
A/B Testing dengan Algoritma Machine Learning

Risya Diany Yulinska (11180910000118)

Ahmad Malik Fajar (11180910000120)

Nabila Lailika Amalia (11180910000128)

**UJIAN AKHIR SEMESTER
SOFTWARE TESTING QUALITY
2021**

1. Latar Belakang
2. Identifikasi Masalah
3. Rumusan Masalah
4. Manfaat
5. Batasan Masalah
6. Metode
7. Tools
8. Eksperimen
9. Kesimpulan
10. Daftar Pustaka

OUTLINE

LATAR BELAKANG

- Dengan munculnya pemasaran digital yang dipimpin oleh alat termasuk Google Analytics, Google Adwords, dan Iklan Facebook, keunggulan kompetitif utama bagi bisnis adalah menggunakan A/B Testing untuk menentukan efek dari upaya pemasaran digital.
- *A/B Testing* memungkinkan untuk menentukan apakah perubahan pada halaman arahan, formulir popup, judul artikel, dan keputusan pemasaran digital lainnya meningkatkan tingkat konversi dan mendapatkan customer purchasing behavior.
- Masalah utama dengan pendekatan inferensi statistik tradisional untuk A/B Testing adalah bahwa A/B Testing hanya membandingkan 2 variabel - experiment/control untuk mendapatkan hasil/outcome.
- Masalahnya adalah bahwa perilaku pelanggan jauh lebih kompleks dari ini. Pelanggan mengambil jalur yang berbeda, menghabiskan jumlah waktu yang berbeda di situs, berasal dari latar belakang yang berbeda (usia, jenis kelamin, minat), dan banyak lagi

IDENTIFIKASI MASALAH

Tujuan *A/B Testing* adalah untuk mendapatkan outcome dari hasil perbandingan 2 variabel yaitu *treatment* dan *control* menggunakan inferensi statistik. Namun masalahnya, dunia bukanlah ruang hampa yang hanya melibatkan eksperimen (*treatment and control*) dan efek. Situasinya jauh lebih kompleks dan dinamis. Pertimbangkan situasi ini:

- Pengguna memiliki karakteristik yang berbeda
- Pengguna menghabiskan jumlah waktu yang berbeda di situs web
- Pengguna menemukan situs web secara berbeda
- Pengguna mengambil jalur yang berbeda

Seringkali pemodelan *A/B Testing* dalam ruang hampa ini dapat menyebabkan kesalahpahaman tentang kisah nyatanya / *True story*.

RUMUSAN MASALAH

1. Apa itu A/B Testing?
2. Algoritma Machine Learning apa yang digunakan dalam eksperimen A/B Testing?
3. Bagaimana hasil dari A/B Testing yang menggunakan implementasi Algoritma Machine Learning?

MANFAAT

- Meningkatkan Traffic & Conversion Rate pada website dengan menggunakan A/B Testing
- Memahami bagaimana implementasi Algoritma Machine Learning pada A/B Testing
- Memenuhi Tugas Ujian Akhir Semester Mata Kuliah Software Testing Quality

BATASAN MASALAH - Metode

1. A/B Testing
2. Algoritma Machine Learning
 - Linear Regression
 - Decision Tree
 - XGBoost

BATASAN MASALAH - Tools

Eksperimen dengan menggunakan :

- Jupyter Notebook
- Bahasa Pemrograman Python
- Python IDE
- Windows System
- Library pandas, numpy, Scikit-Learn, matplotlib, statsmodels

BATASAN MASALAH - Proses

1. Menggunakan Dataset Control Data & Experiment Data yang didapat dari Kaggle
2. A/B Testing dengan menggunakan Algoritma Machine Learning seperti Linear Regression, Decision Tree dan XGBoost

METODE - A/B Testing

A/B Testing adalah tried-and-true method yang biasanya dilakukan dengan menggunakan pendekatan inferensi statistik tradisional yang didasarkan pada uji hipotesis (misalnya t-test, z-score, chi-squared test). Ada 2 tes dijalankan secara paralel:

- ***Treatment Group (Grup A)*** - Grup ini diekspos ke halaman web baru, formulir sembulan, dll.
- ***Control Group (Grup B)*** - Grup ini tidak mengalami perubahan dari pengaturan saat ini.

METODE - Algoritma Machine Learning

- **Linear Regression**

Salah satu algoritma dasar yang digunakan untuk memodelkan hubungan antara variabel dependen dengan satu atau lebih variabel independen dengan meminimalkan jarak kuadrat residu.

- **Decision Tree**

Metode klasifikasi yang menggunakan struktur pohon, dimana setiap node merepresentasikan atribut dan cabangnya merepresentasikan nilai dari atribut, sedangkan daunnya digunakan untuk merepresentasikan kelas.

