
Data Science
:: Bank
Marketing(Ca
mpaign)

Name : Vasu Sharma
Email : vasu.vs45@gmail.com
College : Indiana University Bloomington
Country : United States of America

GitHub Repository : https://github.com/vasuvs45/Project

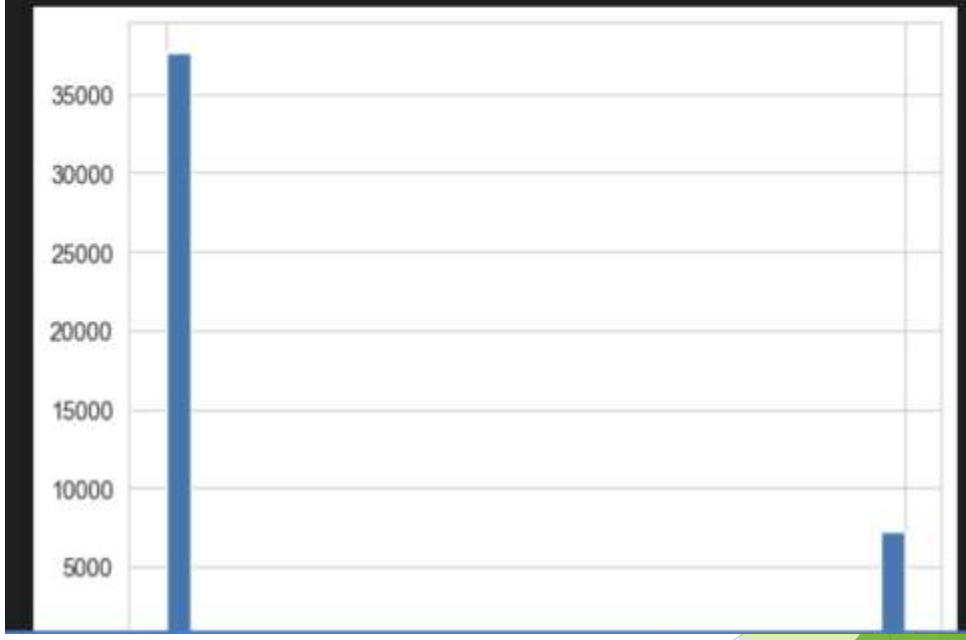
The EDA that’s performed on the provided data sets are as follows:

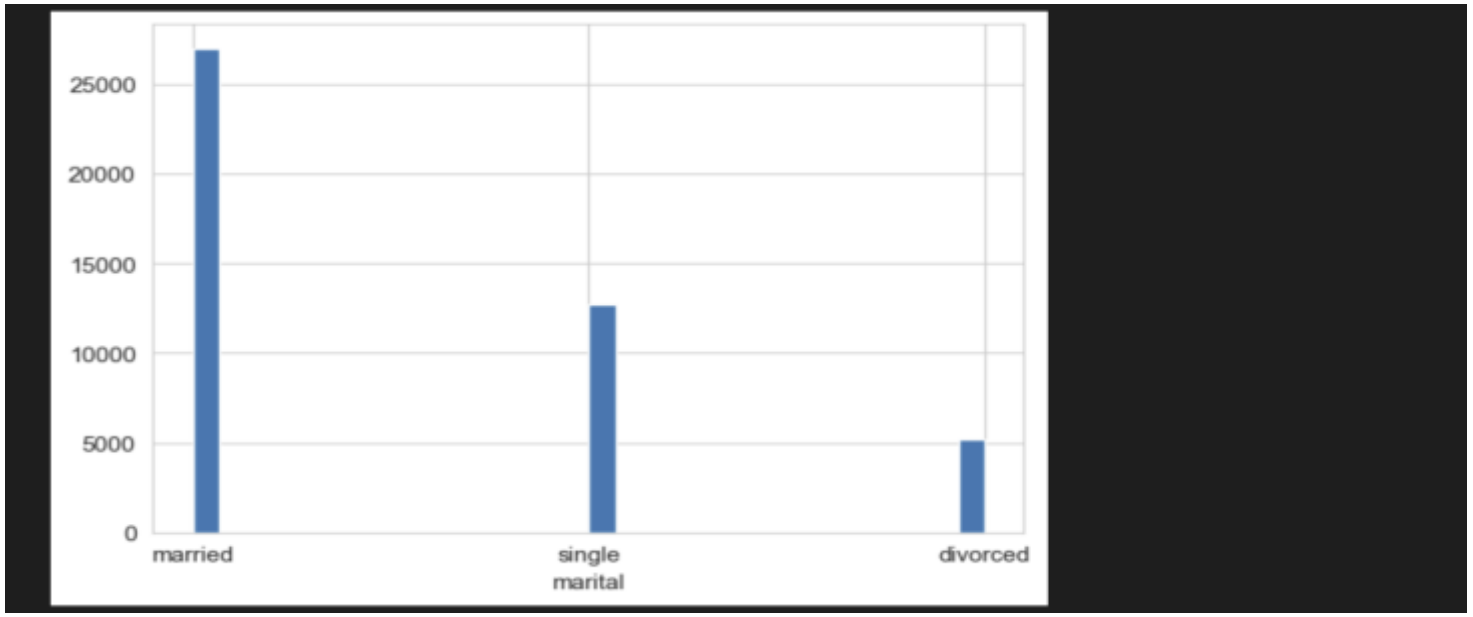
```
sns.set_style('whitegrid')
bank_df_filtered['y'].hist(bins=30)
plt.xlabel('y')
```

