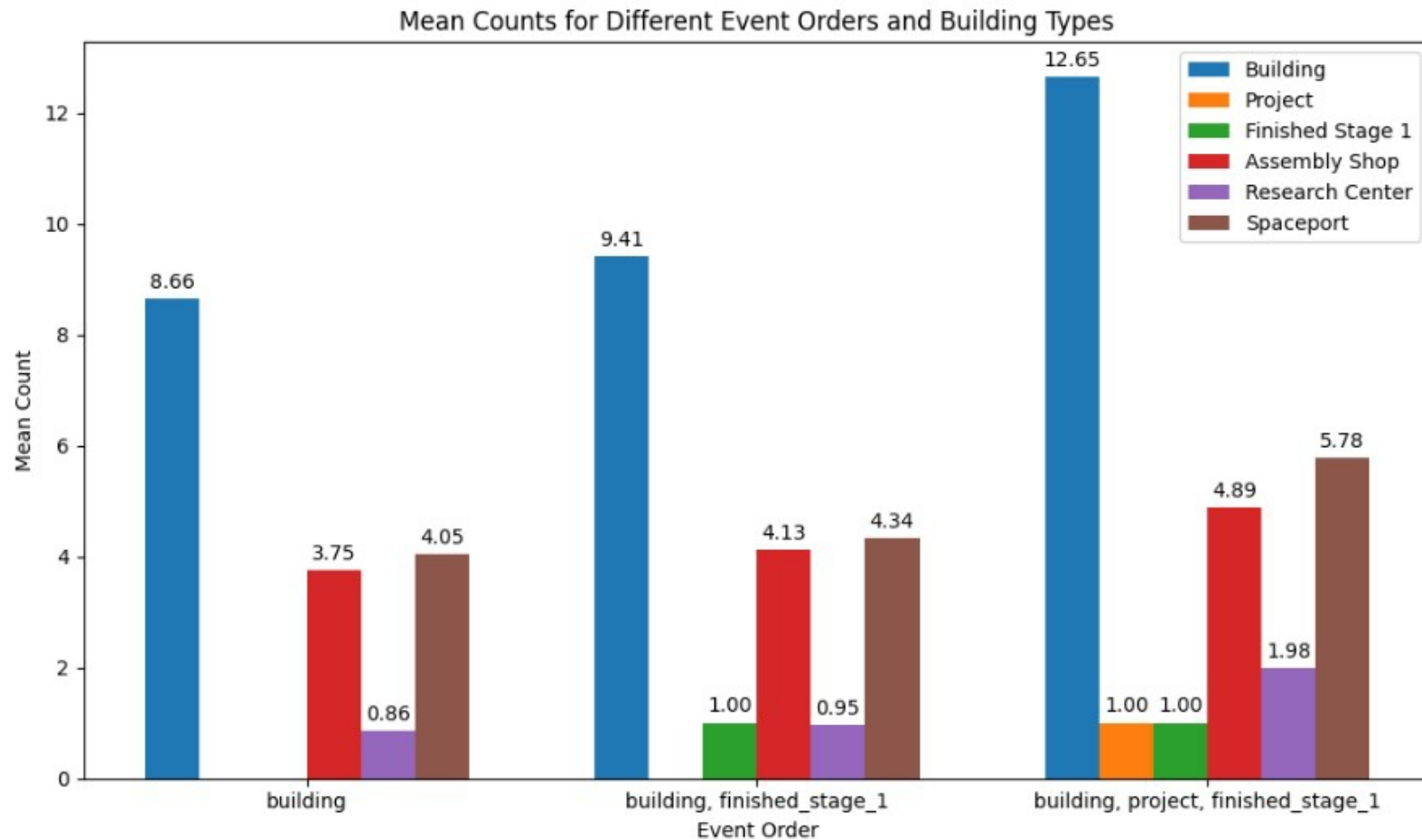


Game : Creating a Monetization Model



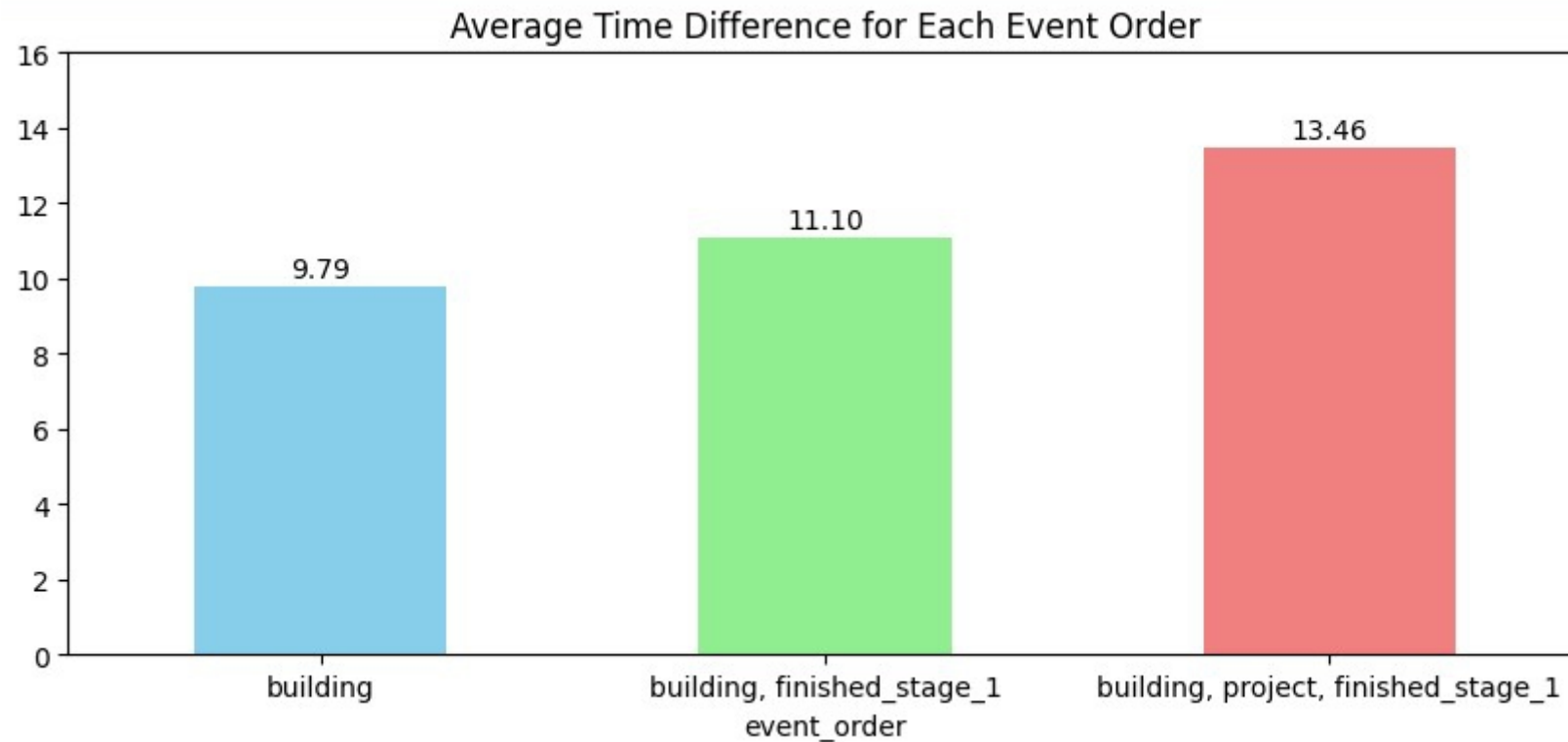
Adi Kusuma

User mana yang melakukan banyak 'build'?



User yang 'finished_stage_1' dengan 'project' melakukan banyak 'build'

Berapa lama waktu yang digunakan user untuk melakukan banyak 'build' ?



User yang 'finished_stage_1' dengan 'project' membutuhkan waktu rata-rata yang lebih lama dari user yang melawan boss

Berapa Optimal Cost ?