- **XGBoost atau eXtreme Gradient Boosting**

Algoritma berbasis pohon yang merupakan metode ensemble dengan tujuan utama mengurangi bias dan varians.

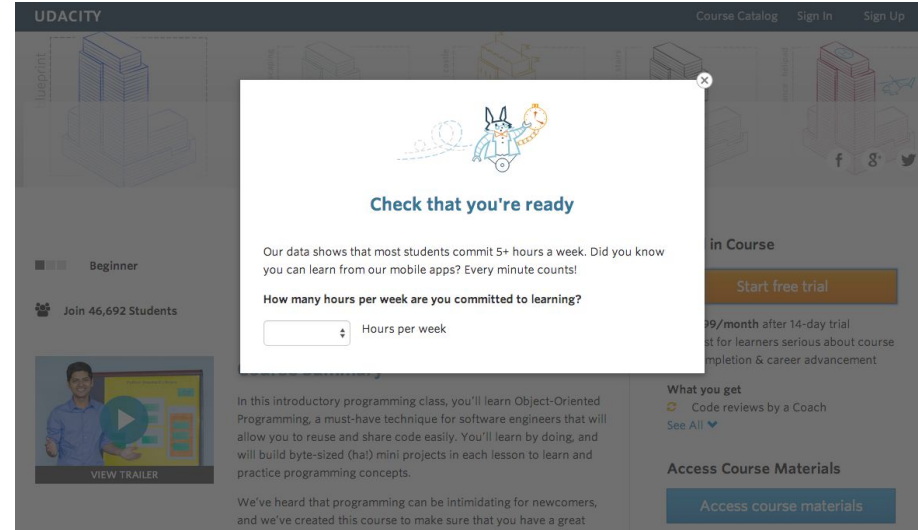
TOOLS

Kami menggunakan Jupyter Notebook yang merupakan IDE open-source berbasis web yang mendukung lebih dari 40 bahasa pemrograman. Dalam eksperimen ini kami menggunakan Python sebagai bahasa pemrograman, karena Python memiliki banyak Library yang dapat digunakan sehingga memungkinkan untuk menyelesaikan permasalahan yang sangat rumit dalam hitungan menit bahkan hitungan detik. Pada eksperimen ini kami me-import beberapa library seperti pandas, numpy, Scikit-Learn, matplotlib, statsmodels.

Experiment Overview: Free Trial Screener

Dalam percobaan, menguji website Udacity saat terjadinya perubahan di mana jika siswa mengklik "*start free trial*", mereka ditanya berapa banyak waktu yang mereka miliki untuk dicurahkan ke *course*.

Jika siswa menunjukkan 5 jam atau lebih per minggu, mereka akan dibawa melalui proses checkout seperti biasa. Jika mereka menunjukkan kurang dari 5 jam per minggu, sebuah pesan akan muncul yang menunjukkan bahwa *course* Udacity biasanya memerlukan komitmen waktu yang lebih besar untuk dapat berhasil menyelesaikannya.



EKSPERIMEN - Hipotesis

Hipotesisnya adalah untuk menetapkan harapan yang lebih jelas bagi siswa diawal, sehingga mengurangi jumlah siswa yang frustrasi dan meninggalkan *free trial* karena mereka tidak memiliki cukup waktu—tanpa secara signifikan mengurangi jumlah siswa untuk melanjutkan melewati *free trial* dan akhirnya menyelesaikan kursus.

Jika hipotesis ini benar, Udacity dapat meningkatkan pengalaman siswa secara keseluruhan dan meningkatkan kapasitas pelatih untuk mendukung siswa yang kemungkinan besar akan menyelesaikan kursus.

EKSPERIMEN - Tujuan

Dalam analisis ini, kami akan menyelidiki fitur mana yang berkontribusi pada pendaftaran dan menentukan apakah ada dampak pada pendaftaran dari formulir "*Setting Expectations*" yang baru.

- Pengguna yang mengalami formulir akan dilambangkan sebagai "*Experiment = 1*"
- Grup kontrol (pengguna yang tidak melihat formulir) akan dilambangkan sebagai "*Experiment = 0*".