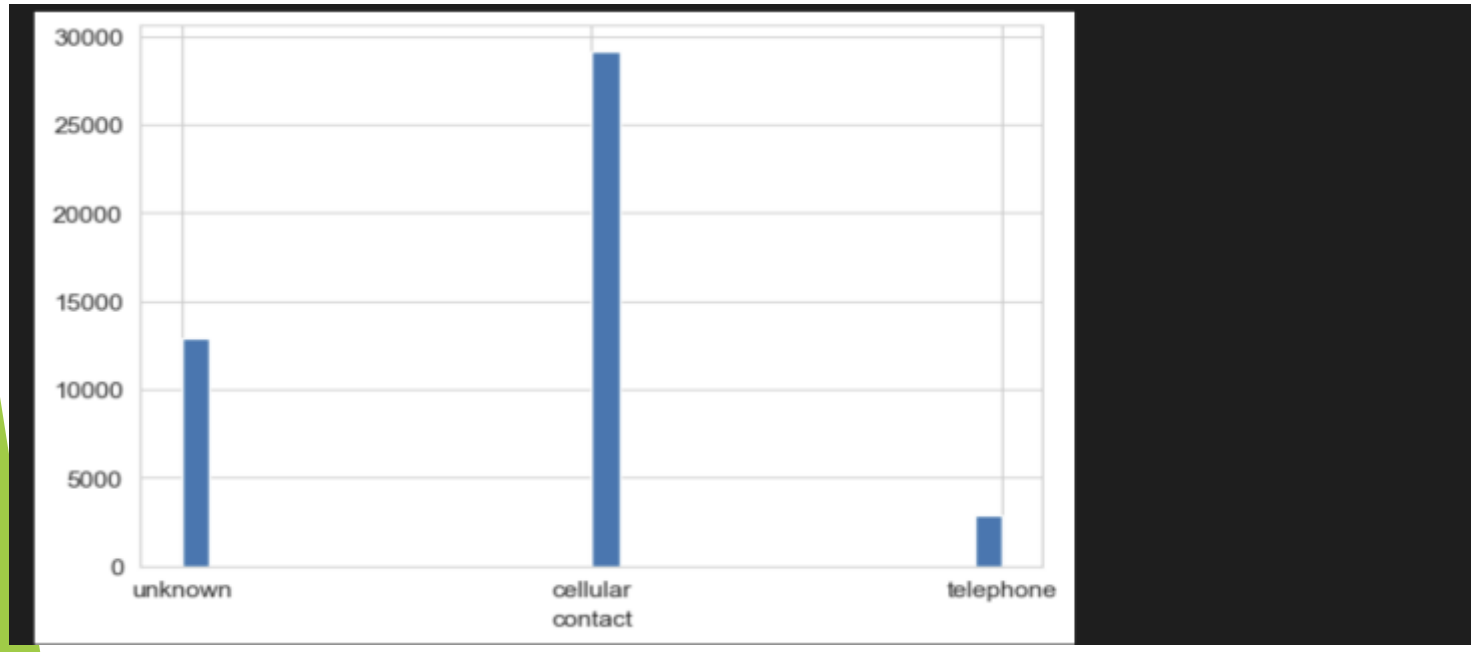
```
Text(0.5, 0, 'y')
```

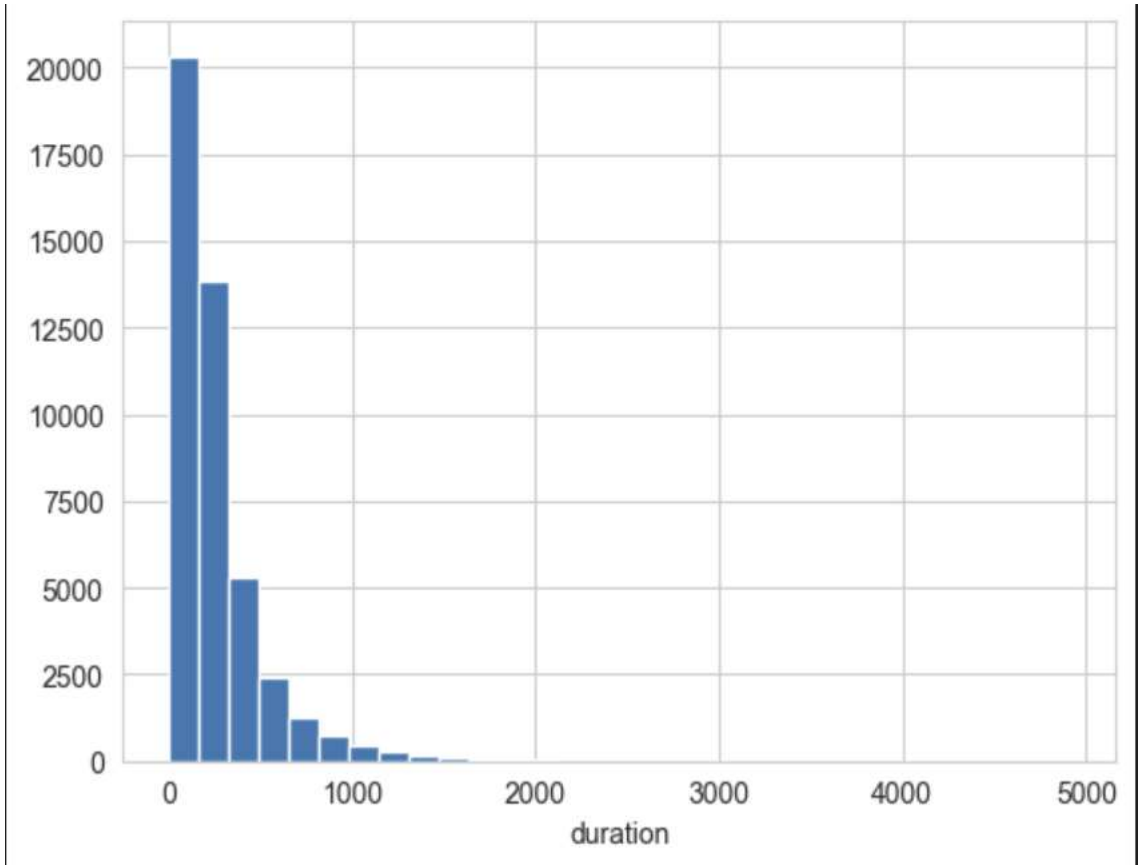

```
sns.set_style('whitegrid')
bank_df_filtered['loan'].hist(bins=30)
plt.xlabel('loan')
```

