_test = train_test_split(  
    df_X, df_y, test_size=0.2, random_state=42)  
  
numerical_transformer = SimpleImputer(strategy='median')  
categorical_transformer = Pipeline(steps=[  
    ('imputer', SimpleImputer(strategy='most_frequent')),  
    ('imput', OrdinalEncoder())  
])  
preprocessor = ColumnTransformer(  
    transformers=[  
        ('num', numerical_transformer, nums),  
        ('cat', categorical_transformer, cats)  
    ])  
pipeline = Pipeline(steps=[('preprocessor', preprocessor), ('scaling', StandardScaler()),  
    #('feature_selection', SelectFromModel(ExtraTreesClassifier(random_state=2), prefit=False)),  
    #('balance', SMOTE()),  
    ('classifier', RandomForestClassifier(random_state=42, max_depth=10))])  
  
pipeline = pipeline.fit(x_train, y_train)  
y_pred = pipeline.predict(X_test)  
print('')  
print('-----')  
print('Accuracy ', metrics.accuracy_score(y_test, y_pred))  
print('Precision ', metrics.precision_score(y_test, y_pred, average='macro'))  
print('Recall ', metrics.recall_score(y_test, y_pred, average='macro'))  
print('Confusion matrix ', metrics.confusion_matrix(y_test, y_pred))  
plot_confusion_matrix(pipeline, X_test, y_test, cmap=plt.cm.Blues)  
plt.show()
```

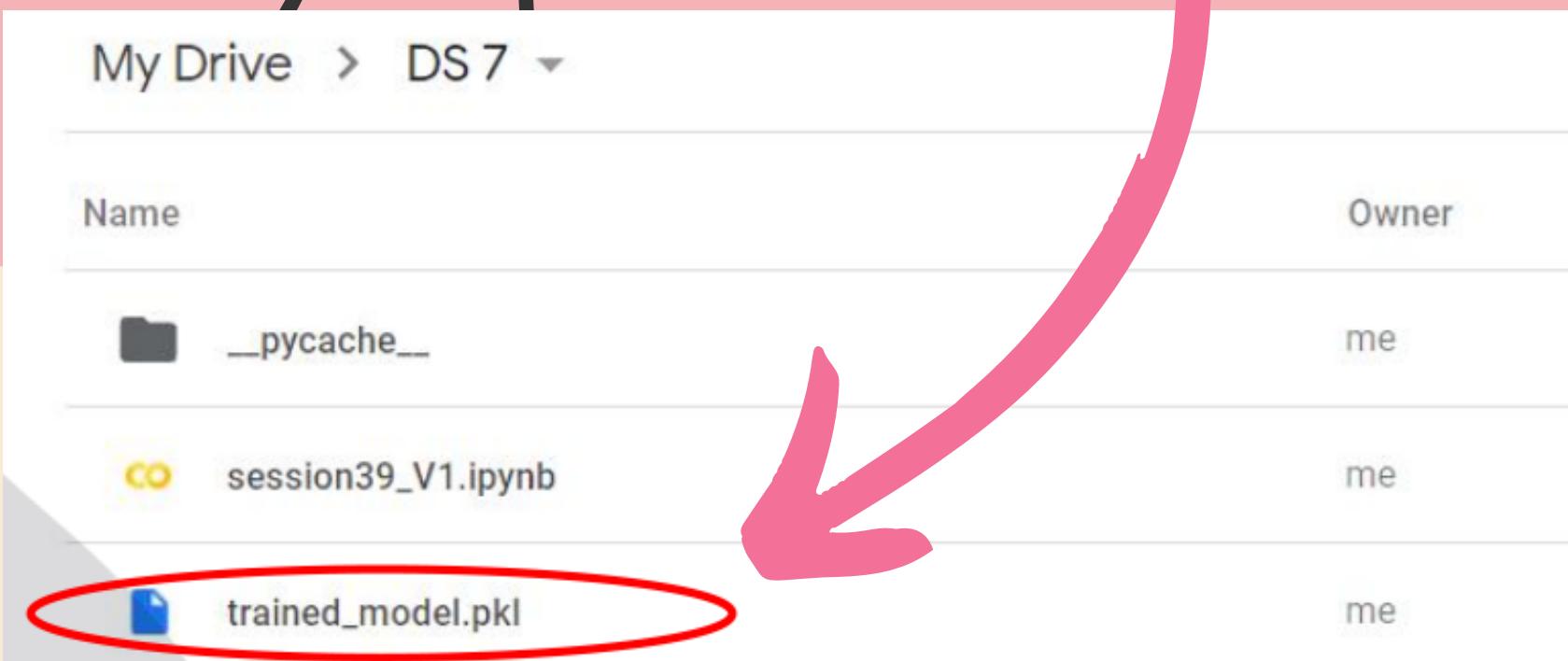
Accuracy 0.8266905508284819
Precision 0.827367604071523
Recall 0.8278412406500327
Confusion matrix [[935 231]
[156 911]]



Save & Load Trained Model

```
pipeline = Pipeline(steps=[('preprocessor', preprocessor), ('scaling', StandardScaler()),  
                         #('feature_selection', SelectFromModel(ExtraTreesClassifier(random_state=2), prefit=False)),  
                         #('balance', SMOTE()),  
                         ('classifier', RandomForestClassifier(random_state=42, max_depth=10))])  
  
pipeline = pipeline.fit(X_train, y_train)  
filename = '/content/drive/MyDrive/DS 7/trained_model.pkl'  
joblib.dump(pipeline, filename)  
  
['/content/drive/MyDrive/DS 7/trained_model.pkl']
```

SAVE



```
import joblib  
import pandas as pd  
  
filename = '/content/drive/MyDrive/DS 7/trained_model.pkl'  
loaded_model = joblib.load(filename)  
  
data = {'age':68, 'job':'admin.', 'marital':'married', 'education':'secondary',  
        'default':'no', 'balance':2000, 'housing':'yes', 'loan':'no', 'contact':'unknown',  
        'day':5, 'month':'may', 'duration':1000, 'campaign':1, 'pdays':-1,  
        'previous':0, 'poutcome':'unknown'}  
df_input = pd.DataFrame(data, index=[0])  
result = loaded_model.predict(df_input)  
  
for i in result:  
    int_result = int(i)  
    if (int_result == 0):  
        decision = 'No'  
    elif (int_result==1):  
        decision = 'Yes'  
    else:  
        decision = 'Not defined'  
  
print('Possibility to deposit is ', decision)
```

Possibility to deposit is Yes

LOAD



Flask

web development,
one drop at a time

- Flask adalah sebuah web framework.
- Artinya Flask menyediakan tools, libraries dan technologies yang memungkinkan kita membangun aplikasi web.
- Ada banyak framework di Python, termasuk Flask, Tornado, Pyramid, dan Django.
- **Note** : mulai di step ini, kita membuat file python di computer local, bukan di google colab.
- Bisa menggunakan editor untuk python, contohnya : spyder (anaconda package)

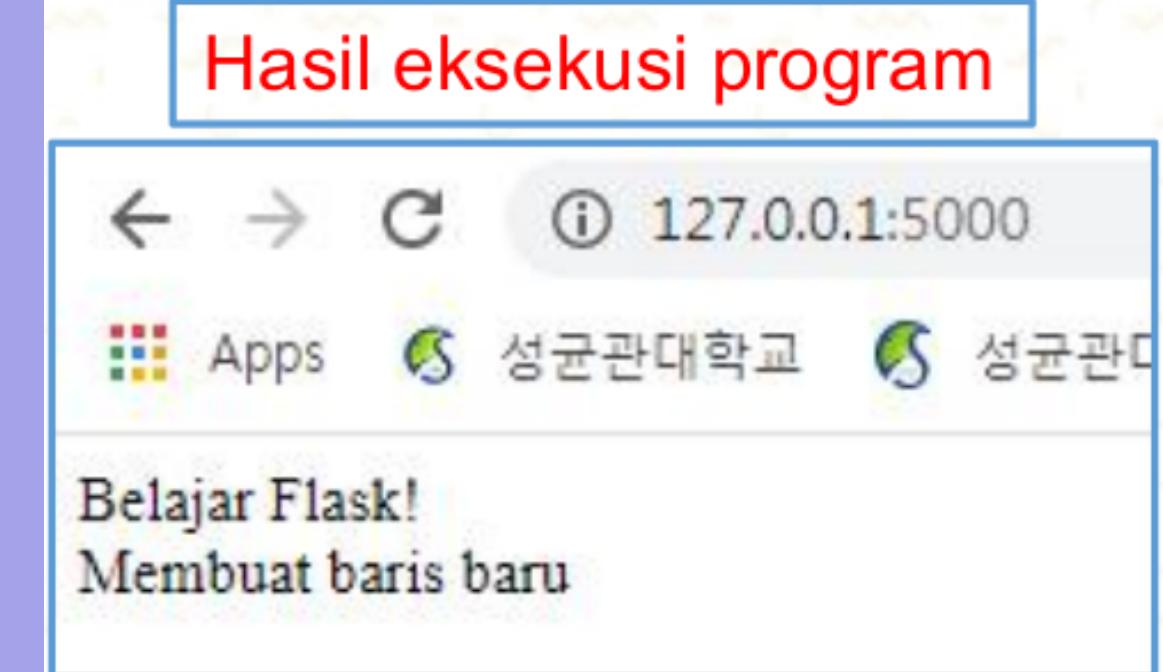


```
1 from flask import Flask
2
3 app = Flask(__name__)
4 @app.route('/')
5 def index():
6     return 'Belajar Flask! <br> Membuat baris baru'
7
8 if __name__ == "__main__":
9     app.run()
```

Console 1/A