EKSPERIMEN - Preprocessing

```
In [1]: import pandas as pd

control_data = pd.read_csv('data/control_data.csv')
experiment_data = pd.read_csv('data/experiment_data.csv')
```

```
In [2]: control_data.head()
```

Out[2]:

	Date	Pageviews	Clicks	Enrollments	Payments
0	Sat, Oct 11	7723	687	134.0	70.0
1	Sun, Oct 12	9102	779	147.0	70.0
2	Mon, Oct 13	10511	909	167.0	95.0
3	Tue, Oct 14	9871	836	156.0	105.0
4	Wed, Oct 15	10014	837	163.0	64.0

```
In [3]: experiment_data.head()
```

Out[3]:

	Date	Pageviews	Clicks	Enrollments	Payments
0	Sat, Oct 11	7716	686	105.0	34.0
1	Sun, Oct 12	9288	785	116.0	91.0
2	Mon, Oct 13	10480	884	145.0	79.0
3	Tue, Oct 14	9867	827	138.0	92.0
4	Wed, Oct 15	9793	832	140.0	94.0


```
In [31]: # Mengecek format data
control_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 37 entries, 0 to 36
Data columns (total 5 columns):
#   Column          Non-Null Count  Dtype
---  -
0   Date            37 non-null    object
1   Pageviews       37 non-null    int64
2   Clicks          37 non-null    int64
3   Enrollments     23 non-null    float64
4   Payments        23 non-null    float64
dtypes: float64(2), int64(2), object(1)
memory usage: 1.6+ KB
```

```
In [32]: experiment_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 37 entries, 0 to 36
Data columns (total 5 columns):
#   Column          Non-Null Count  Dtype
---  -
0   Date            37 non-null    object
1   Pageviews       37 non-null    int64
2   Clicks          37 non-null    int64
3   Enrollments     23 non-null    float64
4   Payments        23 non-null    float64
dtypes: float64(2), int64(2), object(1)
memory usage: 1.6+ KB
```

```
In [33]: # Selanjutnya mendeteksi missing value menggunakan fungsi .isna()  
control_data.isna().sum()
```

```
Out[33]: Date          0  
Pageviews          0  
Clicks             0  
Enrollments       14  
Payments          14  
dtype: int64
```

```
In [34]: experiment_data.isna().sum()
```

```
Out[34]: Date          0  
Pageviews          0  
Clicks             0  
Enrollments       14  
Payments          14  
dtype: int64
```

```
In [35]: control_data[control_data['Enrollments'].isna()]
```

```
Out[35]:
```

	Date	Pageviews	Clicks	Enrollments	Payments
23	Mon, Nov 3	9437	788	NaN	NaN
24	Tue, Nov 4	9420	781	NaN	NaN
25	Wed, Nov 5	9570	805	NaN	NaN
26	Thu, Nov 6	9921	830	NaN	NaN
27	Fri, Nov 7	9424	781	NaN	NaN
28	Sat, Nov 8	9010	756	NaN	NaN
29	Sun, Nov 9	9656	825	NaN	NaN
30	Mon, Nov 10	10419	874	NaN	NaN
31	Tue, Nov 11	9880	830	NaN	NaN
32	Wed, Nov 12	10134	801	NaN	NaN
33	Thu, Nov 13	9717	814	NaN	NaN
34	Fri, Nov 14	9192	735	NaN	NaN
35	Sat, Nov 15	8630	743	NaN	NaN
36	Sun, Nov 16	8970	722	NaN	NaN

Format Data

Pada tahapan ini Kami akan melakukan hal berikut:

- * Gabungkan control_data dan experiment_data, lalu tambahkan kolom "id" yang menunjukkan apakah data tersebut merupakan bagian dari eksperimen atau bukan
- * Tambahkan kolom "row_id" untuk membantu melacak baris mana yang dipilih untuk pelatihan dan pengujian di bagian pemodelan
- * Buat fitur "Hari dalam Seminggu" dari kolom "Date"
- * Hapus kolom "Tanggal" yang tidak perlu dan kolom "Payment"
- * Tangani data yang hilang (NA) dengan menghapusnya.
- * Acak baris untuk mencampur data untuk dipelajari
- * Atur ulang kolom

```
In [36]: # Gabungkan dengan data Eksperimen
data_total = pd.concat([control_data, experiment_data])
data_total.sample(10)
```

Out[36]:

	Date	Pageviews	Clicks	Enrollments	Payments
22	Sun, Nov 2	8836	693	206.0	67.0
19	Thu, Oct 30	9345	734	167.0	75.0
9	Mon, Oct 20	10496	860	153.0	98.0
27	Fri, Nov 7	9272	767	NaN	NaN
24	Tue, Nov 4	9420	781	NaN	NaN
8	Sun, Oct 19	8459	691	131.0	60.0
3	Tue, Oct 14	9867	827	138.0	92.0
25	Wed, Nov 5	9570	805	NaN	NaN
26	Thu, Nov 6	9921	830	NaN	NaN
3	Tue, Oct 14	9871	836	156.0	105.0

```
In [37]: import numpy as np
np.random.seed(7)
import sklearn.utils

# Tambahkan row id
data_total['row_id'] = data_total.index

# Buat fitur "Hari dalam Seminggu" dari kolom "Date"
data_total['DOW'] = data_total['Date'].str.slice(start=0, stop=3)

# Menghapus Missing value
data_total.dropna(inplace=True)

# Tambahkan Eksperimen kolom biner untuk menunjukkan apakah data tersebut merupakan bagian dari eksperimen atau tidak (Acak).
data_total['Experiment'] = np.random.randint(2, size=len(data_total))

# Remove missing data
data_total.dropna(inplace=True)

# Hapus kolom Tanggal dan Pembayaran
del data_total['Date'], data_total['Payments']

# Shuffle the data
data_total = sklearn.utils.shuffle(data_total)
```

```
In [38]: # Check data yang baru  
data_total.head()
```