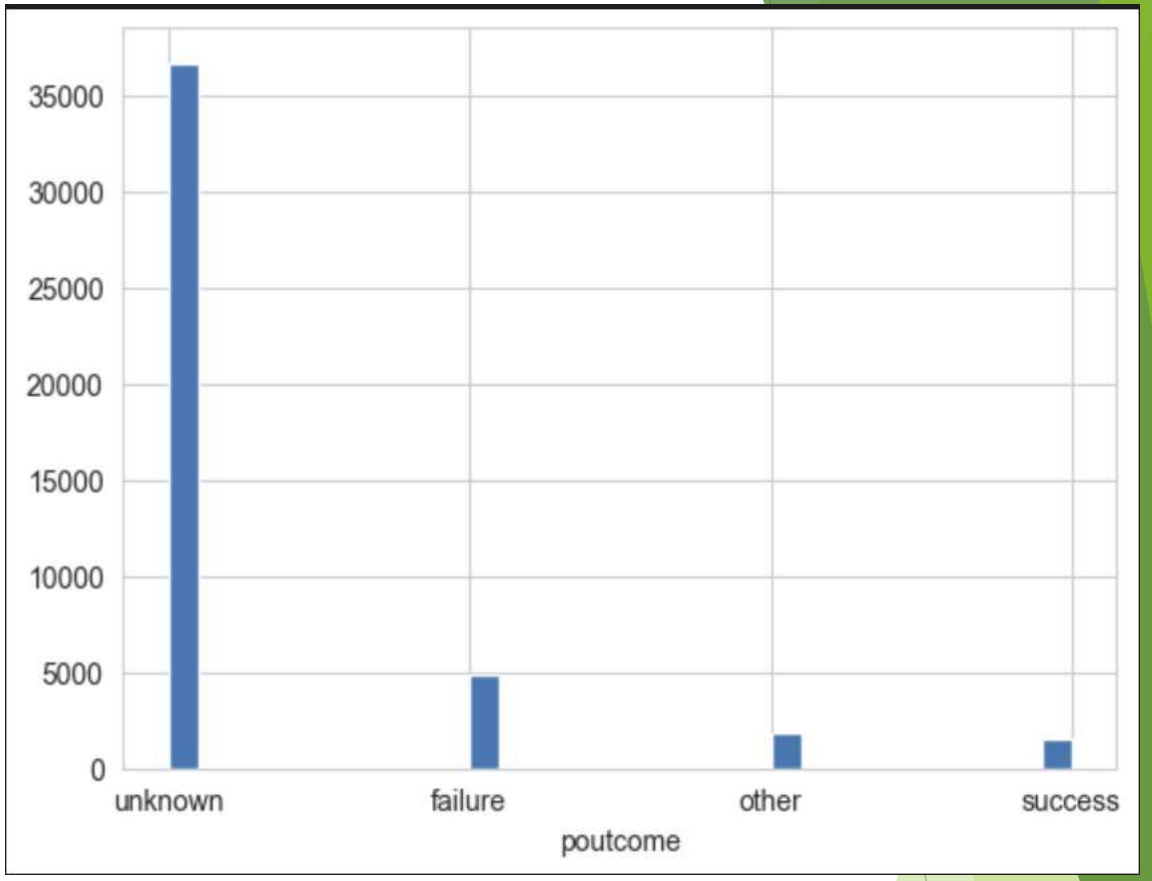
```
Text(0.5, 0, 'loan')
```

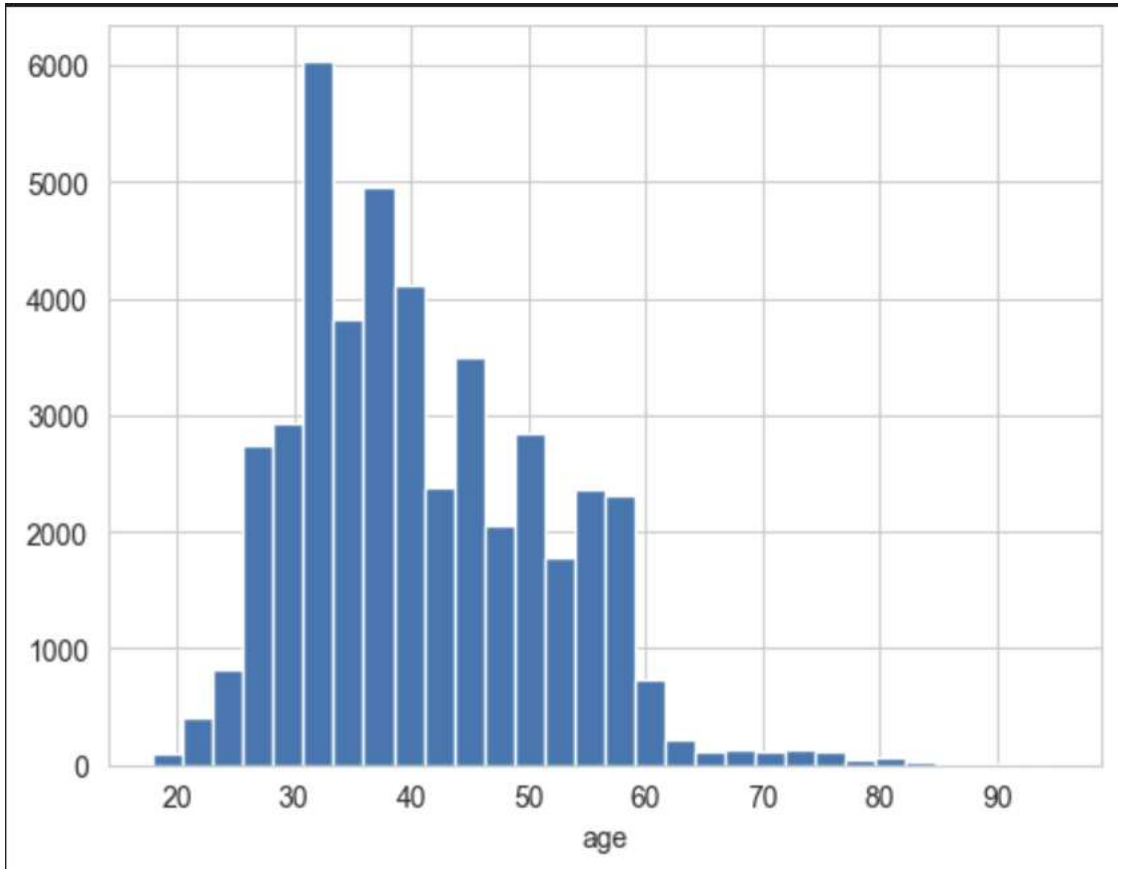


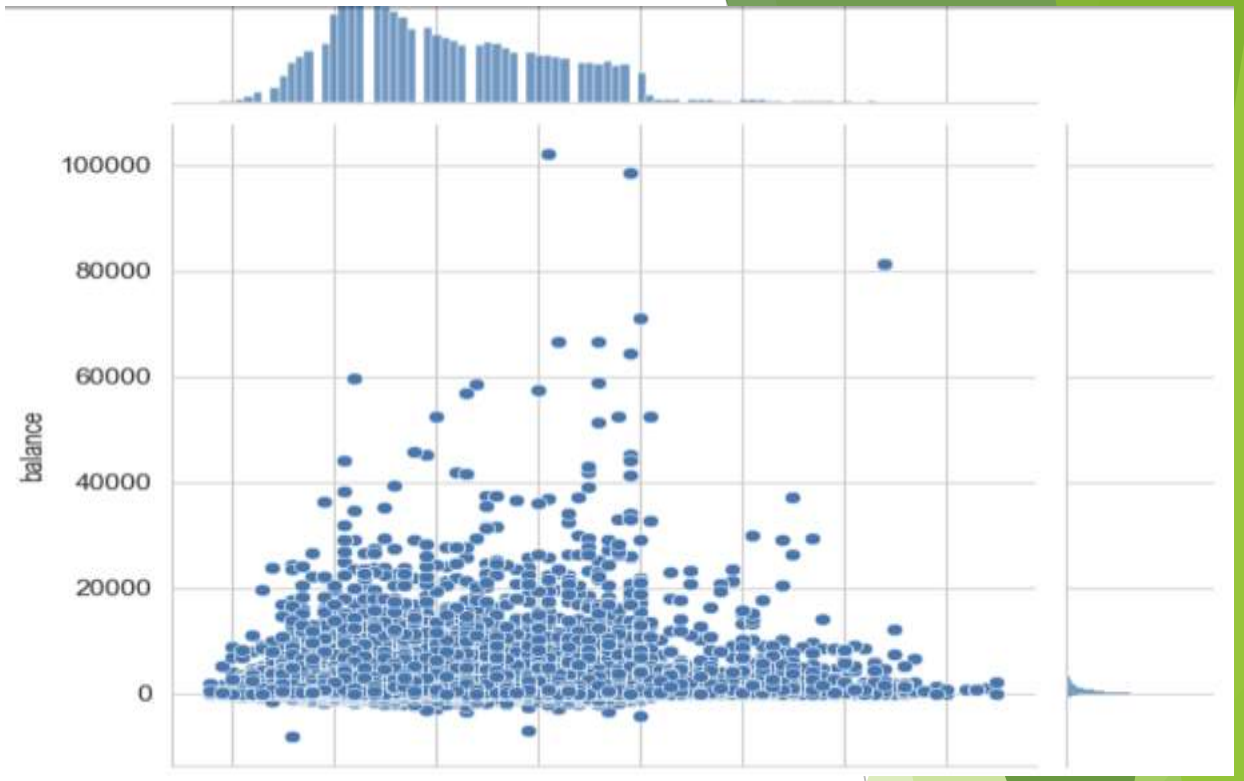

25000
20000
15000
10000
5000
0
married
single
divorced
marital


30000
25000
20000
15000
10000
5000
0
unknown
cellular
telephone
contact


20000
17500
15000
12500
10000
7500
5000
2500
0
0
1000
2000
3000
4000
5000
duration