```
In [8]: runfile('C:/DATA GANJAR/TEACHING/Digital Skola/batch_7/session_39/session_39_shared/create_model/hello_world.py', wdir='C:/DATA GANJAR/TEACHING/Digital Skola/batch_7/session_39/session_39_shared/create_model')
* Serving Flask app "hello_world" (lazy loading)
* Environment: production
WARNING: This is a development server. Do not use it in a production deployment.
Use a production WSGI server instead.
* Debug mode: off
* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
```

Web Development Using Flask

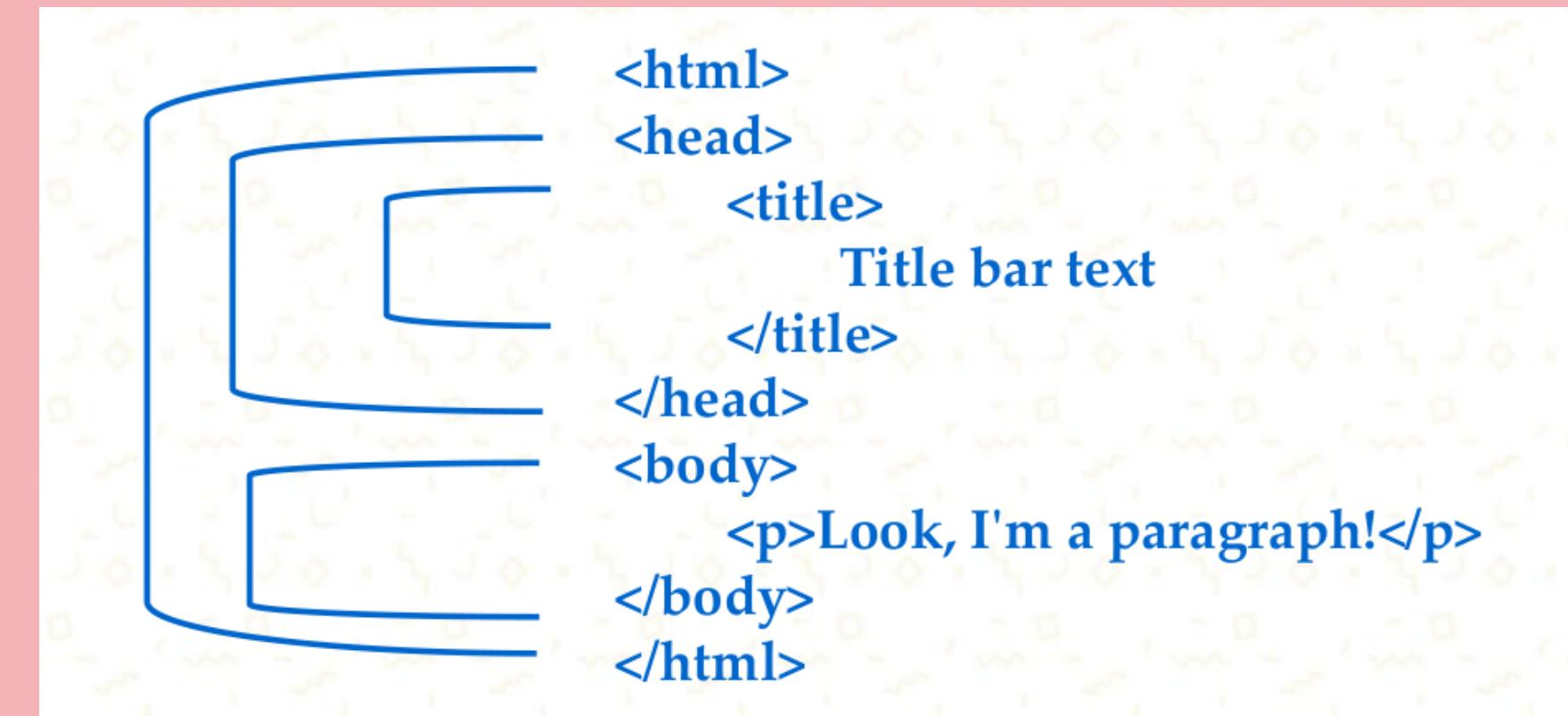


Introduction to HTML

- HTML (HyperText Markup Language)
- Memberikan isi dan struktur dokumen,
- HTML pages menggunakan tags
 - Tags menunjukkan bagaimana program pemrosesan harus menampilkan teks dan grafik
 - Diproses oleh browser
 - Tags biasanya berpasangan
 - Mempunyai nama yang masuk akal

Tag	Description
<html> ... </html>	Declares the Web page to be written in HTML
<head> ... </head>	Delimits the page's head
<title> ... </title>	Defines the title (not displayed on the page)
<body> ... </body>	Delimits the page's body
<h _n > ... </h _n >	Delimits a level <i>n</i> heading
 ... 	Set ... in boldface
<i> ... </i>	Set ... in italics
<center> ... </center>	Center ... on the page horizontally
 ... 	Brackets an unordered (bulleted) list
 ... 	Brackets a numbered list
 ... 	Brackets an item in an ordered or numbered list
 	Forces a line break here
<p>	Starts a paragraph
<hr>	Inserts a horizontal rule
	Displays an image here
 ... 	Defines a hyperlink

HTML Document Layout



HTML: Table

- Tujuan utamanya adalah untuk menampilkan tables
- Table adalah koleksi dari baris (rows)
- Baris (rows) adalah koleksi dari cells
- Cells dapat diisi apa saja (bahkan tables lain)

```


| </th>                            // Header 1 <th>&lt;/th&gt;                            // Header 2 </th> | </th>                            // Header 2 |
|-----------------------------------------------------------------------------------------------------------|----------------------------------------------|
| </td>                            // Column 1 <td>&lt;/td&gt;                            // Column 2 </td> | </td>                            // Column 2 |



|               |               |
|---------------|---------------|
| Row 1, Cell 1 | Row 1, Cell 2 |
| Row 2, Cell 1 | Row 2, Cell 2 |


```

Diagram illustrating the structure of the provided HTML code:

- The first code block shows a simple table structure with two rows. The first row contains two header cells (). The second row contains two data cells (). | |
- The second code block shows a table with a border of 1. It has two rows. The first row has a gray background color and contains two cells. The second row has an orange background color (#FF3399) and also contains two cells.
- Below the second code block is a diagram showing the resulting table structure. It consists of two rows:
 - Row 1:** Contains two cells: "Row 1, Cell 1" and "Row 1, Cell 2".
 - Row 2:** Contains two cells: "Row 2, Cell 1" and "Row 2, Cell 2".

Example:

```

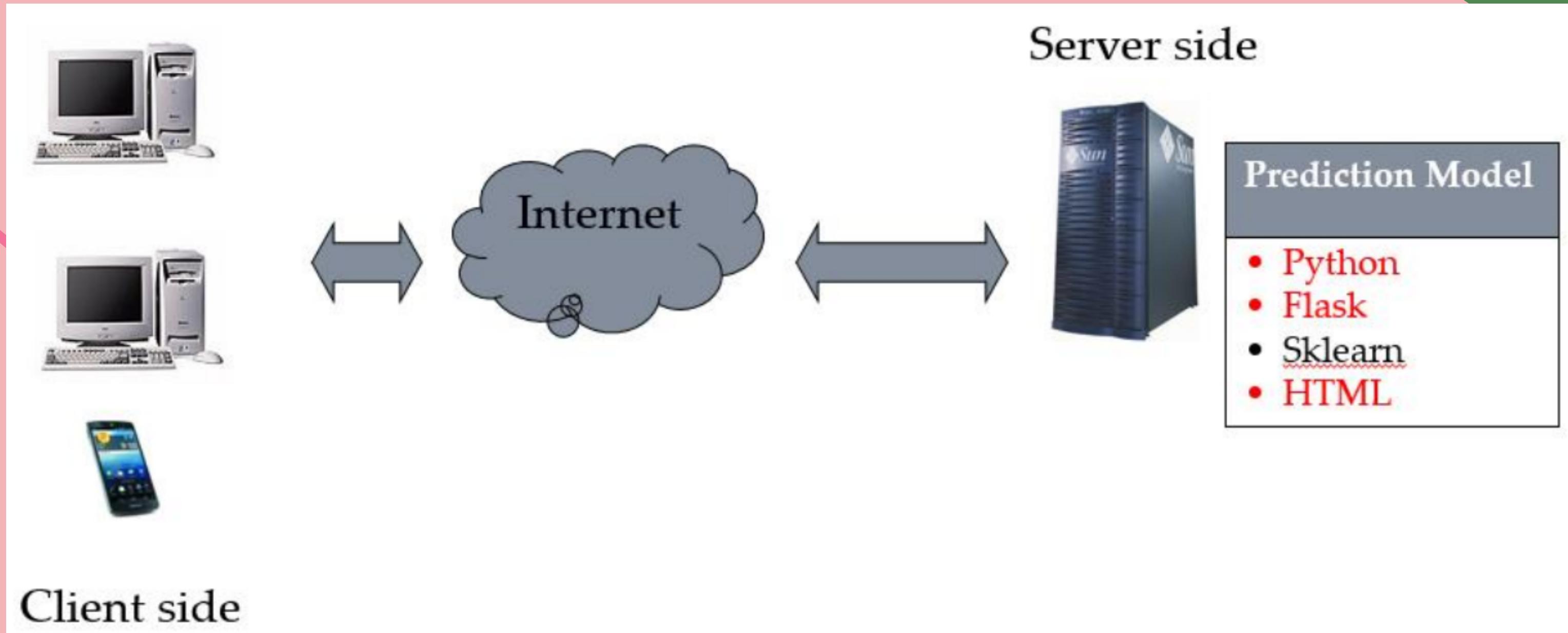

| Name       | Telephone  |            |
|------------|------------|------------|
| Bill Gates | 555 77 854 | 555 77 855 |


```

Name	Telephone
Bill Gates	555 77 854 555 77 855

Integrating ML Model into Web App

Proposed System

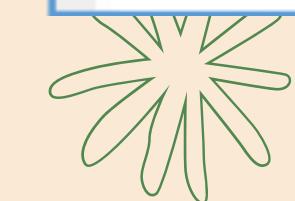


Integrating ML Model into Web App

Save Trained Model into File

```
X_train, X_test, y_train, y_test = train_test_split(  
    df_X, df_y, test_size=0.2, random_state=42)  
  
numerical_transformer = SimpleImputer(strategy='median')  
categorical_transformer = Pipeline(steps=[('imputer', SimpleImputer(strategy='most_frequent')),  
                                         ('imput', OrdinalEncoder())])  
  
preprocessor = ColumnTransformer(  
    transformers=[  
        ('num', numerical_transformer, nums),  
        ('cat', categorical_transformer, cats)  
    ])  
  
pipeline = Pipeline(steps=[('preprocessor', preprocessor), ('scaling', StandardScaler()),  
                        #('feature_selection', SelectFromModel(ExtraTreesClassifier(random_state=42)),  
                        #('balance', SMOTE()),  
                        ('classifier', RandomForestClassifier(random_state=42, max_depth=10))])  
  
pipeline = pipeline.fit(X_train, y_train)  
filename = 'trained_model.pkl'  
joblib.dump(pipeline, filename)
```

이름	수정한 날짜	유형	크기
bank.csv	21/09/2019 19:16	Microsoft Excel C...	898KB
create_model.py	24/08/2021 15:07	PY 파일	2KB
hello_world.py	25/08/2021 13:30	PY 파일	1KB
index.html	25/08/2021 13:52	HTML 문서	1KB
trained_model.pkl	25/08/2021 13:56	PKL 파일	48,744KB



Integrating ML Model into Web App

Web App (Back-End)

```
import pandas as pd
import joblib
from flask import Flask, redirect, url_for, request, render_template

app = Flask(__name__)
#load index.html/ first page. receive input variable from user
@app.route("/")
def index():
    return render_template('index.html')

#load result.html. the result of prediction is presented here.
@app.route('/result/', methods=["POST"])
def prediction_result():
    #receiving parameters sent by client
    age = int(request.form.get('age'))
    job = request.form.get('job')
    marital = request.form.get('marital')
    education = request.form.get('education')
    default = request.form.get('default')
    balance = int(request.form.get('balance'))
    housing = request.form.get('housing')
    loan = request.form.get('loan')
    contact = request.form.get('contact')
    day = int(request.form.get('day'))
    month = request.form.get('month')
    duration = int(request.form.get('duration'))
    campaign = int(request.form.get('campaign'))
    pdays = int(request.form.get('pdays'))
    previous = int(request.form.get('previous'))
    poutcome = request.form.get('poutcome')
    #load the trained model.
    filename = 'trained_model.pkl'
    loaded_model=joblib.load(filename)
    #create new dataframe
    data = {'age':age, 'job':job, 'marital':marital, 'education':education, 'default':default, 'balance':balance, 'housing':housing,
            'loan':loan, 'contact':contact, 'day':day, 'month':month, 'duration':duration, 'campaign':campaign, 'pdays':pdays,
            'previous':previous, 'poutcome':poutcome}
    pd.set_option('display.max_columns', None)
    pd.set_option('display.max_rows', None)
    df_input = pd.DataFrame(data, index=[0])
    #print(df_input.dtypes)
    result = loaded_model.predict(df_input)
    #print(result)
    for i in result:
        int_result = int(i)
        if (int_result == 0):
            decision = 'No'
        elif (int_result==1):
            decision = 'Yes'
        else:
            decision = 'Not defined'
    #return the output and Load result.html
    return render_template('result.html', status=decision)