Out[38]:

	Pageviews	Clicks	Enrollments	row_id	DOW	Experiment
21	8460	681	156.0	21	Sat	0
3	9867	827	138.0	3	Tue	1
21	8448	695	142.0	21	Sat	0
13	9434	673	220.0	13	Fri	1
6	9008	748	146.0	6	Fri	1

Training and Testing Sets

```
In [39]: # Susun ulang kolom
data_total = data_total[['row_id', 'Experiment', 'Pageviews', 'Clicks', 'DOW', 'Enrollments']]
```

```
In [40]: # Splitting the data
from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(data_total.loc[:, data_total.columns != 'Enrollments'],\
                                                    data_total['Enrollments'], test_size=0.2)
```

```
In [41]: # Mengubah string pada kolom DOW menjadi angka
from sklearn.preprocessing import LabelEncoder

lb = LabelEncoder()
X_train['DOW'] = lb.fit_transform(X_train['DOW'])
X_test['DOW'] = lb.transform(X_test['DOW'])
```



```
In [42]: X_train.head()
```

Out[42]:

	row_id	Experiment	Pageviews	Clicks	DOW
16	16	0	9535	759	1
10	10	1	10551	864	5
2	2	0	10480	884	1
3	3	1	9867	827	5
16	16	0	9655	771	1

```
In [43]: X_test.head()
```

Out[43]:

	row_id	Experiment	Pageviews	Clicks	DOW
12	12	0	8324	665	4
18	18	1	9262	727	6
1	1	0	9288	785	3
1	1	0	9102	779	3
19	19	0	9345	734	4

Helper functions

```
In [44]: # Pertama, kami akan membuat fungsi pelaporan metrik yang disederhanakan
from sklearn.metrics import mean_squared_error, r2_score, mean_absolute_error

def calculate_metrics(y_test, y_preds):
    rmse = np.sqrt(mean_squared_error(y_test, y_preds))
    r_sq = r2_score(y_test, y_preds)
    mae = mean_absolute_error(y_test, y_preds)

    print('RMSE Score: {}'.format(rmse))
    print('R2_Squared: {}'.format(r_sq))
    print('MAE Score: {}'.format(mae))
```



```
In [45]: #Selanjutnya kita dapat membuat fungsi visualisasi yang disederhanakan, menggunakan matplotlib.pyplot  
import matplotlib.pyplot as plt  
plt.style.use('ggplot')  
%matplotlib inline  
  
def plot_preds(y_test, y_preds, model_name):  
    N = len(y_test)  
    plt.figure(figsize=(10,5))  
    original = plt.scatter(np.arange(1, N+1), y_test, c='blue')  
    prediction = plt.scatter(np.arange(1, N+1), y_preds, c='red')  
    plt.xticks(np.arange(1, N+1))  
    plt.xlabel('# Observation')  
    plt.ylabel('Enrollments')  
    title = 'True labels vs. Predicted Labels ({}).format(model_name)  
    plt.title(title)  
    plt.legend((original, prediction), ('Original', 'Prediction'))
```

Linear regression: A baseline

```
In [46]: import statsmodels.api as sm
```

```
X_train_refined = X_train.drop(columns=['row_id'], axis=1)
linear_regression = sm.OLS(y_train, X_train_refined)
linear_regression = linear_regression.fit()
```

```
In [47]: X_test_refined = X_test.drop(columns=['row_id'], axis=1)
y_preds = linear_regression.predict(X_test_refined)
```