Hasil dari (5.2.3 Uji Kruskal-Wallis source dengan time_difference) tidak terdapat perbedaan yang signifikan maka cost pada semua source dijumlahkan dan melakukan 'Machine Learning' untuk mendapatkan 'optimal cost'

```
[238]: # Pilih fitur yang relevan dan variabel target
features = pivot_table[['% building, finished_stage_1', '% building, project, finished_stage_1']]
target = pivot_table['sum_cost'] # target variable adalah 'optimal cost' dari kolom 'sum_cost'

# Bagi data menjadi set pelatihan dan pengujian
X_train, X_test, y_train, y_test = train_test_split(features, target, test_size=0.2, random_state=42)

# Buat model regresi linier
model = LinearRegression()

# Latih modelnya
model.fit(X_train, y_train)

# Buat prediksi pada set tes
predictions = model.predict(X_test)

# Tambahkan prediksi ke DataFrame
X_test['predictions'] = predictions

# Urutkan DataFrame berdasarkan prediksi
sorted_df = X_test.sort_values(by='predictions', ascending=False)

# Menampilkan DataFrame yang diurutkan
sorted_df
```

```
[238]: event_order  % building, finished_stage_1  % building, project, finished_stage_1  predictions
      event_date
2020-05-04                29.846416                15.716724  1278.740415
2020-05-05                30.258852                14.757513  1221.963516
```

'optimal cost' yang didapatkan adalah 1221.963516 dan 1278.740415

Alokasi 'Optimal Cost' pada setiap source ?

allocation_df							
	source	allocation_absolute	efficiency_ratio	rank_efficiency_ratio	percentage_allocation	optimal_cost_1221	optimal_cost_1278
0	facebook_ads	9.647416	1.273294	4	16.991797	207.633564	217.280980
1	google_play_ads	16.343637	2.157080	2	28.785716	351.750950	368.094587
2	instagram_new_adverts	11.732610	1.548503	3	20.664409	252.511534	264.244144
3	youtube_channel_network	19.053236	2.514701	1	33.558078	410.067468	429.120704

Facebook Ads memiliki optimal cost range antara 207.63 dan 217.28.

Google Play Ads memiliki optimal cost range antara 351.75 dan 368.09.

Instagram New Adverts memiliki optimal cost range antara 252.51 dan 264.24.

Youtube Channel Network memiliki optimal cost range antara 410.07 dan 429.12.

Prediksi jumlah pengguna 'Optimal Cost'

```
[231]: # Misalkan data Anda disimpan dalam DataFrame 'df'
X = pivot_table[['sum_cost']] # Variabel independen
y = pivot_table['count_user_id'] # Variabel dependen

# Pemisahan data menjadi data pelatihan dan data uji
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Inisialisasi dan melatih model regresi linear
model = LinearRegression()
model.fit(X_train, y_train)

# Evaluasi model
y_pred = model.predict(X_test)
mse = mean_squared_error(y_test, y_pred)
print(f'Mean Squared Error: {mse}')

# Prediksi count_user_id untuk rentang sum_cost yang diinginkan
sum_cost_range = np.array([[1221.963516], [1278.740415]])
predicted_count_user_id = model.predict(sum_cost_range)

print(f'Predicted count_user_id for sum_cost range: {predicted_count_user_id}')
```