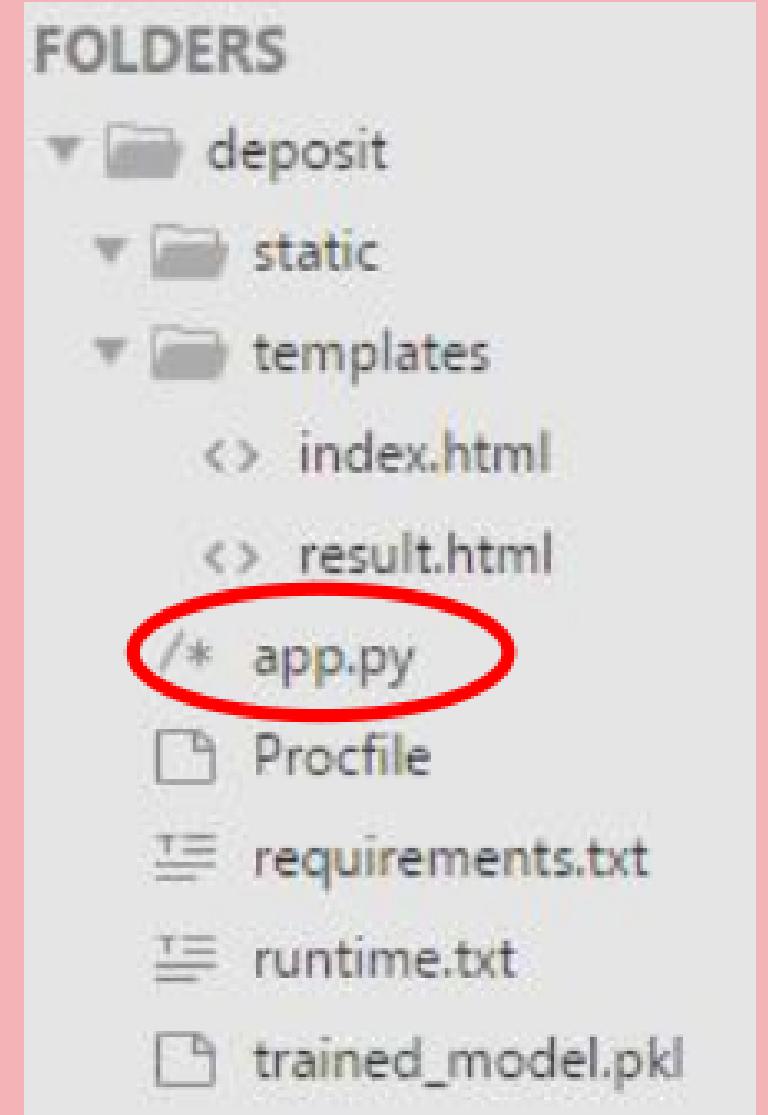
if __name__ == "__main__":
    #host= ip address, port = port number
    #app.run(host='127.0.0.1', port='5001')
    app.run()
```

index.html

app.py

trained_model.pkl

result.html

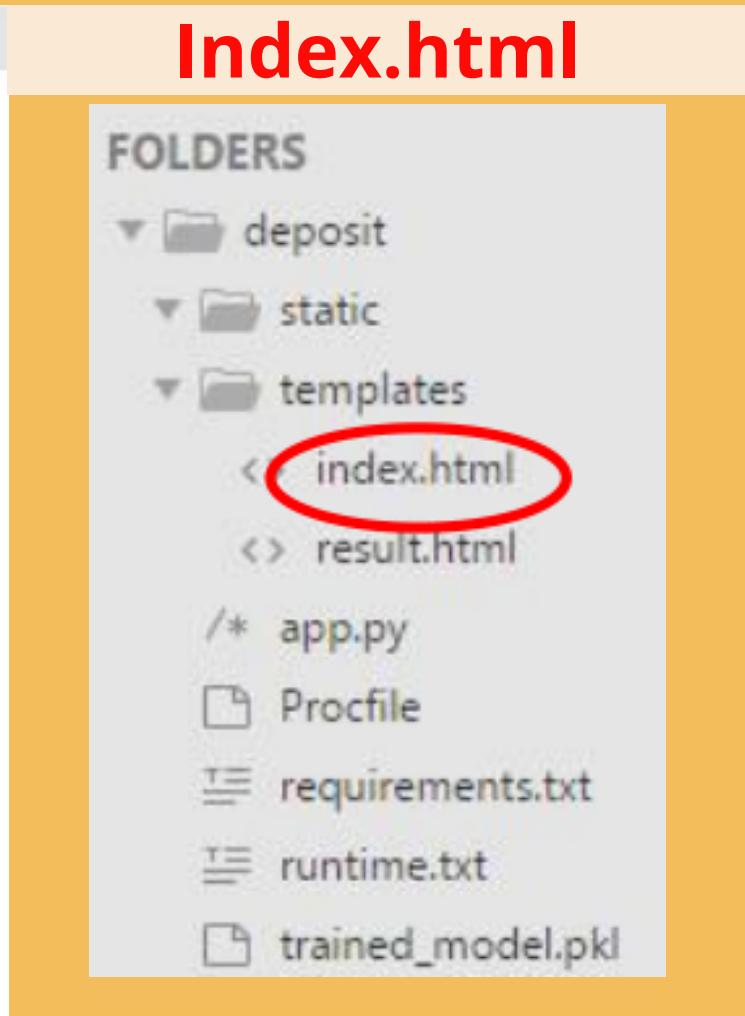


Integrating ML Model into Web App

HTML Interface (Front-End)

```
index.html
<head>
  <title>Bank Deposit Prediction</title>
</head>

<body>
  <center>
    <h2> Bank Long-term Deposit Prediction </h2><p><p>
    <form method = "post" action = "/result/">
      <table border=1 width=50%>
        <tr>
          <td>Age</td>
          <td>
            <input type = "text" name="age" value="69">
          </td>
        </tr>
        <tr>
          <td>Job</td>
          <td>
            <select name="job">
              <option value="admin.">admin.</option>
              <option value="technician">technician</option>
              <option value="services">services</option>
              <option value="management">management</option>
              <option value="retired">retired</option>
              <option value="blue-collar">blue-collar</option>
              <option value="unemployed">unemployed</option>
              <option value="entrepreneur">entrepreneur</option>
              <option value="housemaid">housemaid</option>
              <option value="self-employed">self-employed</option>
              <option value="student">student</option>
              <option value="unknown">unknown</option>
            </select>
          </td>
        </tr>
        <tr>
          <td>Marital</td>
          <td>
            <select name="marital">
              <option value="married">married</option>
              <option value="single">single</option>
              <option value="divorced">divorced</option>
            </select>
          </td>
        </tr>
      </table>
    </form>
  </center>
</body>
```

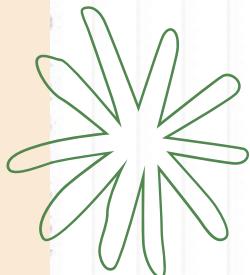


Hasil Eksekusi

Bank Long-term Deposit Prediction

Age	69
Job	admin.
Marital	married
Education	primary
Default	no
Balance	2000
Housing	no
Loan	no
Contact	telephone
Day	5
Month	jan
Duration	1000
Campaign	1
Pdays	-1
Previous	0
Poutcome	failure

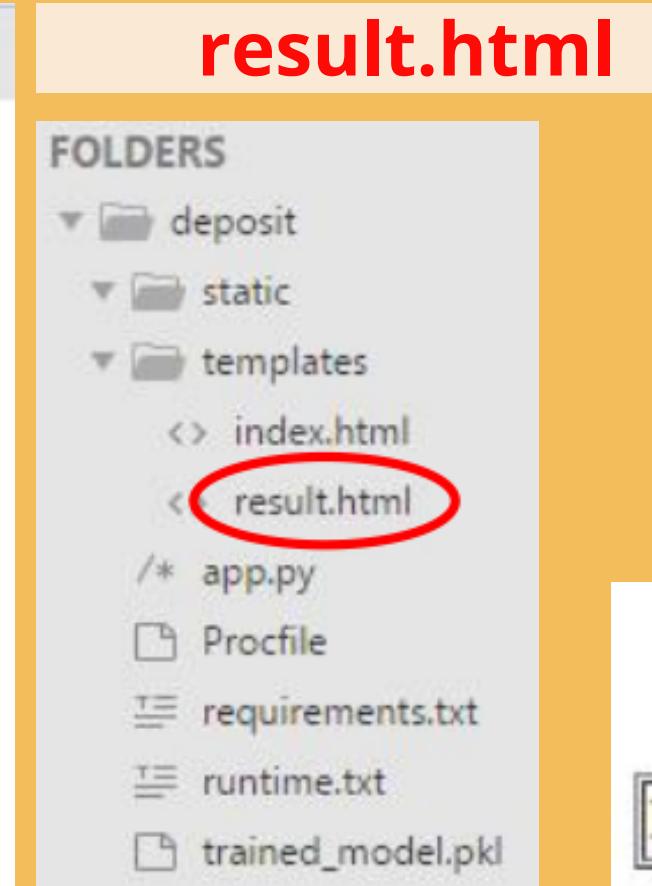
Buttons: Reset, Submit



Integrating ML Model into Web App

HTML Interface (Front-End)

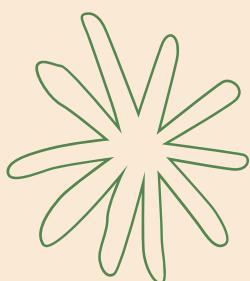
```
result.html x
1 <!DOCTYPE html>
2 <html lang="en">
3 <html>
4 <head>
5   <title>Bank Long-term Deposit Prediction</title>
6 </head>
7 <body>
8   <center>
9     <h2> Prediction Result </h2><p><p>
10    <table border=1 width=20%>
11
12      <tr>
13        <td>Possibility to subscribe the long-term deposit is </td>
14        <td>{{status}}</td>
15      </tr>
16
17    </table>
18
19  <p><p>
20
21 </body>
22 </html>
```



Hasil Eksekusi

Prediction Result

Possibility to subscribe the long-term deposit is Yes



Integrating ML Model into Web App

Result : Running on Local Computer

C ⓘ 127.0.0.1:5000

성균관대학교 성균관대학교 아이... 동국대 메일

Bank Long-term Deposit Prediction

Age	17
Job	unknown
Marital	single
Education	primary
Default	no
Balance	2000
Housing	no
Loan	no
Contact	telephone
Day	5
Month	jan
Duration	10
Campaign	1
Pdays	-1
Previous	0
Poutcome	failure
	Reset Submit



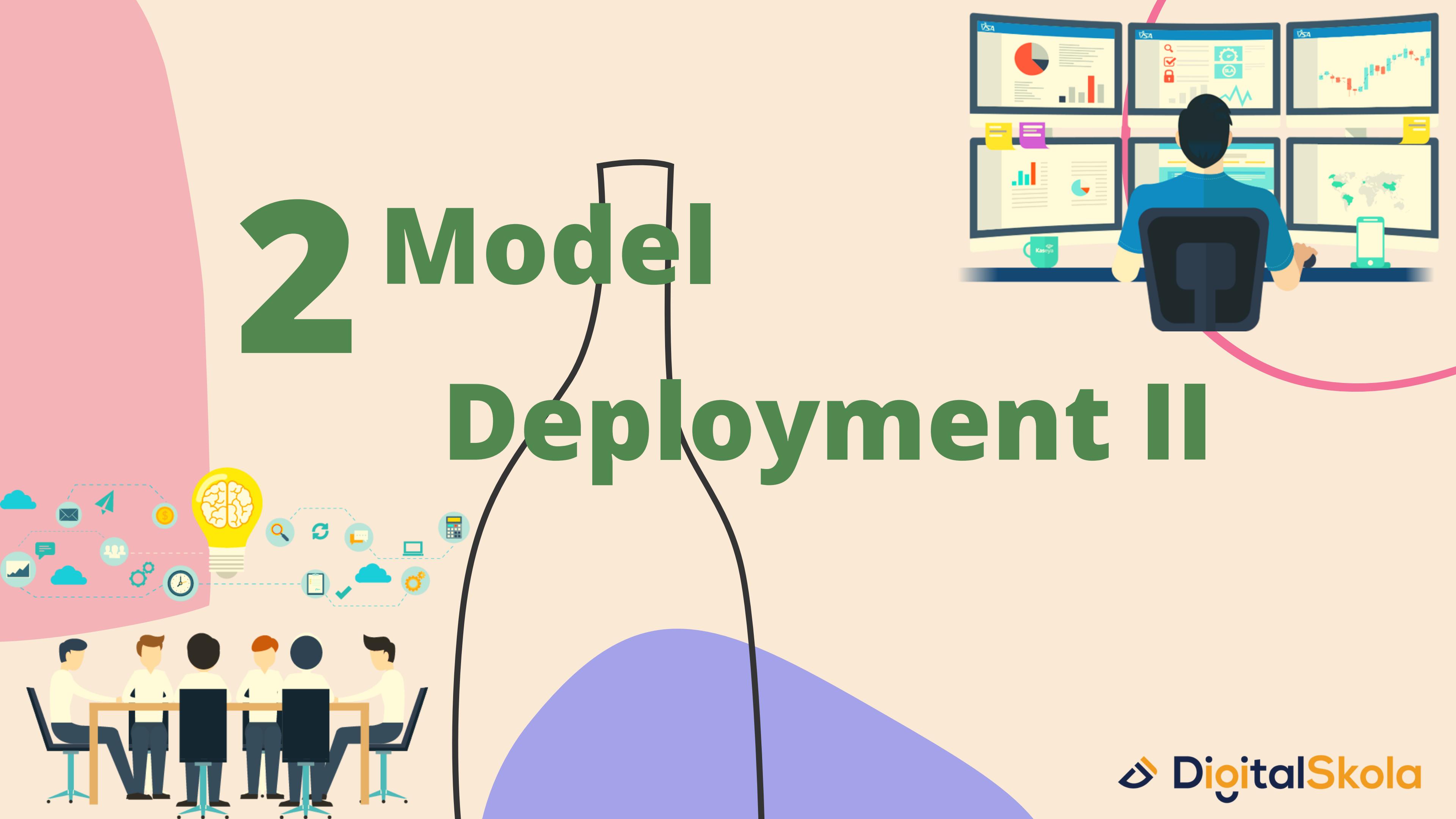
C ⓘ 127.0.0.1:5000/result/

성균관대학교 성균관대학교 아이... 동국대

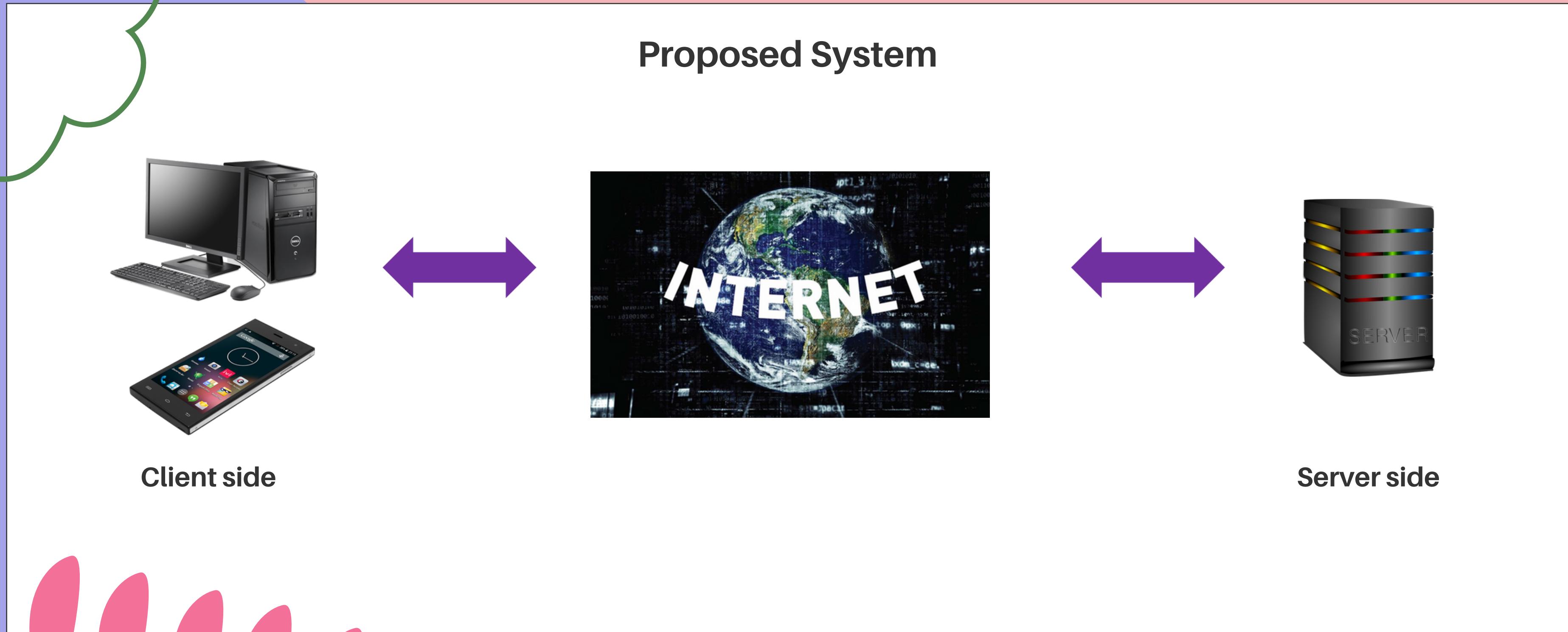
Prediction Result

Possibility to subscribe the long-term deposit is

2 Model Deployment II



Integrating Model ML to Web App



Save Trained Model Into Model