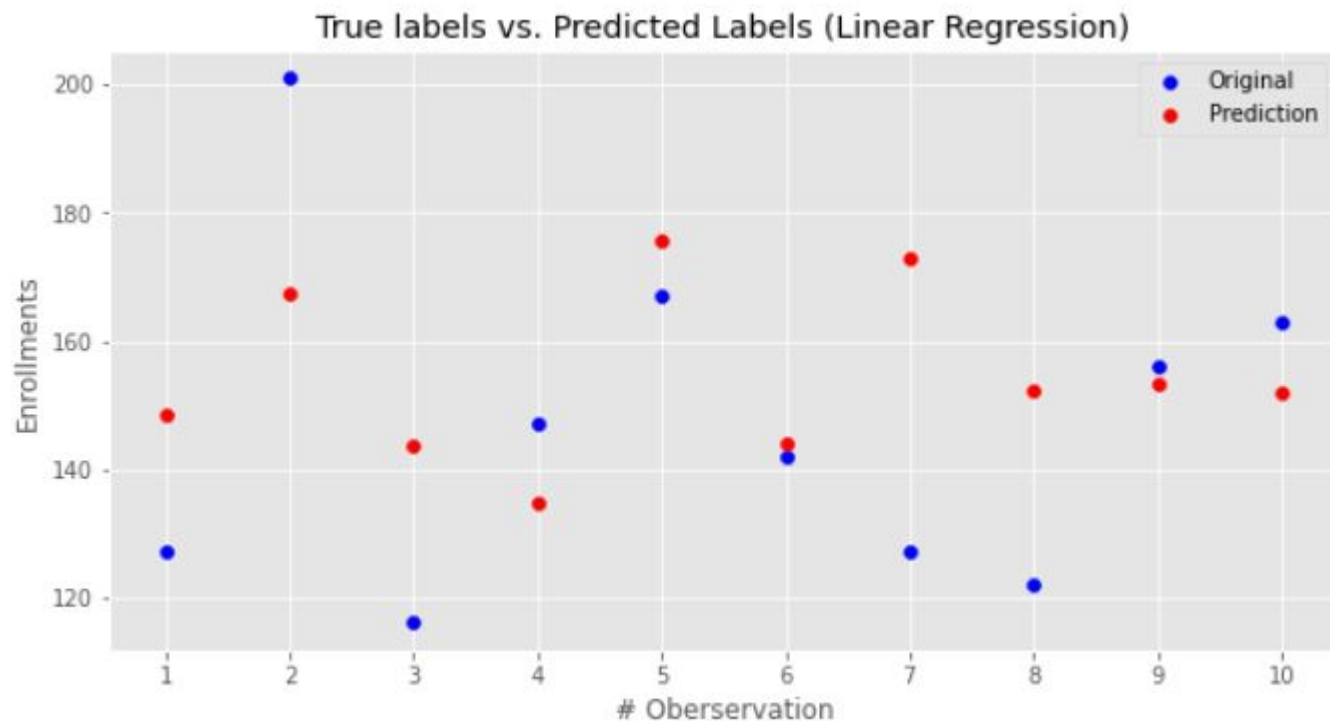
```
In [48]: calculate_metrics(y_test, y_preds)
```

RMSE Score: 23.988871314039933

R2_Squared: 0.050950018270387254

MAE Score: 19.58606183877512

```
In [49]: # Kita dapat menyelidiki prediksi dengan memvisualisasikannya menggunakan plot_preds().  
plot_preds(y_test, y_preds, 'Linear Regression')
```



```
In [50]: print(linear_regression.summary())
```

```

                                OLS Regression Results
=====
Dep. Variable:          Enrollments    R-squared (uncentered):          0.980
Model:                  OLS            Adj. R-squared (uncentered):      0.978
Method:                 Least Squares   F-statistic:                     400.1
Date:                   Sun, 20 Jun 2021 Prob (F-statistic):              7.92e-27
Time:                   22:17:49        Log-Likelihood:                  -163.65
No. Observations:       36             AIC:                            335.3
Df Residuals:           32             BIC:                            341.6
Df Model:                4
Covariance Type:        nonrobust
=====
               coef      std err          t      P>|t|      [0.025      0.975]
-----
Experiment    -1.4515      8.373      -0.173      0.863     -18.506      15.603
Pageviews      0.0672      0.010       6.593      0.000       0.046       0.088
Clicks        -0.6022      0.126     -4.776      0.000      -0.859     -0.345
DOW           -2.6156      2.137     -1.224      0.230      -6.969       1.738
=====
Omnibus:                 3.688    Durbin-Watson:                 2.146
Prob(Omnibus):            0.158    Jarque-Bera (JB):                 2.538
Skew:                     0.475    Prob(JB):                         0.281
Kurtosis:                 2.112    Cond. No.                      1.95e+04
=====
```