35000
30000
25000
20000
15000
10000
5000
0
unknown
failure
other
success
poutcome


6000
5000
4000
3000
2000
1000
0
20
30
40
50
60
70
80
90
age

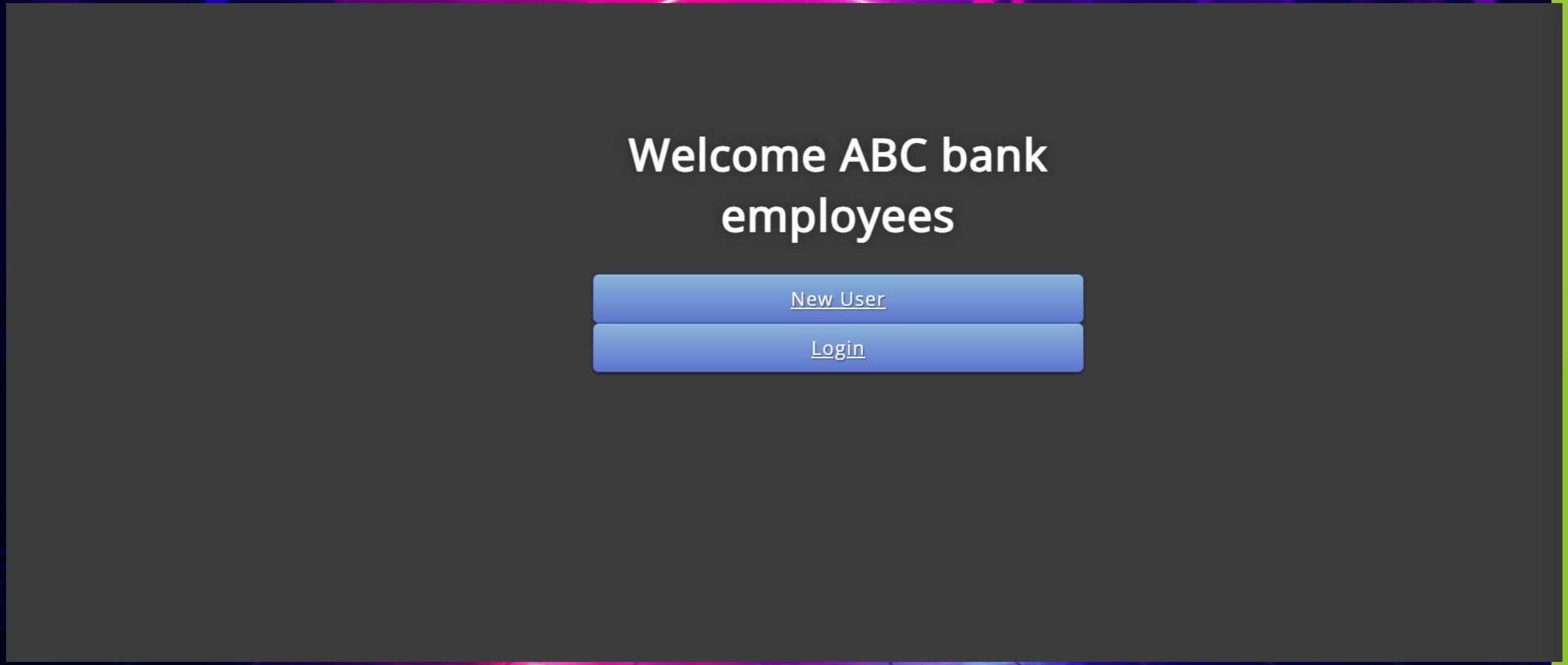

100000
80000
60000
40000
20000
0
balance

Technique to remove imbalance

Oversampling

```
#pip install imbalanced-learn
from imblearn.over_sampling import SMOTE
from imblearn.combine import SMOTETomek
final_dataset = pd.read_csv('final_dataset.csv')
final_dataset.head(1)
X=final_dataset[['age','job','marital','education','default','balance','housing','loan','contact','day','duration','cam
y = final_dataset['y']
smote = SMOTE(sampling_strategy='minority',random_state=7)
X_resampled, y_resampled = smote.fit_resample(X, y)
fig, ax = plt.subplots(figsize=(20,10))
# Imbalanced DataFrame Correlation
corr = final_dataset.corr()
sns.heatmap(corr, cmap='YlGnBu', annot_kws={'size':30}, ax=ax)
ax.set_title("Imbalanced Correlation Matrix", fontsize=14)
plt.show()
oversampled_train = pd.concat([pd.DataFrame(X_resampled), pd.DataFrame(y_resampled)], axis=1)
oversampled_train.head(1)
oversampled_train.to_csv('oversampled_train.csv')
```

Tthe Website that has been creted and ML algorithms that are created are as follows:



Already Have An Account? Sign In

Join Today

Username

vasu369

Email

admin@blog.com

Password

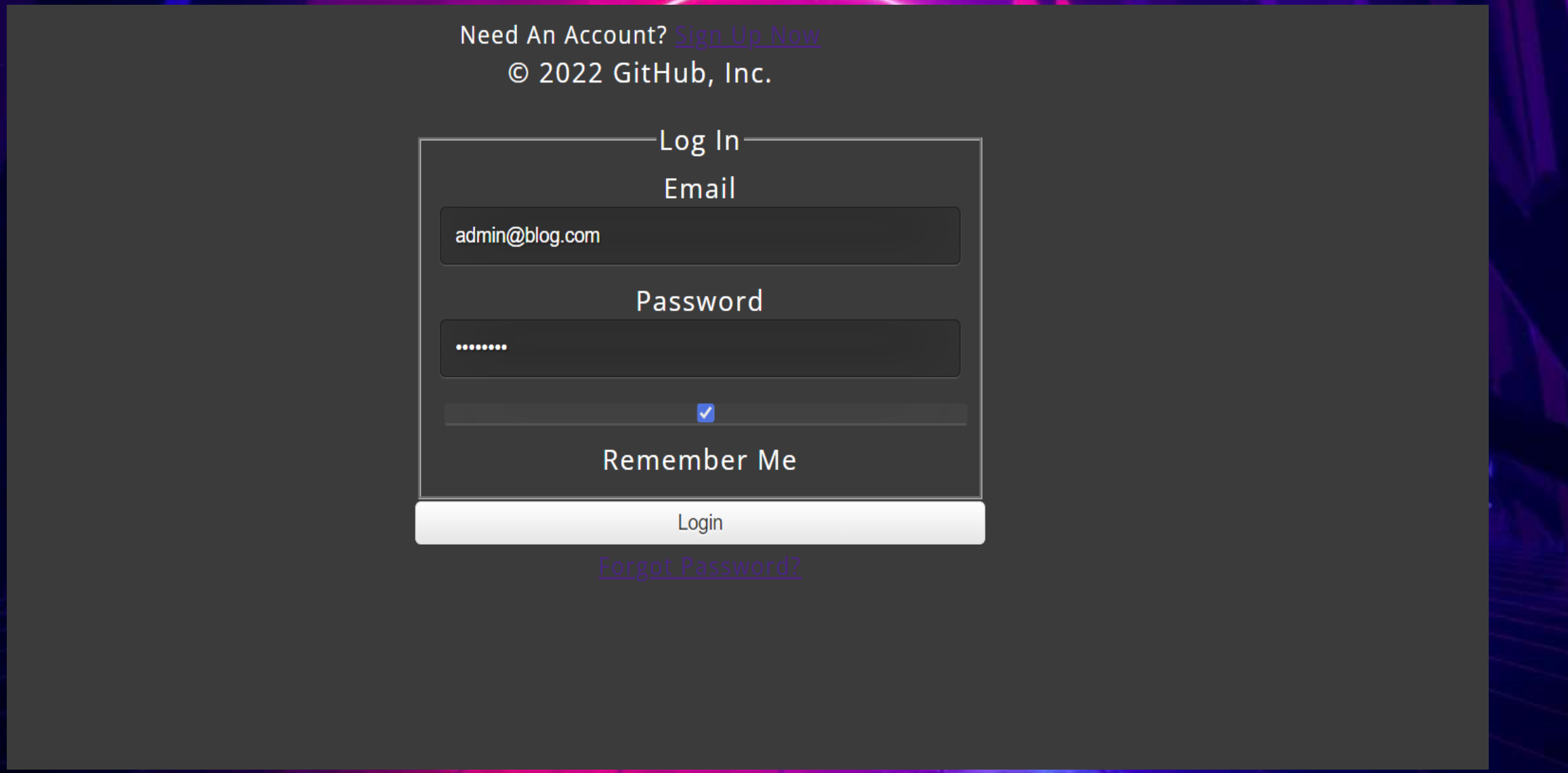
••••••••

Confirm Password

••••••••

Field must be equal to password.

Sign Up


Need An Account? Sign Up Now
© 2022 GitHub, Inc.
Log In
Email
admin@blog.com
Password
••••••••
Remember Me
Login
Forgot Password?