```
X_train, X_test, y_train, y_test = train_test_split(  
    df_X, df_y, test_size=0.2, random_state=42)  
  
numerical_transformer = SimpleImputer(strategy='median')  
categorical_transformer = Pipeline(steps=[('imputer', SimpleImputer(strategy='most_frequent')),  
                                         ('imput', OrdinalEncoder())])  
  
preprocessor = ColumnTransformer(  
    transformers=[  
        ('num', numerical_transformer, nums),  
        ('cat', categorical_transformer, cats)  
    ])  
  
pipeline = Pipeline(steps=[('preprocessor', preprocessor), ('scaling', StandardScaler()),  
                        #('feature_selection', SelectFromModel(ExtraTreesClassifier(random_state=42)),  
                        #('balance', SMOTE()),  
                        ('classifier', RandomForestClassifier(random_state=42, max_depth=10))])  
  
pipeline = pipeline.fit(X_train, y_train)  
filename = 'trained_model.pkl'  
joblib.dump(pipeline, filename)
```

이름	수정한 날짜	유형	크기
bank.csv	21/09/2019 19:16	Microsoft Excel C...	898KB
create_model.py	24/08/2021 15:07	PY 파일	2KB
hello_world.py	25/08/2021 13:30	PY 파일	1KB
index.html	25/08/2021 13:52	HTML 문서	1KB
trained_model.pkl	25/08/2021 13:56	PKL 파일	48,744KB

Web App (Backend)

```

import pandas as pd
import joblib
from flask import Flask, redirect, url_for, request, render_template

app = Flask(__name__)
#load index.html/ first page, receive input variable from user
@app.route("/")
def index():
    return render_template("index.html")

#load result.html, the result of prediction is presented here.
@app.route("/result/", methods=["POST"])
def prediction_result():
    #receiving parameters sent by client
    age = int(request.form.get('age'))
    job = request.form.get('job')
    marital = request.form.get('marital')
    education = request.form.get('education')
    default = request.form.get('default')
    balance = int(request.form.get('balance'))
    housing = request.form.get('housing')
    loan = request.form.get('loan')
    contact = request.form.get('contact')
    day = int(request.form.get('day'))
    month = request.form.get('month')
    duration = int(request.form.get('duration'))
    campaign = int(request.form.get('campaign'))
    pdays = int(request.form.get('pdays'))
    previous = int(request.form.get('previous'))
    poutcome = request.form.get('poutcome')
    #load the trained model.
    filename = 'trained_model.pkl'
    loaded_model= joblib.load(filename)
    #create new dataframe
    data = {'age':age, 'job':job, 'marital':marital, 'education':education, 'default':default, 'balance':balance, 'housing':housing,
            'loan':loan, 'contact':contact, 'day':day, 'month':month, 'duration':duration, 'campaign':campaign, 'pdays':pdays,
            'previous':previous, 'poutcome':poutcome}
    pd.set_option('display.max_columns', None)
    pd.set_option('display.max_rows', None)
    df_input = pd.DataFrame(data, index=[0])
    #print(df_input.dtypes)
    result = loaded_model.predict(df_input)
    #print(result)
    for i in result:
        int_result = int(i)
        if (int_result == 0):
            decision = 'No'
        elif (int_result==1):
            decision = 'Yes'
        else:
            decision = 'Not defined'
    #return the output and load result.html
    return render_template('result.html', status=decision)

if __name__ == "__main__":
    #host= ip address, port = port number
    app.run(host='127.0.0.1', port='5001')
    app.run()

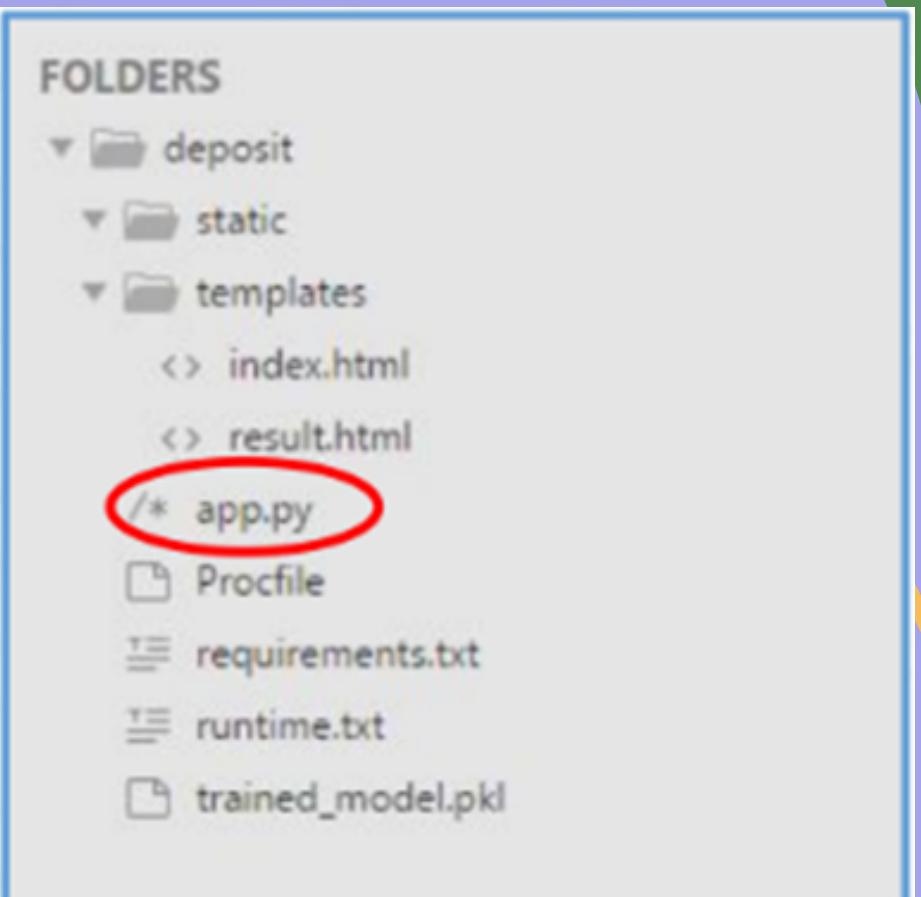
```

app.py

index.html

trained_model.pkl

result.html



HTML Interface (Frontend)