Decision Tree

In [52]: `from sklearn.tree import DecisionTreeRegressor`

```
dtree = DecisionTreeRegressor(max_depth=5, min_samples_leaf =4, random_state=7)
dtree.fit(X_train_refined, y_train)
y_preds = dtree.predict(X_test_refined)

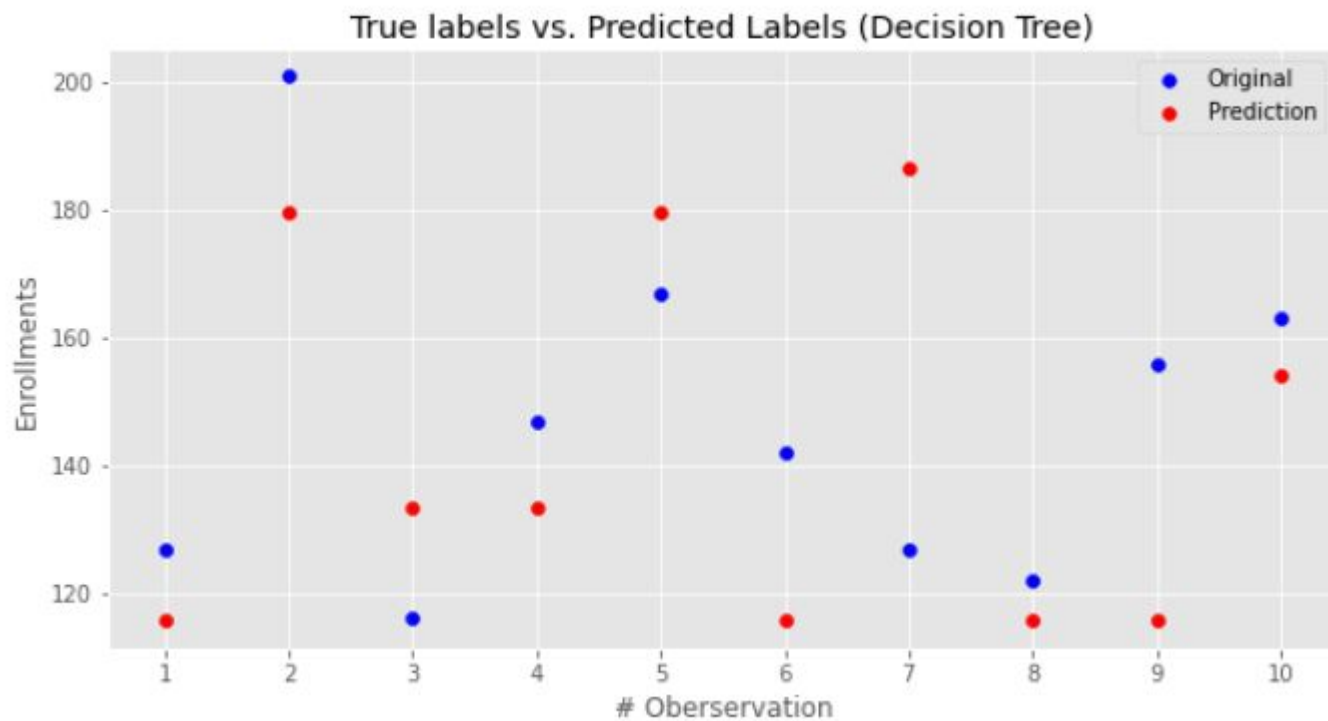
calculate_metrics(y_test, y_preds)
```

RMSE Score: 26.8540034259326

R2_Squared: -0.1892893660531696

MAE Score: 21.783333333333333

```
In [53]: plot_preds(y_test, y_preds, 'Decision Tree')
```



