DECEMBER 05, 2018

PREDICT BEHAVIOR TO RETAIN CUSTOMERS

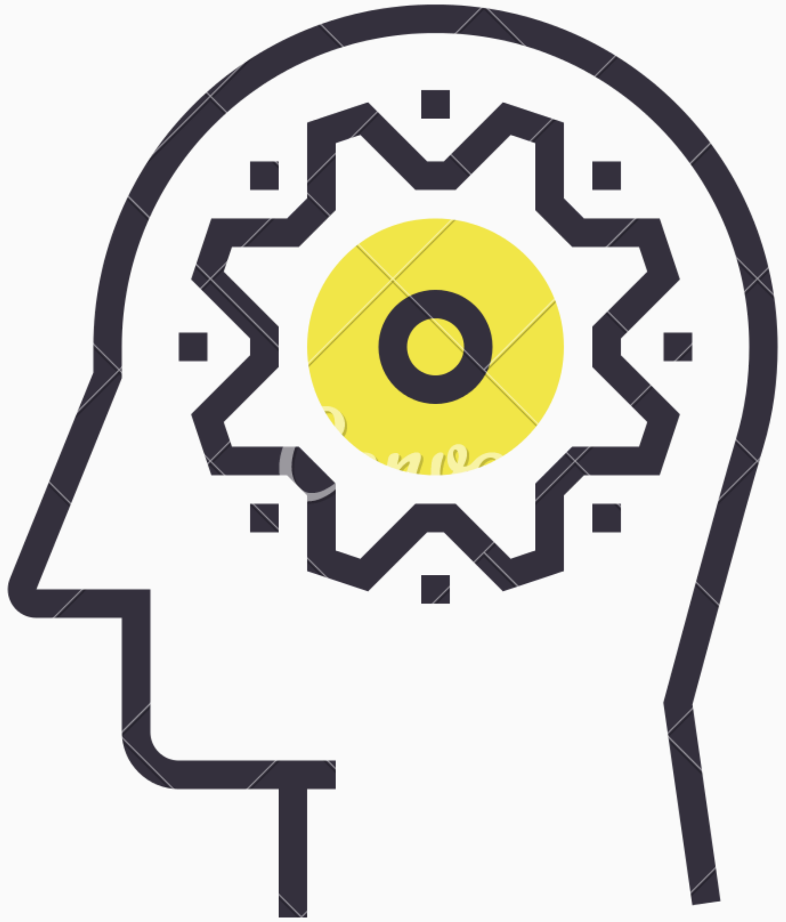
Telco Customer Churn

Oleh :

Faiz Naufal Wardhana

Muhammad Savero

Yusuf Rohmatu Rifa'i



Outline Presentasi

What You Can Expect:

Latar Belakang

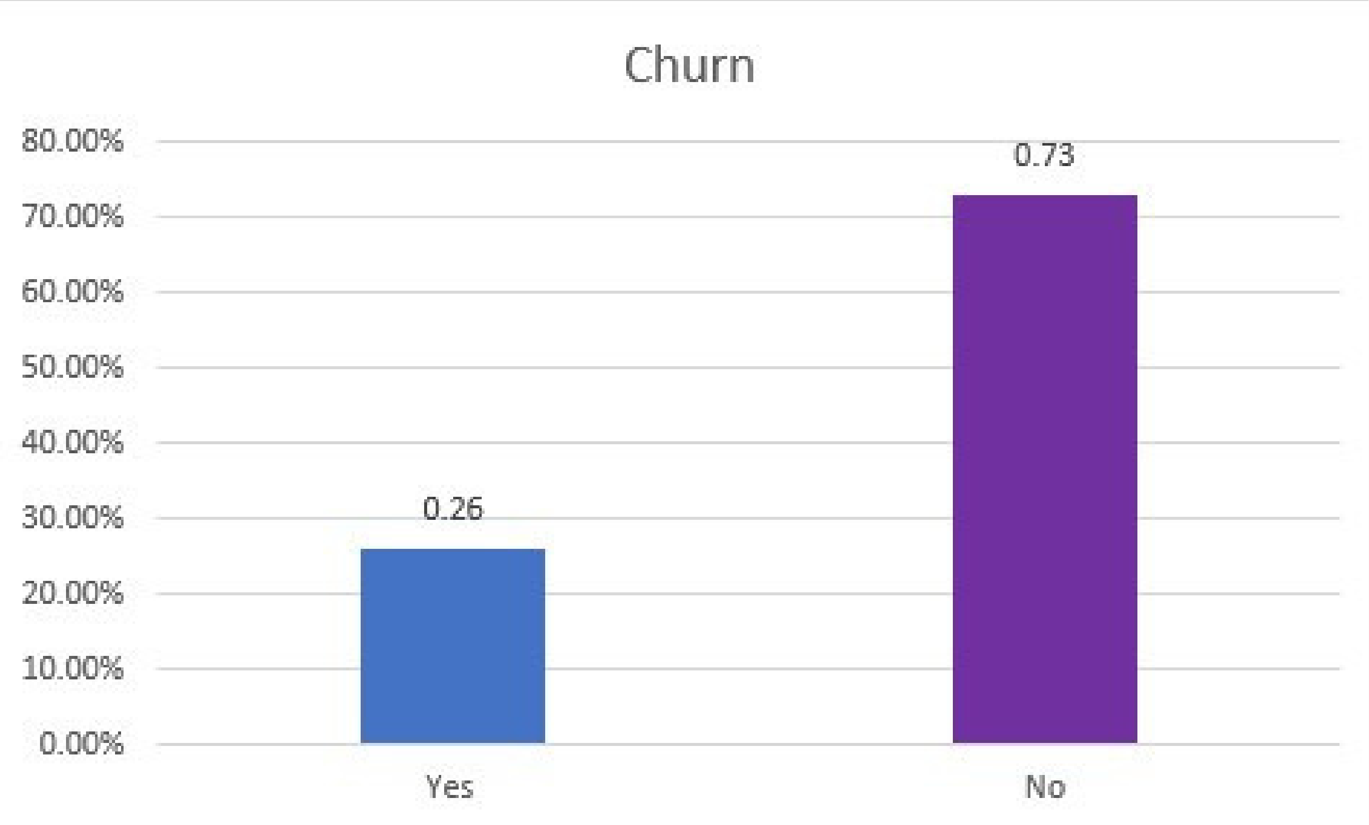
Tujuan

Metode

Pembahasan

Analisis

Latar Belakang



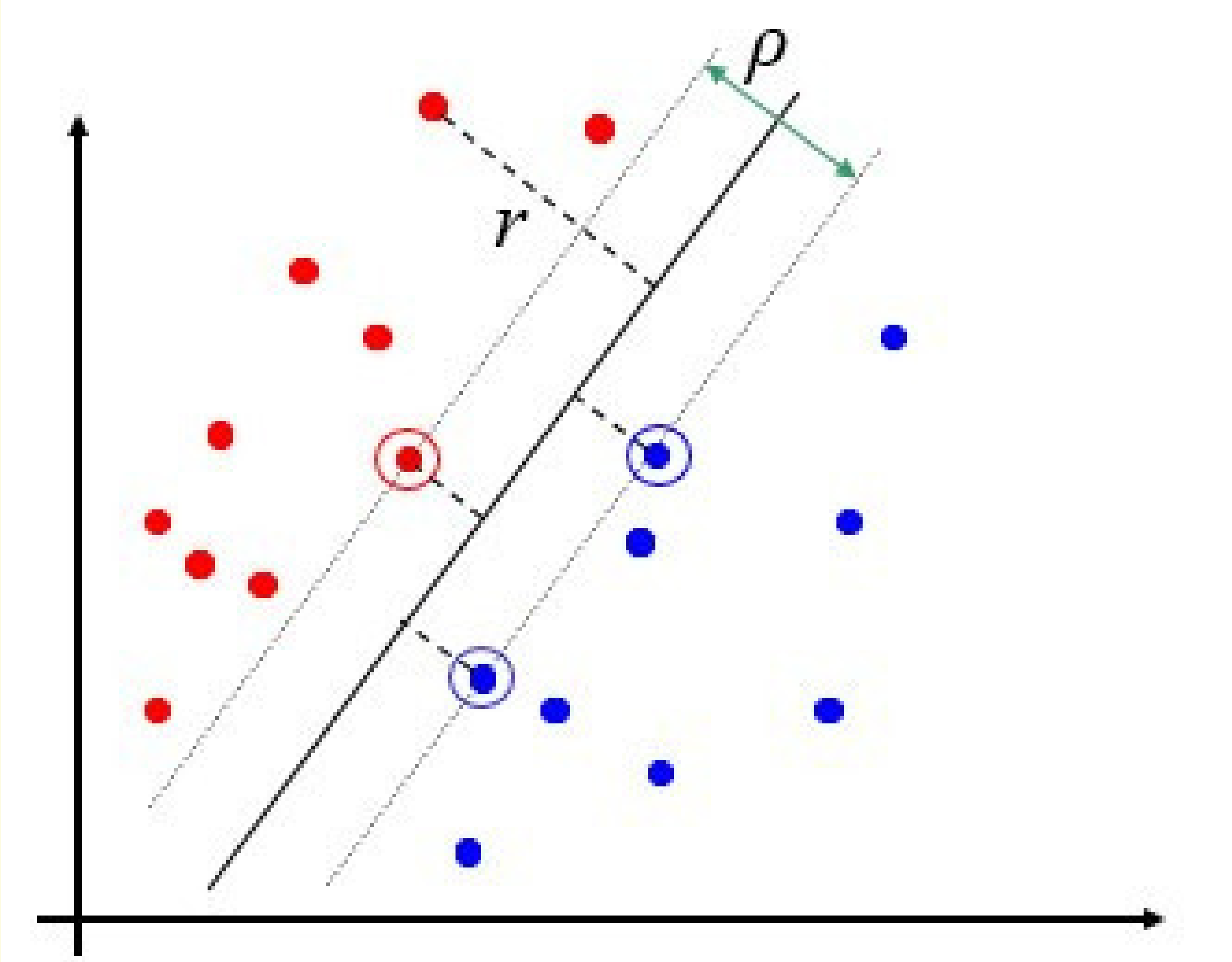
- Banyaknya customer yang berhenti berlangganan
- Pendapatan perusahaan menurun