```

index.html
x
<head>
  <title>Bank Deposit Prediction</title>
</head>

<body>
  <center>
    <h2> Bank Long-term Deposit Prediction </h2><p><p>
    <form method = "post" action = "/result/">
      <table border=1 width=50%>
        <tr>
          <td>Age</td>
          <td>
            <input type = "text" name="age" value="69">
          </td>
        </tr>
        <tr>
          <td>Job</td>
          <td>
            <select name="job">
              <option value="admin.">admin.</option>
              <option value="technician">technician</option>
              <option value="services">services</option>
              <option value="management">management</option>
              <option value="retired">retired</option>
              <option value="blue-collar">blue-collar</option>
              <option value="unemployed">unemployed</option>
              <option value="entrepreneur">entrepreneur</option>
              <option value="housemaid">housemaid</option>
              <option value="self-employed">self-employed</option>
              <option value="student">student</option>
              <option value="unknown">unknown</option>
            </select>
          </td>
        </tr>
        <tr>
          <td>Marital</td>
          <td>
            <select name="marital">
              <option value="married">married</option>
              <option value="single">single</option>
              <option value="divorced">divorced</option>
            </select>
          </td>
        </tr>
      </table>
      <input type="submit" value="Predict" />
    </form>
  </center>
</body>

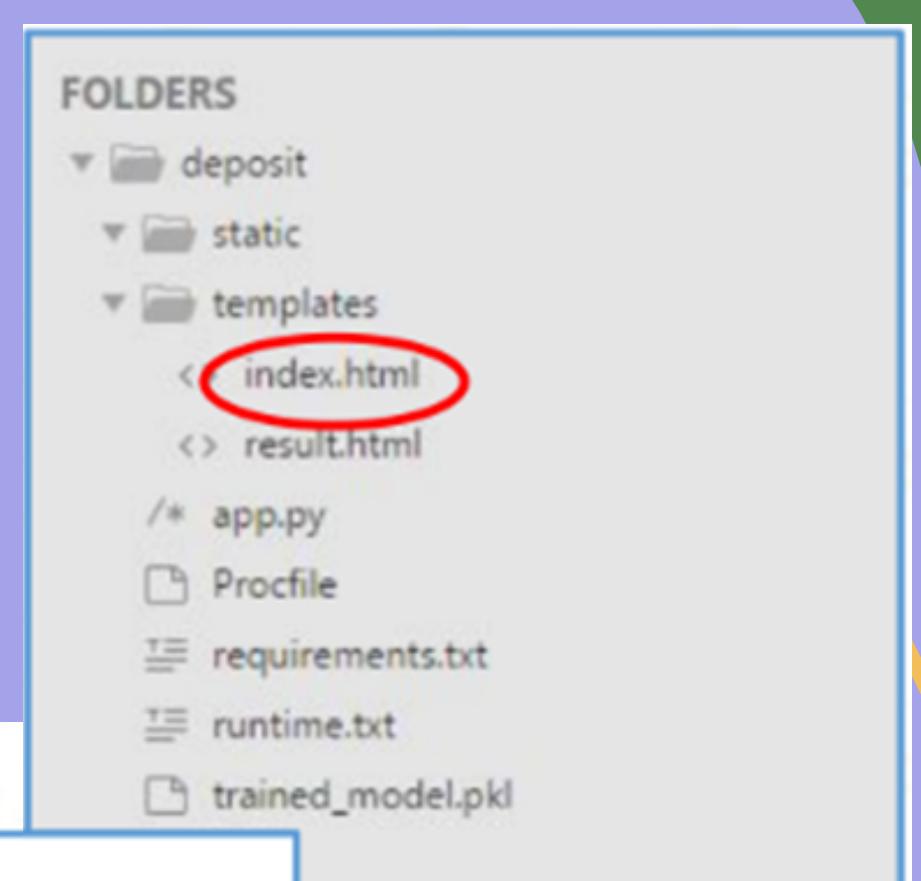
```

index.html

hasil eksekusi

Bank Long-term Deposit Prediction

Age	69
Job	admin.
Marital	married
Education	primary
Default	no
Balance	2000
Housing	no
Loan	no
Contact	telephone
Day	5
Month	jan
Duration	1000
Campaign	1
Pdays	-1
Previous	0
Poutcome	failure
	<input type="button" value="Reset"/> <input type="button" value="Submit"/>



HTML Interface (Frontend)

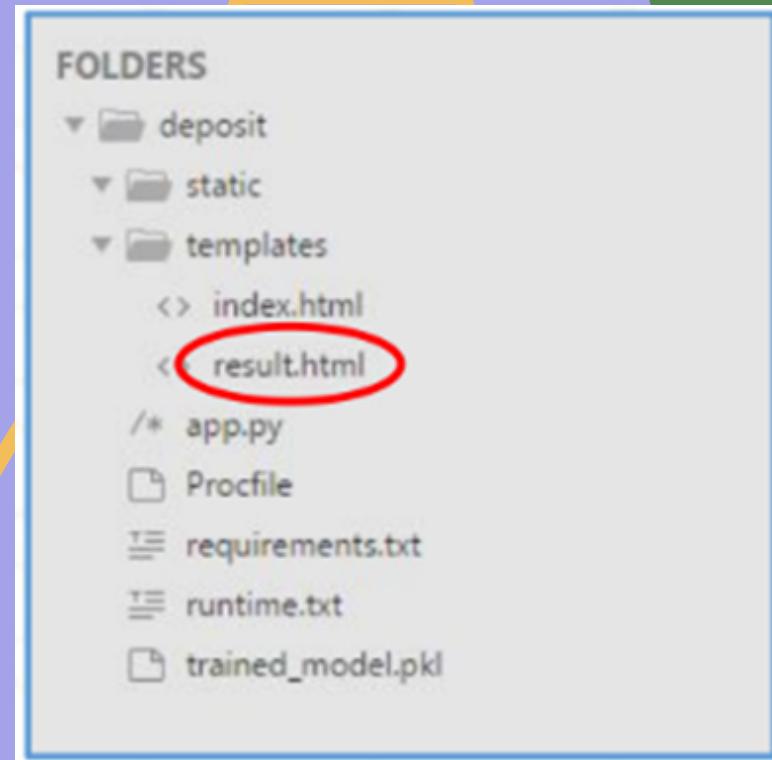
```
result.html      x
1 <!DOCTYPE html>
2 <html lang="en">
3 <html>
4 <head>
5   <title>Bank Long-term Deposit Prediction</title>
6 </head>
7 <body>
8   <center>
9     <h2> Prediction Result </h2><p><p>
10    <table border=1 width=20%>
11
12    <tr>
13      <td>Possibility to subscribe the long-term deposit is </td>
14      <td>{{status}}</td>
15    </tr>
16
17  </table>
18
19  <p><p>
20
21 </body>
22 </html>
```

result.html

hasil eksekusi

Prediction Result

Possibility to subscribe the long-term deposit is Yes



RESULT : Running on Local Computer

C ⓘ 127.0.0.1:5000

성균관대학교 성균관대학교 아이... 동국대 메일 »

Bank Long-term Deposit Prediction

Age	17
Job	unknown
Marital	single