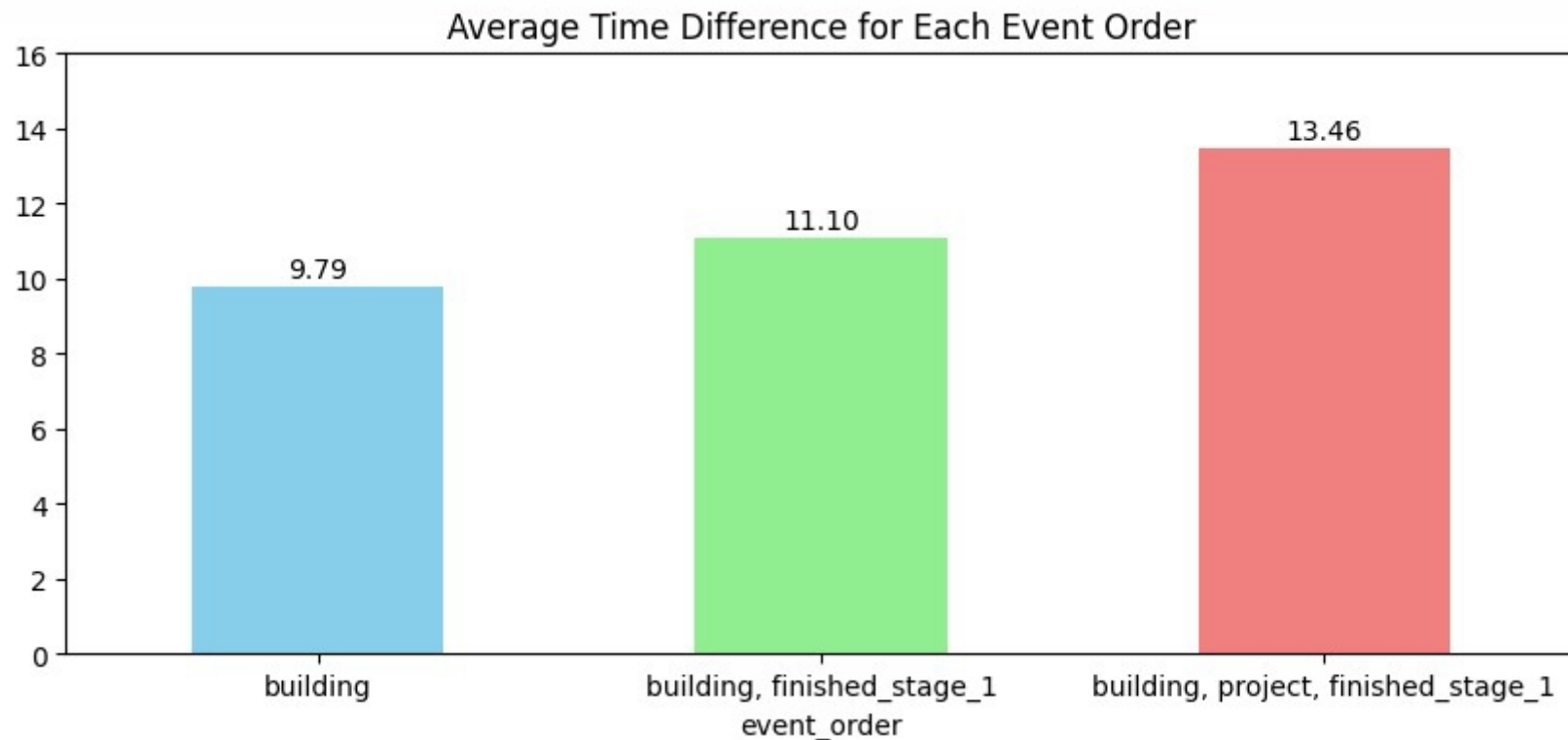
Mean Squared Error: 2124.109067997294

Predicted count_user_id for sum_cost range: [2189.44104224 2291.32341333]

Untuk 'Optimal Cost' 1221.963516, model memprediksi count_user_id sekitar 2189.44.
Untuk 'Optimal Cost' 1278.740415, model memprediksi count_user_id sekitar 2291.32.

Rentang waktu mengeluarkan 'Optimal Cost'

Berdasarkan uji ANOVA pada index (5.2.10 Uji ANOVA day_name dengan hour) tidak terdapat perbedaan yang signifikan antara hour dengan day_name. Semua hari dan jam mempunyai frekuensi yang tidak berbeda, jadi kita bisa menempatkan 'Optimal Cost' atau iklan kapan saja



mengeluarkan 'Optimal Cost' disarankan dalam rentang waktu mean_time_difference_days yaitu 5 sampai 10 hari (sebelum menyelesaikan 'stage 1').

Nilai Cost dan Monetize

Untuk monetisasi iklan memakai kolom monetize pada summary_df sebagai acuan harga dengan keuntungan minimal 10% dikali dengan normalized

summary_df						
	building	mean	%	normalized	cost	monetize
0	count_building_assembly_shop	4.013995	38.50	4.01	0.224859	0.625859
1	count_building_research_center	1.041323	9.99	1.04	0.058346	0.162346
2	count_building_spaceport	4.369844	41.92	4.37	0.244833	0.681833
3	count_event_project	1.000000	9.59	1.00	0.056010	0.156010

Building Assembly Shop memiliki biaya (cost) sebesar 0.224859 dan pendapatan monetize sebesar 0.625859.
 Building Research Center memiliki biaya (cost) sebesar 0.058346 dan pendapatan monetize sebesar 0.162346.
 Building Spaceport memiliki biaya (cost) sebesar 0.244833 dan pendapatan monetize sebesar 0.681833.
 Event Project memiliki biaya (cost) sebesar 0.056010 dan pendapatan monetize sebesar 0.156010.

SARAN

Perlu analisa lanjutan dan 'continues' agar mendapatkan model terbaik untuk mendapatkan keuntungan optimal