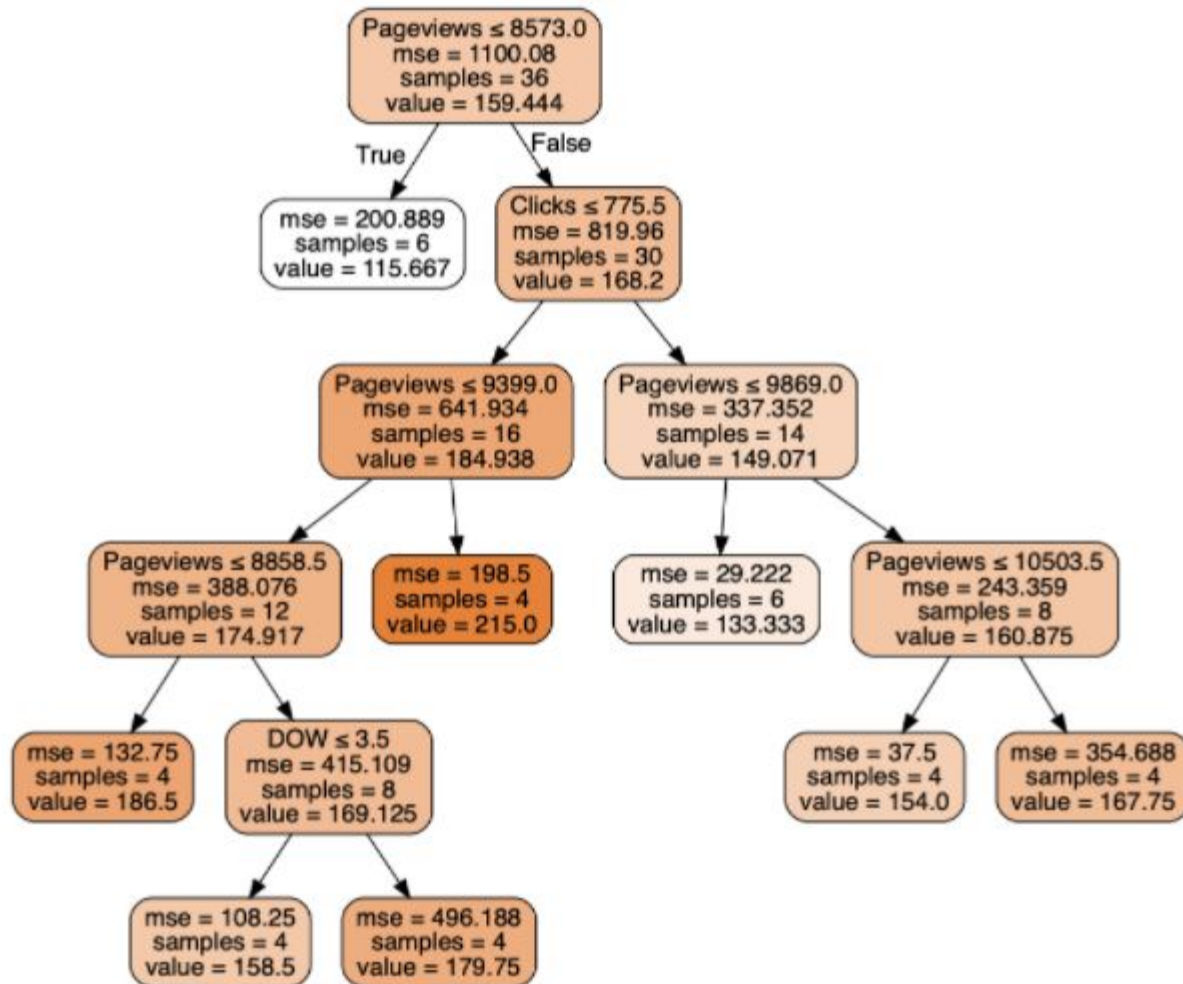
Decision tree visualization

```
In [54]: from sklearn.externals.six import StringIO
from IPython.display import Image
from sklearn.tree import export_graphviz
import pydotplus

dot_data = StringIO()

export_graphviz(dtree, out_file=dot_data,
                feature_names=X_train_refined.columns,
                filled=True, rounded=True,
                special_characters=True)

graph = pydotplus.graph_from_dot_data(dot_data.getvalue())
Image(graph.create_png())
```

XGBoos

```
In [133]: import xgboost as xgb
```

```
In [134]: DM_train = xgb.DMatrix(data=X_train_refined,label=y_train)
DM_test = xgb.DMatrix(data=X_test_refined,label=y_test)
```

```
In [220]: parameters = {
    'max_depth': 6,
    'objective': 'reg:linear',
    'booster': 'gblinear',
    'n_estimators': 1000,
    'learning_rate': 0.2,
    'gamma': 0.01,
    'random_state': 7,
    'subsample': 1.
}
```