Education	primary
Default	no
Balance	2000
Housing	no
Loan	no
Contact	telephone
Day	5
Month	jan
Duration	10
Campaign	1
Pdays	-1
Previous	0
Poutcome	failure
	<input type="button" value="Reset"/> <input type="button" value="Submit"/>

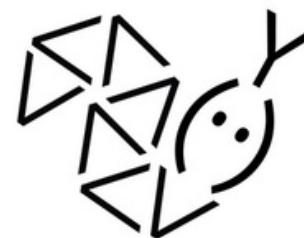
C ⓘ 127.0.0.1:5000/result/

성균관대학교 성균관대학교 아이... 동국대 메일 »

Prediction Result

Possibility to subscribe the long-term deposit is

Deploy ML Model on PyhtonAnywhere

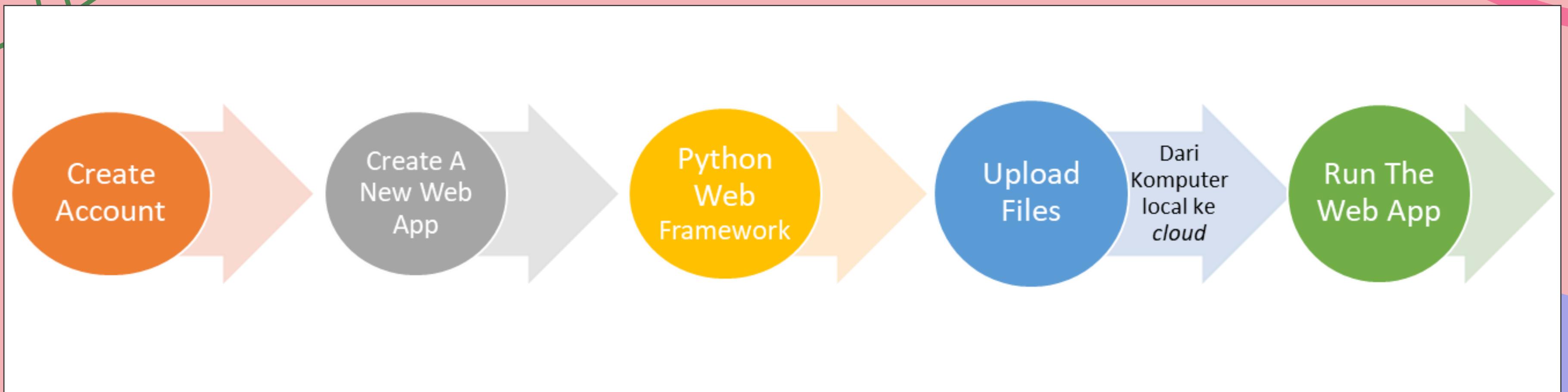


pythonanywhere

Sebuah *online integrated development environment* dan layanan *web hosting* berbasis bahasa pemrograman Python.

Menyediakan Layanan akses menggunakan browser ke server berbasis Python dan juga *command line interface*.

Steps for Deploy ML Model on PyhtonAnywhere



Deploy ML Model on Heroku



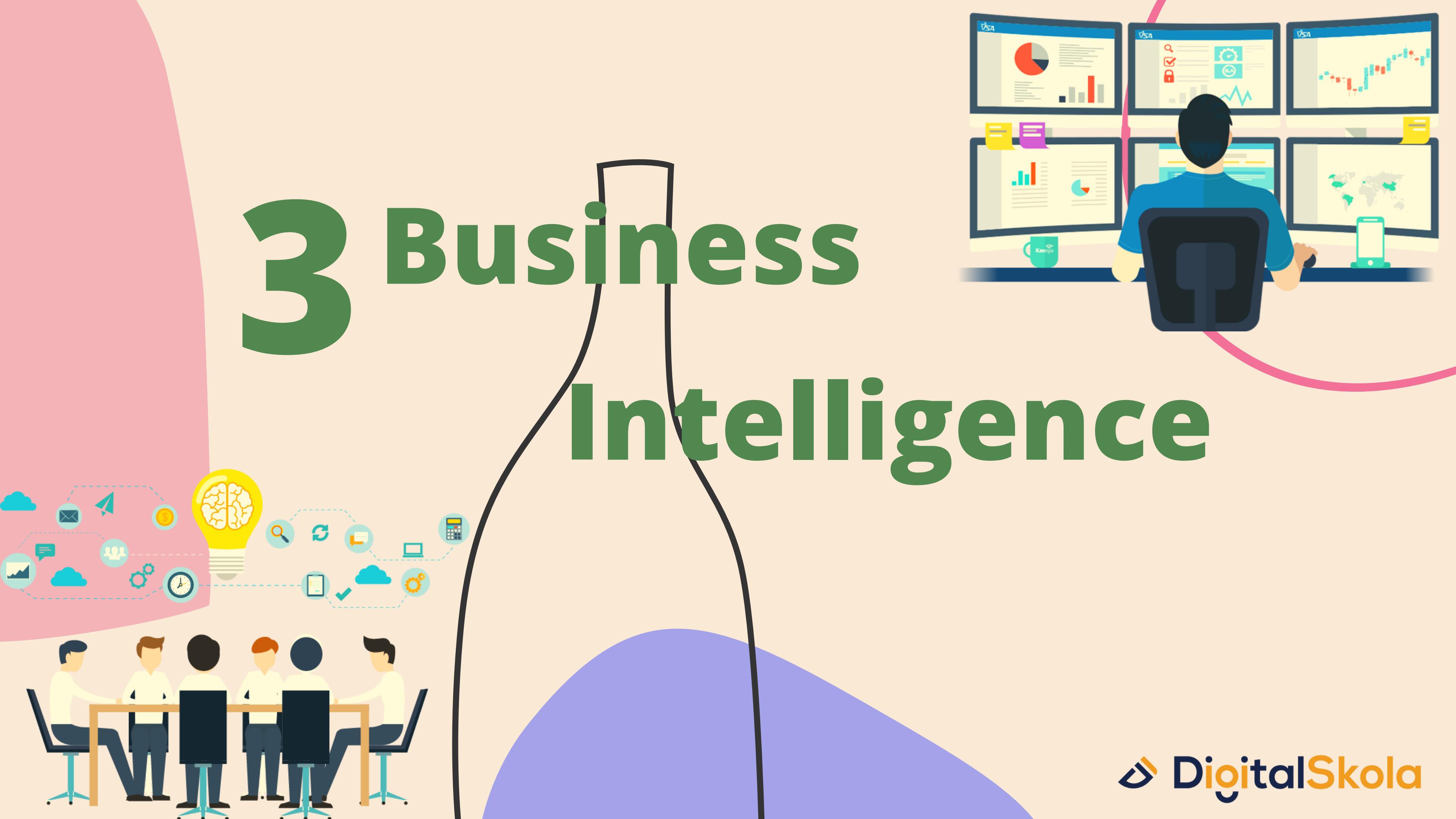
Sebuah *platform as a service* (PaaS), yang memungkinkan *programmer* untuk membangun, menjalankan, dan mengoperasikan aplikasi sepenuhnya di *cloud*.

Sebelum *deploy* model ML di Heroku, pastikan terlebih dahulu aplikasi telah di uji dan sukses di pada mesin lokal terlebih dahulu.

Steps for Deploy ML Model on Heroku



3 Business Intelligence



Business Intelligence

merupakan seperangkat proses, arsitektur dan teknologi yang mengubah data mentah menjadi informasi bermakna yang mengarahkan pada tindakan bisnis yang menguntungkan.

menggabungkan analisis bisnis, *data-mining*, visualisasi data, peralatan dan infrastruktur data, dan *best-practice* yang membantu sebuah organisasi membuat lebih banyak keputusan berdasarkan data.



kenapa BI penting?

- 1 mengidentifikasi cara meningkatkan profit
- 2 menganalisa perilaku customer
- 3 membandingkan data dengan kompetitor
- 4 *Track Performance*
- 5 mengoptimalkan operations
- 6 memprediksi keberhasilan
- 7 *Spot Market Trends*
- 8 menemukan issue atau masalah

Pemain Kunci dalam BI

DATA SCIENTIST



Professional Data Analyst

selami data dan dapatkan wawasan

Tim IT

me-maintain infrastruktur

CEO



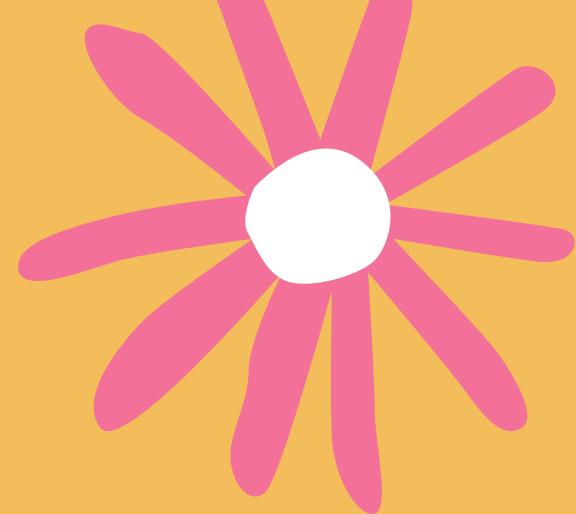
Head of the Company

sebagai pengambil keputusan

Business User

melakukan evaluasi / monitoring





BI & Data Science

Business Intelligence		Data Science	
	menganalisa dengan Retrospeksi		memprediksi suatu kondisi
	menampilkan situasi saat ini atau masa lalu		kalkulasi statistik
	Kasus terlambat ke kantor		Kasus terlambat ke kantor
	mengapa kita terlambat ke kantor?		menampilkan opsi agar tidak terlambat nanti
	jalan mana yg kita pilih?		memprediksi jalan yg tidak macet
	transportasi apa yg kita gunakan?		memilih transportasi terbaik



Dashboard & Report

Dashboard

informasi yang disajikan menyeluruh dan dapat di lihat dari waktu ke waktu

- penjualan, revenue dan profit