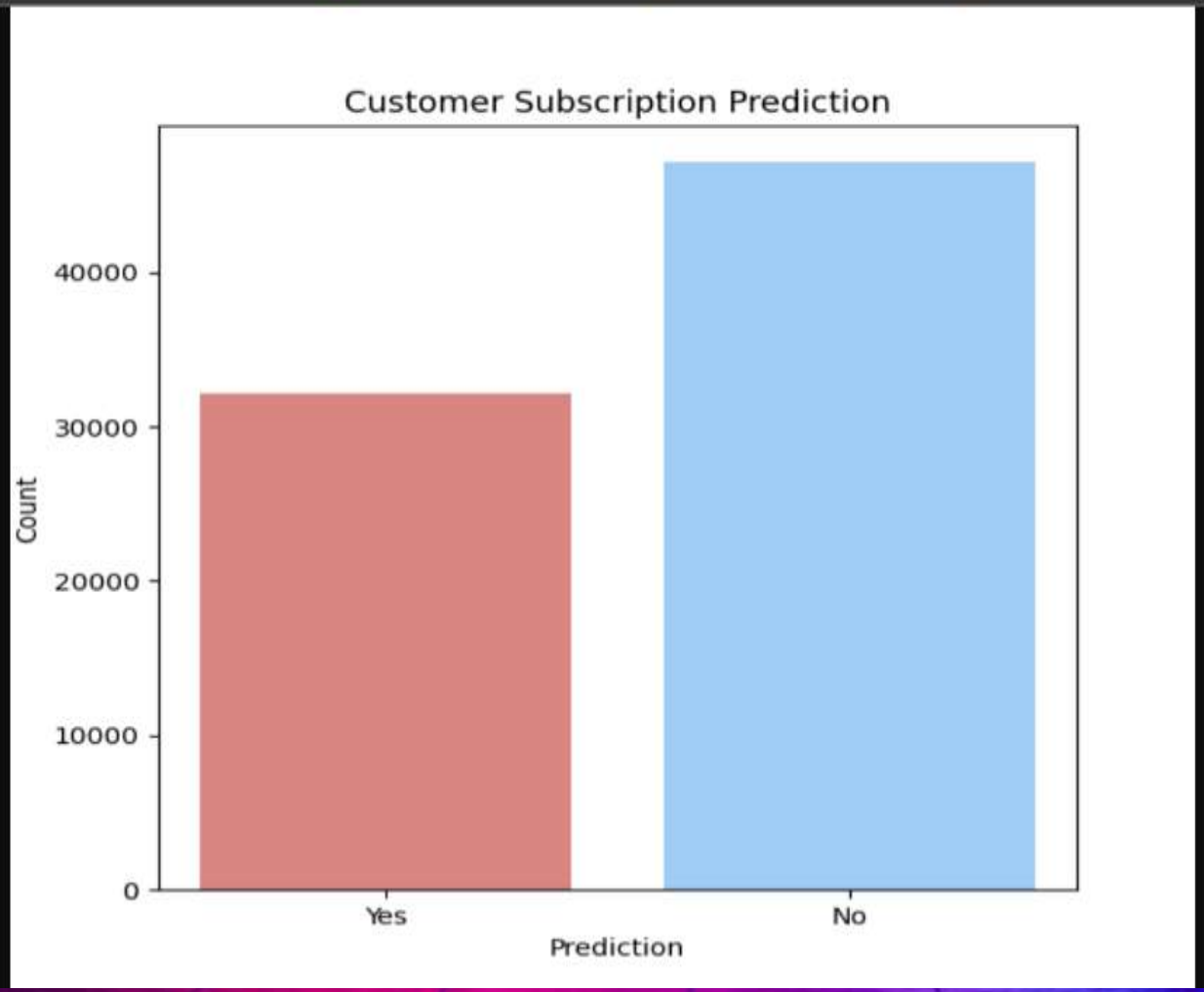
Select the algorithm to predict

- Logistic Regression
- K-Means
- Random Forest
- Support Vector Machine

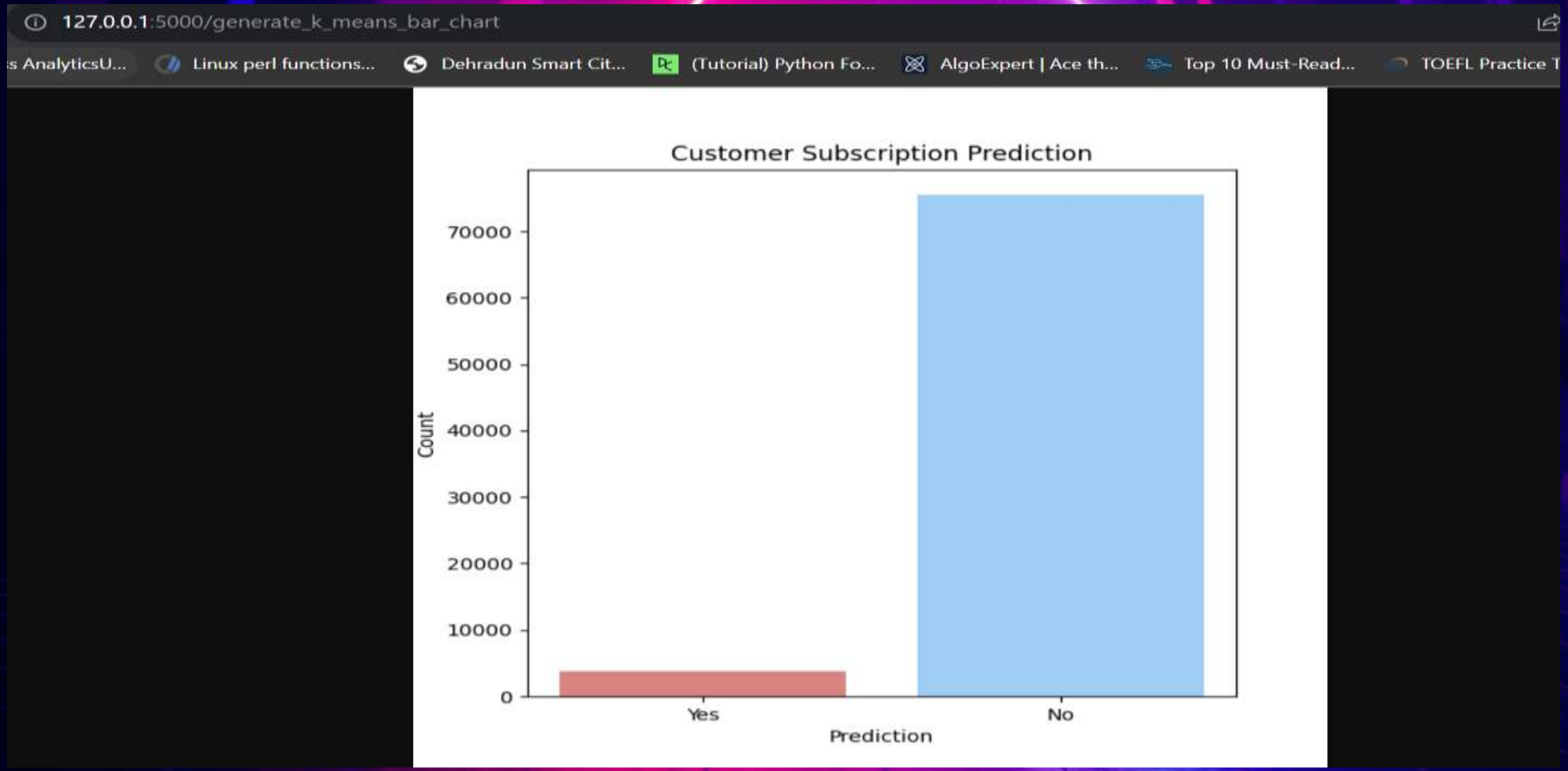
Logistic Regression

127.0.0.1:5000/generate_logistic_bar_chart

AnalyticsU... Linux perl functions... Dehradun Smart Cit... (Tutorial) Python Fo... AlgoExpert | Ace th... Top 10 Must-Read... TOEFL Practice Test



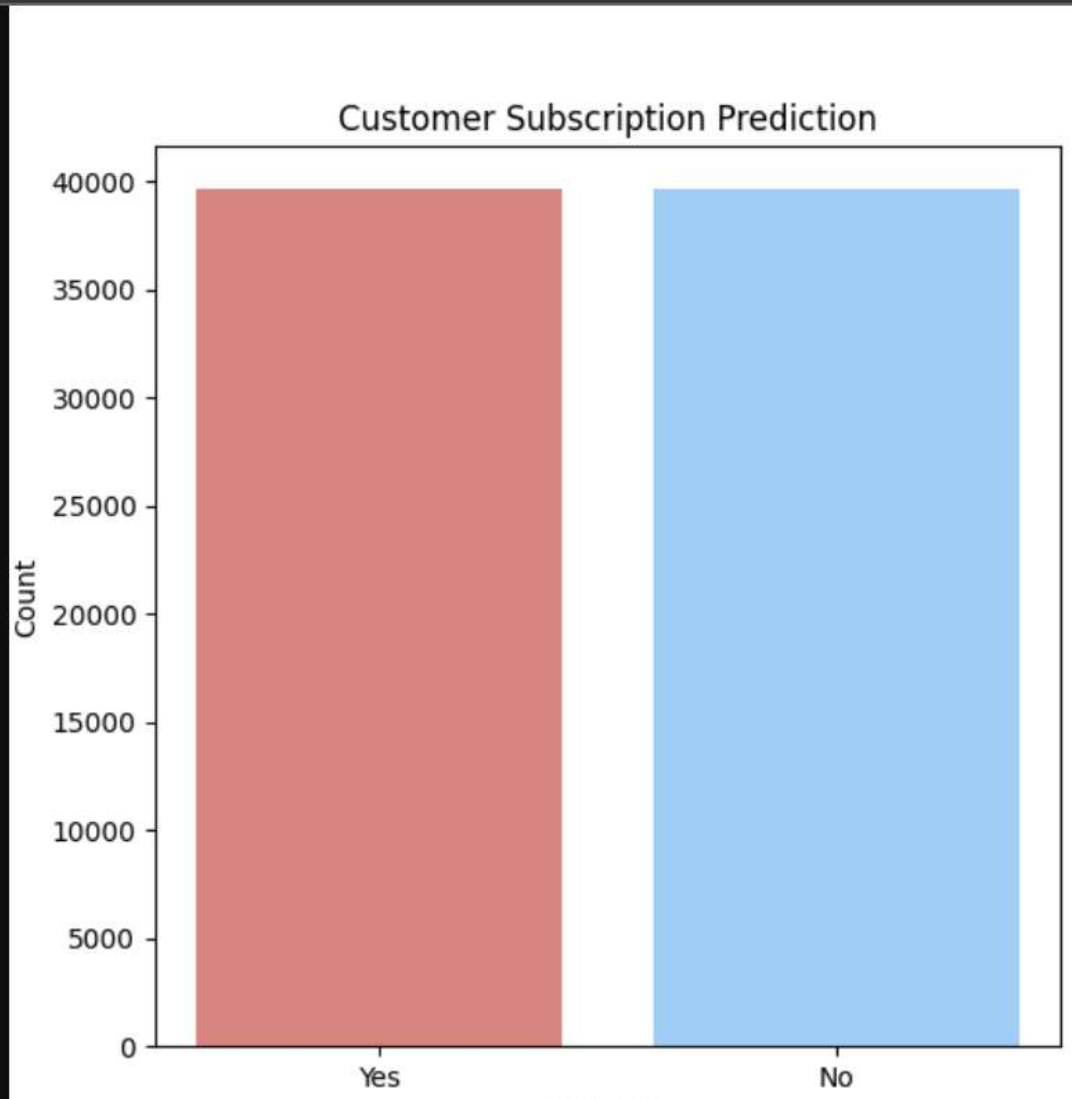
K-Means Algorithm



Random Forest

127.0.0.1:5000/generate_random_forest_chart

ess AnalyticsU... | Linux perl functions... | Dehradun Smart Cit... | (Tutorial) Python Fo... | AlgoExpert | Ace th... | Top 10 Must-Read... | TOEFL Practice Test | » | Other bookr



Support Vector Machine

In the end, I would suggest using either Logistic Regression or K-Means would be the best choice as the visualization and decisions made based on these algorithms seems more logical. Also Logistic Regression provides more accurate results as compared to other ML algorithms. Hence I would suggest using either Logistic Regression or K-Means ML algorithms.