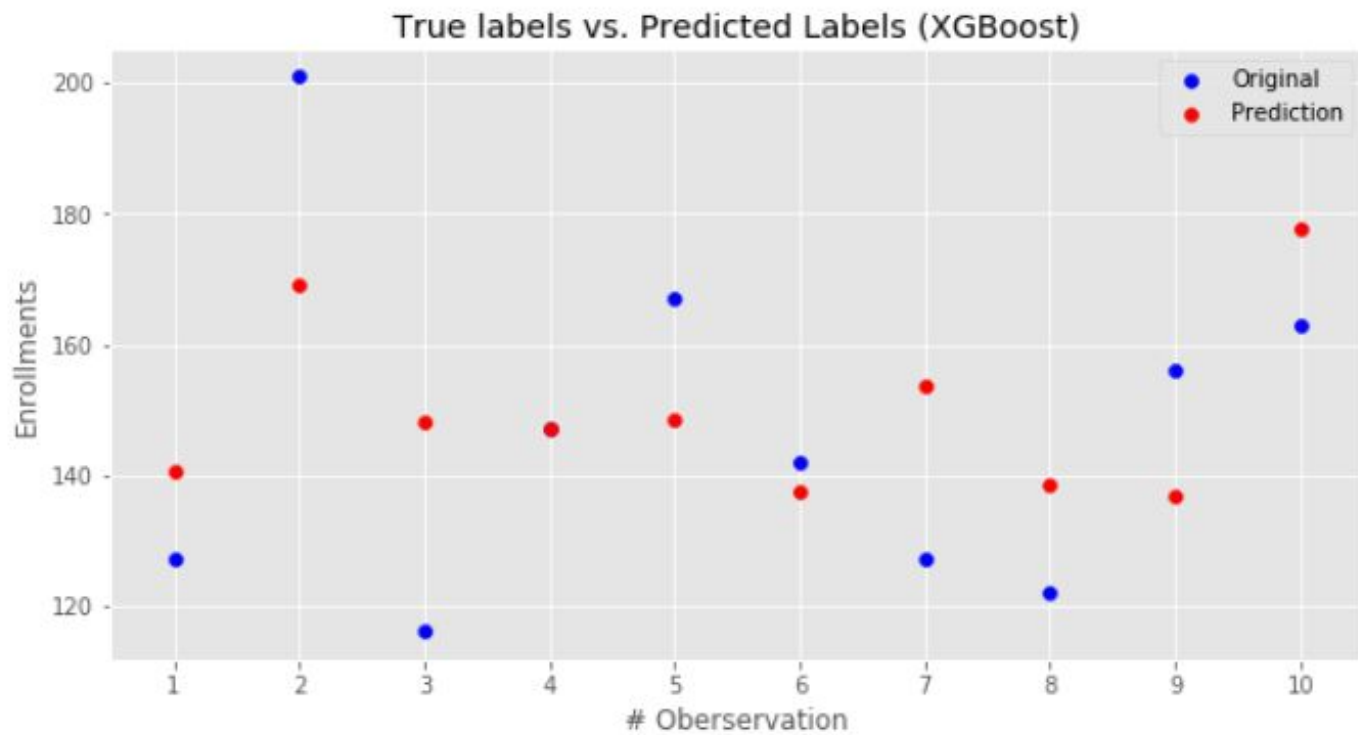
```
In [236]: parameters = {
    'max_depth': 6,
    'objective': 'reg:linear',
    'booster': 'gblinear',
    'n_estimators': 1000,
    'learning_rate': 0.2,
    'gamma': 0.01,
    'random_state': 7,
    'subsample': 1.
}
```

```
In [237]: xg_reg = xgb.train(params = parameters, dtrain=DM_train, num_boost_round=8)
y_preds = xg_reg.predict(DM_test)
```

```
In [238]: calculate_metrics(y_test, y_preds)
```

RMSE Score: 20.394753742379482
R2_Squared: 0.3140280028163105
MAE Score: 17.774598693847658

```
In [239]: plot_preds(y_test, y_preds, 'XGBoost')
```



SARAN

- Jika Udacity ingin memaksimalkan pendaftaran, Udacity harus fokus pada peningkatan Page Views dari kandidat yang memenuhi syarat. Page Views adalah fitur terpenting dalam 2 dari 3 model.
- Jika Udacity ingin mengingatkan orang tentang komitmen waktu, formulir sembulan tambahan diharapkan dapat mengurangi jumlah pendaftaran. Dampak negatif dapat dilihat pada pohon keputusan (ketika $\text{Experiment} \leq 0,5$, Enrollments turun) dan dalam istilah model Linear Regression (-17,6 Enrollments saat $\text{Experiment} = 1$).

KESIMPULAN

- Sistem didorong oleh Page Views dan Click. Inferensi Statistik tidak akan mengidentifikasi driver ini tapi Machine Learning yang mengidentifikasinya.
- Eksperimen = 1 menurunkan pendaftaran sebesar -17,6 Pendaftaran Per Hari dalam Regresi Linier. Untuk Decision Tree juga terlihat penurunan serupa. Inferensi statistik tidak akan mengidentifikasi besarnya dan arah. Hanya apakah Eksperimen itu berpengaruh atau tidak.

DOKUMENTASI KELOMPOK

Google Drive

https://drive.google.com/drive/folders/1ZNisl7mRm_Q7ok3RZRefmQfm21SapzT9?usp=sharing

GitHub

<https://github.com/lailikanabila/AB-Testing-With-Machine-Learning>

DAFTAR PUSTAKA

Dataset :

Control Data : <https://www.kaggle.com/tammyrotem/control-data>

Experiment Data : <https://www.kaggle.com/tammyrotem/experiment-data>

Eksperiment :

<https://www.business-science.io/business/2019/03/11/ab-testing-machine-learning.html>

<https://github.com/sayakpaul/A-B-testing-with-Machine-Learning.git>