- performa seseorang menggunakan voucher
- pengeluaran tim operasional

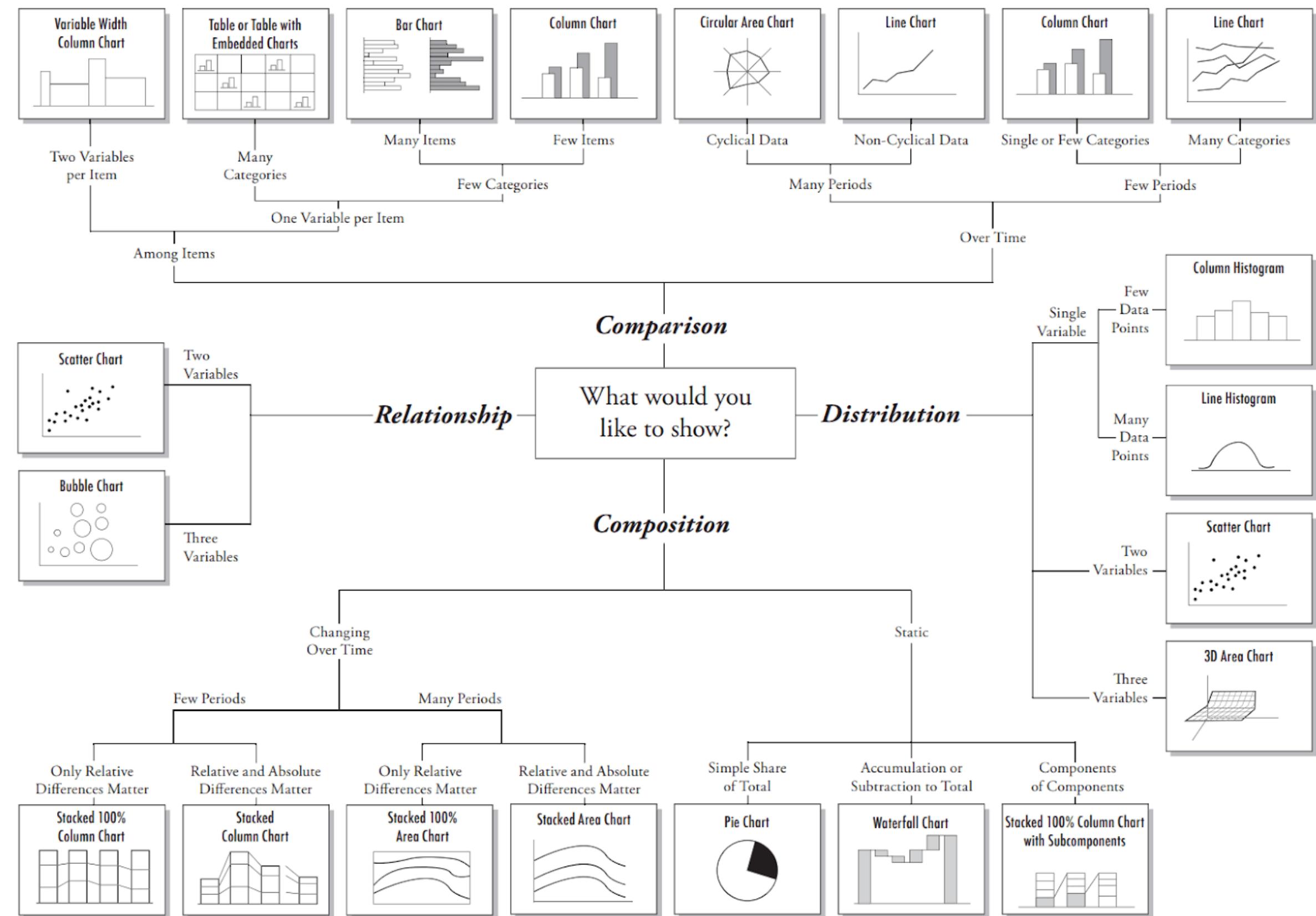
Report

spesifik informasi yang dibutuhkan dan di lihat secara periode mingguan, bulanan atau tahunan

- perbedaan penjualan setelah dilakukan kampanye oleh tim marketing
- pertumbuhan kinerja bisnis model A dalam bulanan

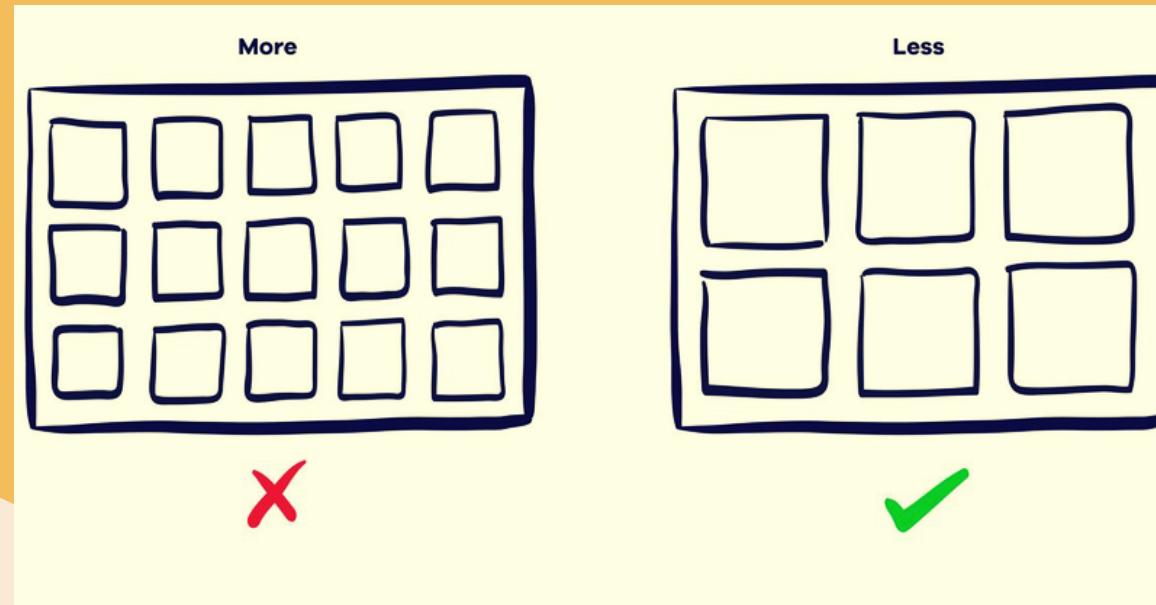
Data Visualization

Chart Suggestions—A Thought-Starter



Dashboard Constructing

- Perjelas hal yang ingin dicapai
- Hanya sertakan hal yang penting



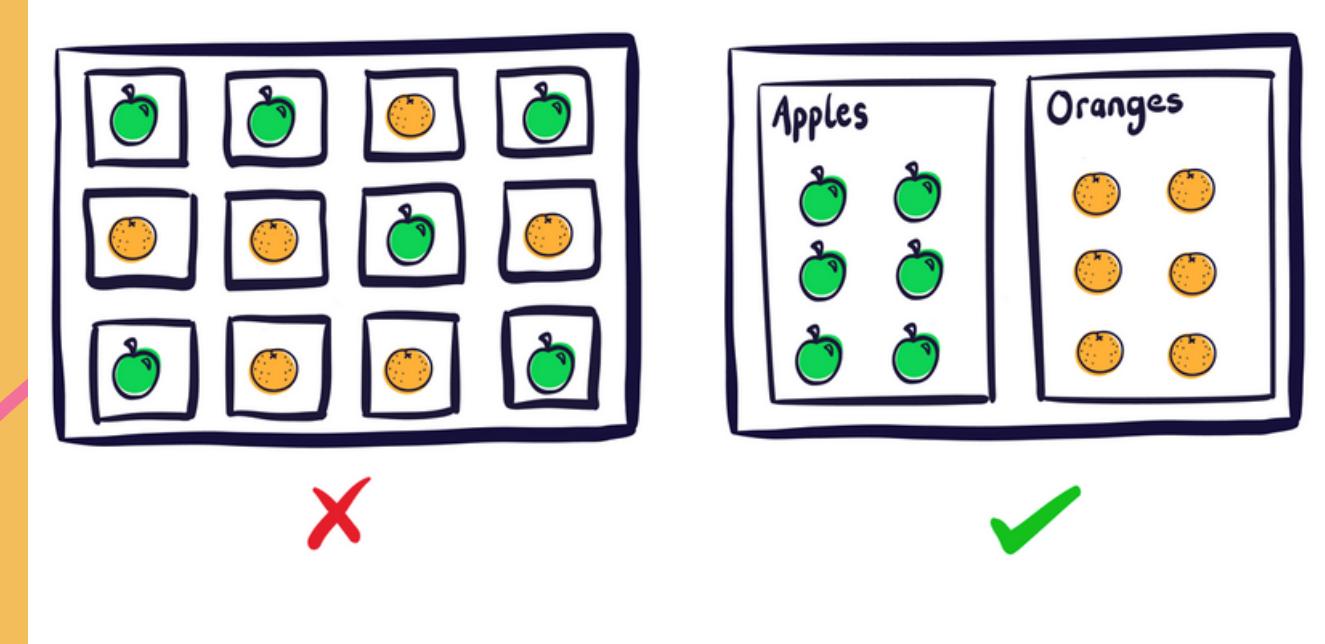
- Berikan konteks pada nomor anda



- Gunakan ukuran dan posisi untuk menunjukan hirarki



- Kelompokkan metrik yg terkait



Dashboard Constructing

- Konsisten



- Bulatkan angka



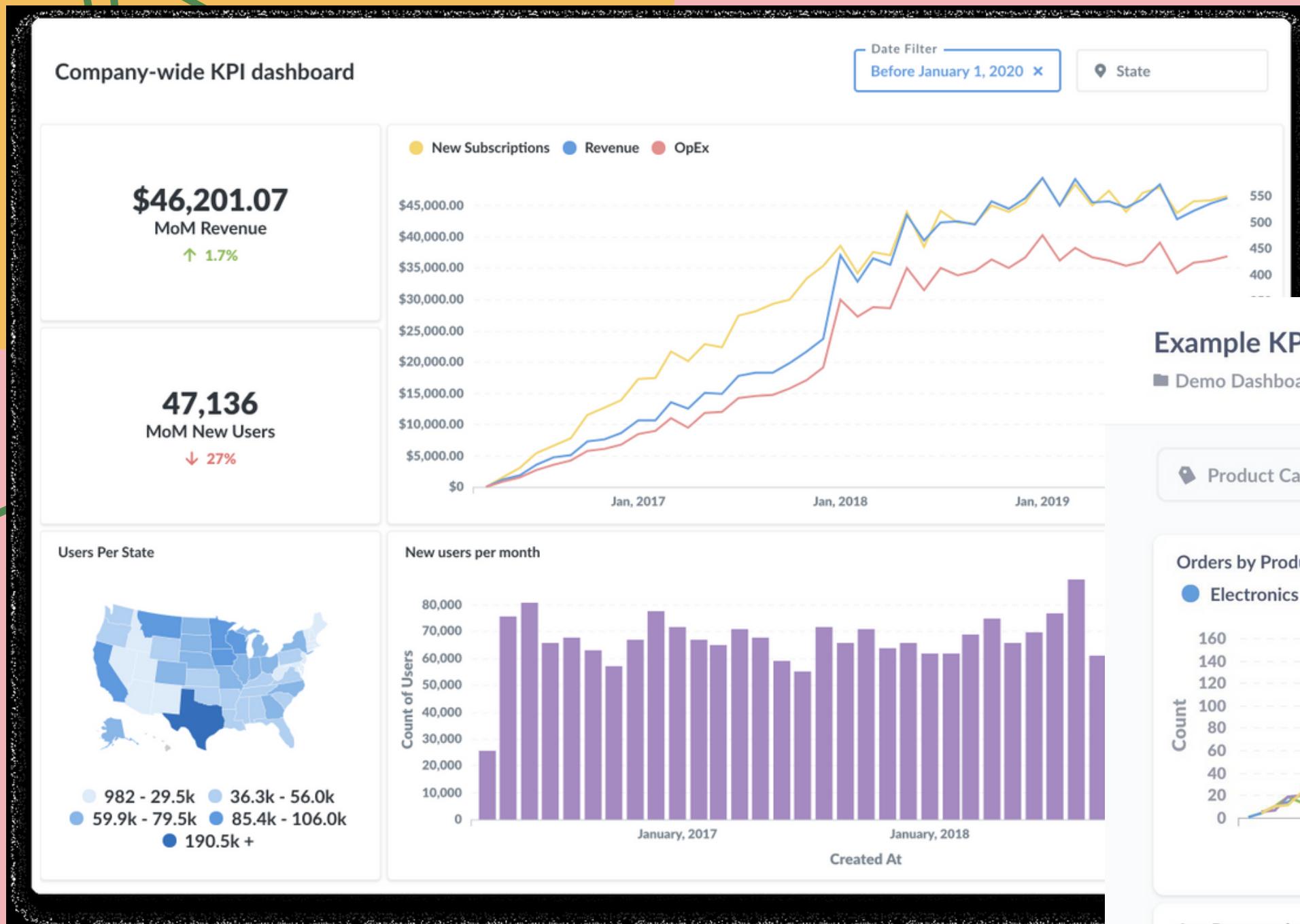
- Gunakan label yang jelas untuk audiens

- Terus kembangkan dashboard anda
- minta feedback dari stakeholder:
- - hal apa yang sering mereka lihat atau di nilai paling berguna, dan mengapa?
 - - hal apa yang tidak pernah mereka lihat atau di nilai paling tidak berguna dan mengapa?
 - - apakah itu mengubah sesuatu tentang cara mereka bekerja?

BI Tools

adalah jenis perangkat lunak aplikasi yang mengumpulkan dan memproses sejumlah besar data tidak terstruktur dari sistem internal dan eksternal.

peralatan ini juga membantu menyiapkan data untuk analisis sehingga kita dapat membuat laporan, dashboard, dan visualisasi data.



Metabase

Example KPIs dashboard

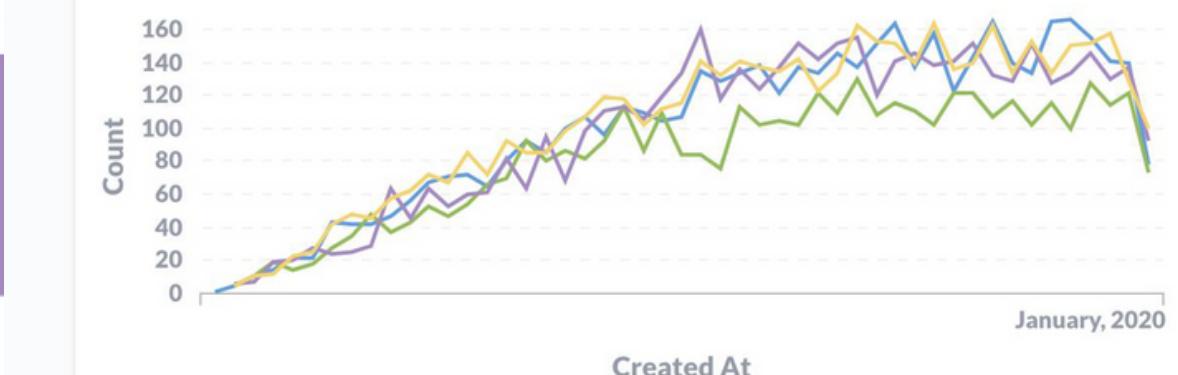
Demo Dashboards

Product Category

Product Title

Orders by Product Line

Electronics (Blue), Home and Office (Green), Kitchen (Purple), Industrial (Yellow)

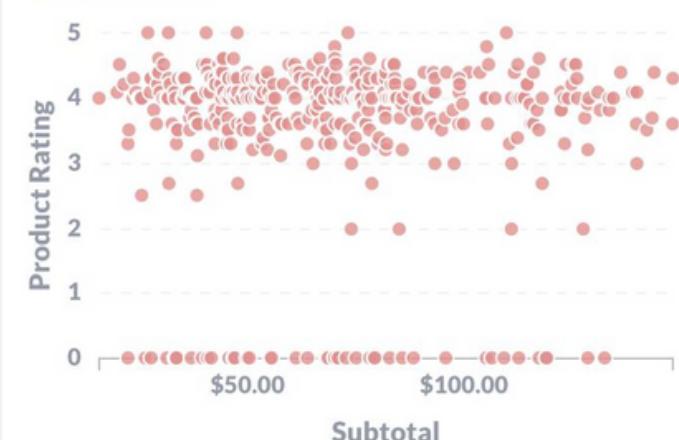


Avg Revenue by Country (new territories)

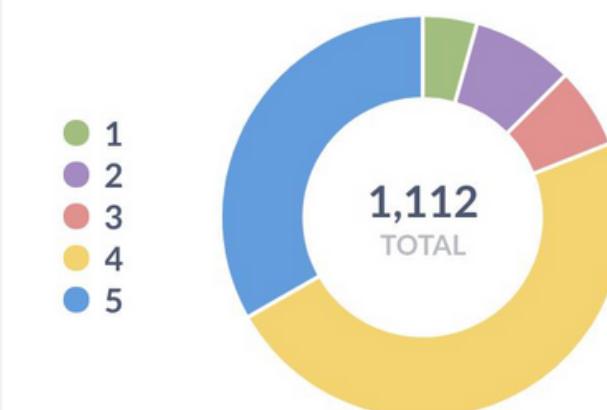
- 6.5M - 7.0M
- 7.4M - 7.6M
- 7.6M - 7.8M
- 7.8M - 8.0M
- 8.2M +



Price vs Rating

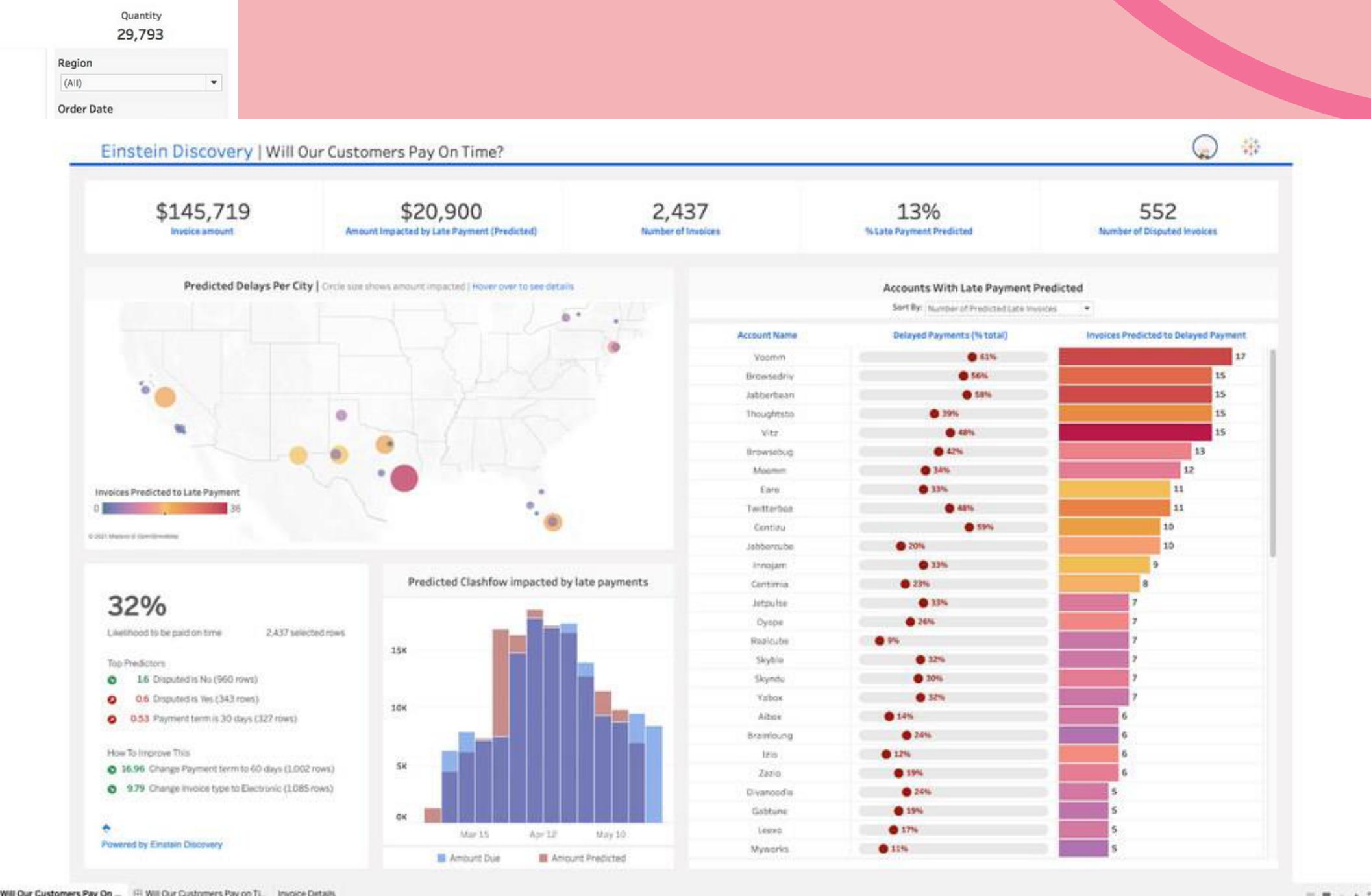
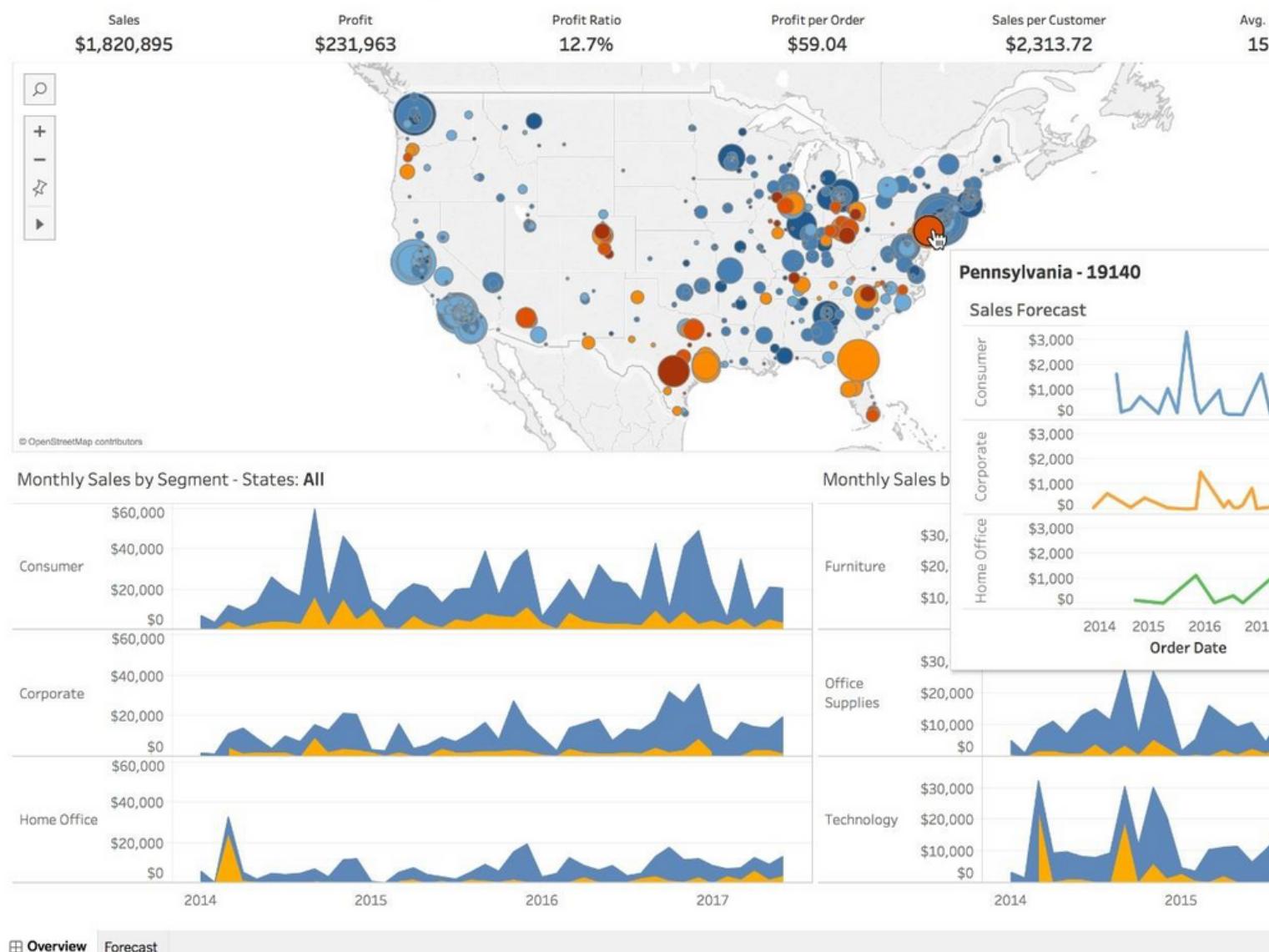


Reviews by Rating





Executive Overview - Profitability (All)





Power BI

Sales Report Option 1 - Power BI Desktop

Nitin Khanna (MSIT)

File Home Insert Modeling View Help

Cut Copy Format painter Paste Get data Excel Power BI datasets SQL Server Enter data Recent sources Transform Refresh data New visual Text box More visuals New measure Quick measure Publish

OVERVIEW Sales Report \$5.3M Australia \$5.3M Canada \$2.6M France \$2.3M Germany \$3.3M UK \$21.8M USA

Key influencers Top segments

What influences NSAT to be 7? When... the likelihood of NSAT being 7 increases by:

- UnitPrice is 298 - 299.9 → 10.20x
- UnitPrice is 196.9 - 199 → 6.58x
- Manufacturer is Litware, Inc. → 2.64x
- Color is Brown → 2.57x
- StockType is High → 1.96x
- Manufacturer is Contoso, Ltd. → 1.34x
- Color is Silver → 1.29x

Units by Country and Sales Size

United States, Canada, Australia, Great Britain, France, Germany

Units Sold by Year, Quarter and Manufacturer

2014 Qtr 1, 2014 Qtr 2, 2014 Qtr 3, 2014 Qtr 4

Sales Amount by Brand Name

Contoso, Fabrikam, Litware, Adventure Works, Wide World Importers, A.D.蝉, Th., Proseware, Southridge Video, Acme, Northern

Sales Amount by Year, Month and Brand Name

2013 February, Contoso, Proseware, Adventure Works, Other, Wide World Importers, 2013 March

Visualizations

Sales & Returns Analysis - Power BI Desktop

Miguel Martinez

File Home Insert Modeling View Help

Cut Copy Format painter Paste Get data Excel Power BI datasets SQL Server Enter data Recent sources Transform Refresh data New visual Text box More visuals New measure Quick measure Publish

Microsoft | Skateboard Store

Last Refresh: Jun 30th, 2019 / Chicago, IL, USA

What If... We Decrease Our Return Rate (%) to: 25

Net Sales (Forecast) \$30,772 Extra Profit \$0 0.0% Profit Increase

Return Rate 24%

Net Sales (Forecast) by Location

Park Ridge, Elmwood Park, Oak Park, Cicero, Burbank, Lakewood, ORD, MDW

"What If" Analysis Forecast Predicted Net Sales Forecast

Date Jan 2019, Mar 2019, May 2019, Jul 2019

Fields

Search

Analysis DAX Design DAX % Return Rate Age Associated Product Association Calendar Customer Details Issues and Promotions Product Sales STable Store Tooltip Info Product is OneNote

DigitalSkola

©Microsoft Corporation. All rights reserved.

PAGE 5 OF 18

Google Data Studio

Google Data Studio is a powerful tool for creating custom reports from multiple data sources. It allows you to visualize your data in various ways, such as charts, tables, and maps.

The interface includes:

- Header:** Data Last Updated: 10/20/2017 10:35:48 AM | Hide Sampling | Privacy Policy | Google Data Studio
- Left Sidebar:** Data Studio logo, Create button, Recent section (Shared with me, Owned by me, Trash), Reports tab selected.
- Recent Reports:** Blank Report (Data Studio), Tutorial Report (Data Studio), Acme Marketing (Google Analytics), Search Console Report (Search Console), Google Ads Overview (Google Ads).
- Main Content Area:** A dashboard featuring:
 - A table showing user statistics by language and city.
 - A pie chart showing the distribution of users by language.
 - A world map showing user distribution.
 - Various summary metrics like Users, New Users, Sessions, etc.
- Right Side:** Google Analytics Audience Overview report showing audience at a glance, language breakdown, country breakdown, and device usage.
- Bottom Navigation:** Name, Owned by anyone, Last opened by me.
- Footer:** DigitalSkola logo.



DigitalSkola

THANK YOU!

By Omicron