Tujuan

- Membantu perusahaan mempertahankan pelanggan
- Meningkatkan kualitas pelayanan
- Memprediksi perilaku pelanggan

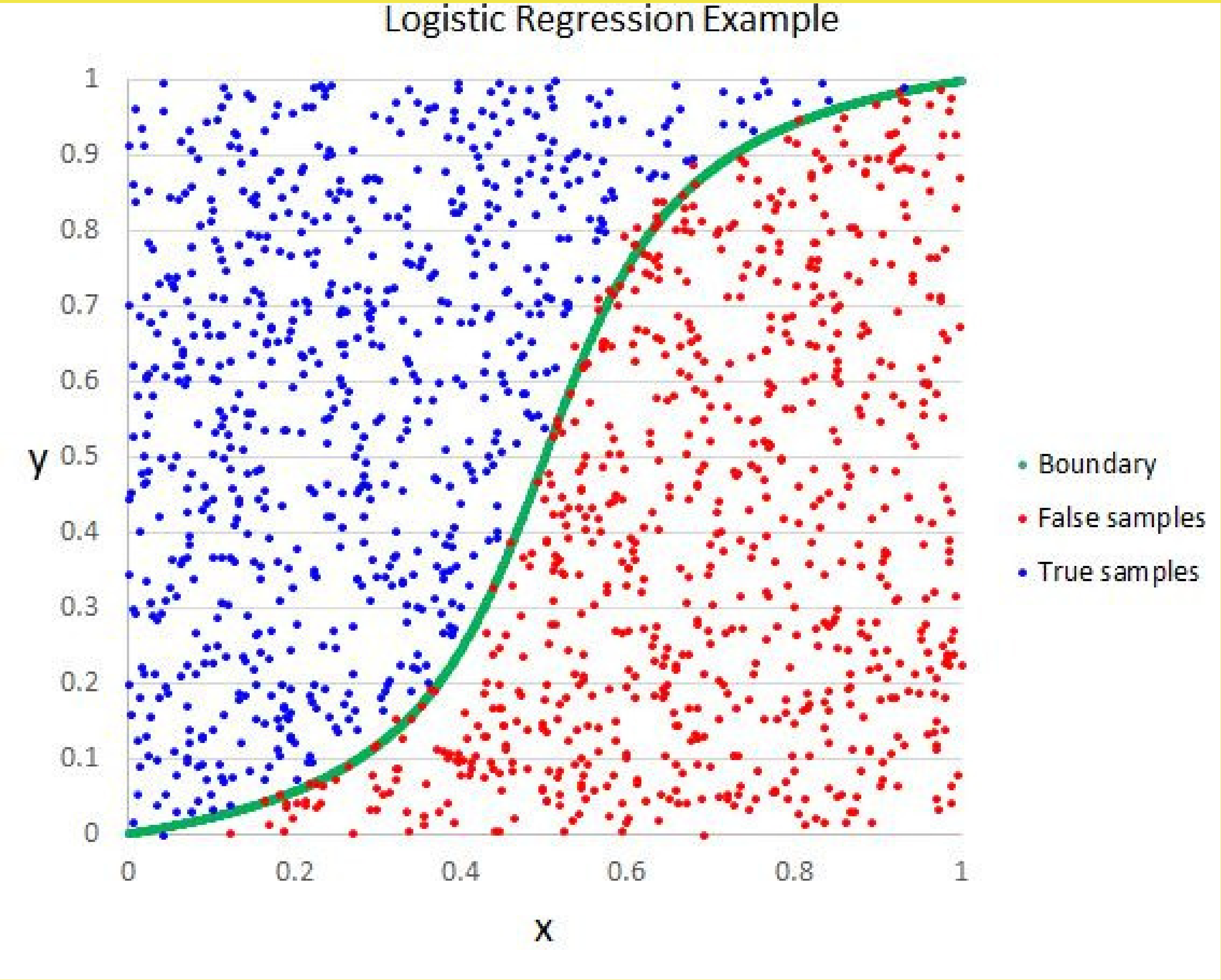
Metode

- **SVM (Support Vector Machine)**



Metode

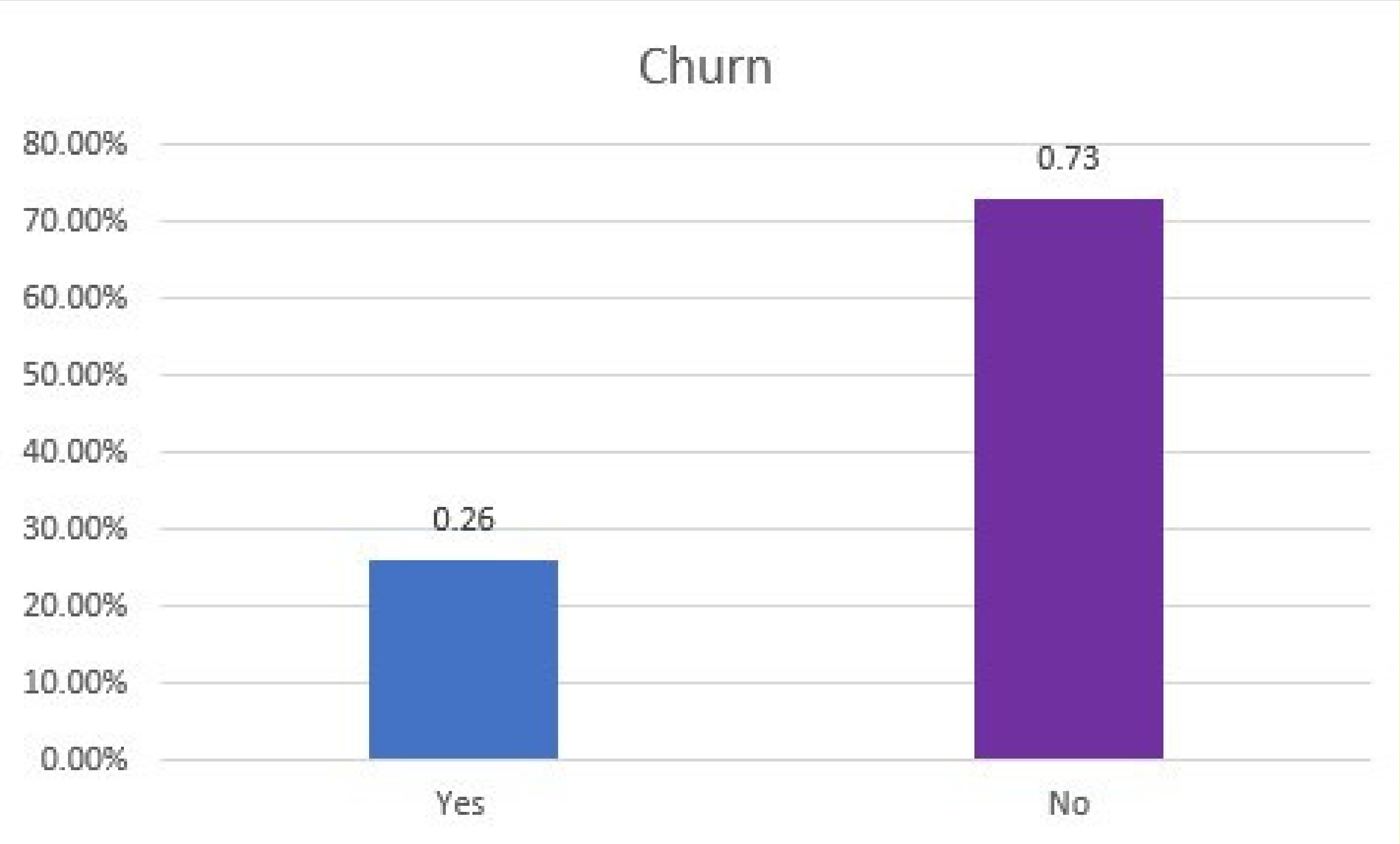
- **Logistic Regression**

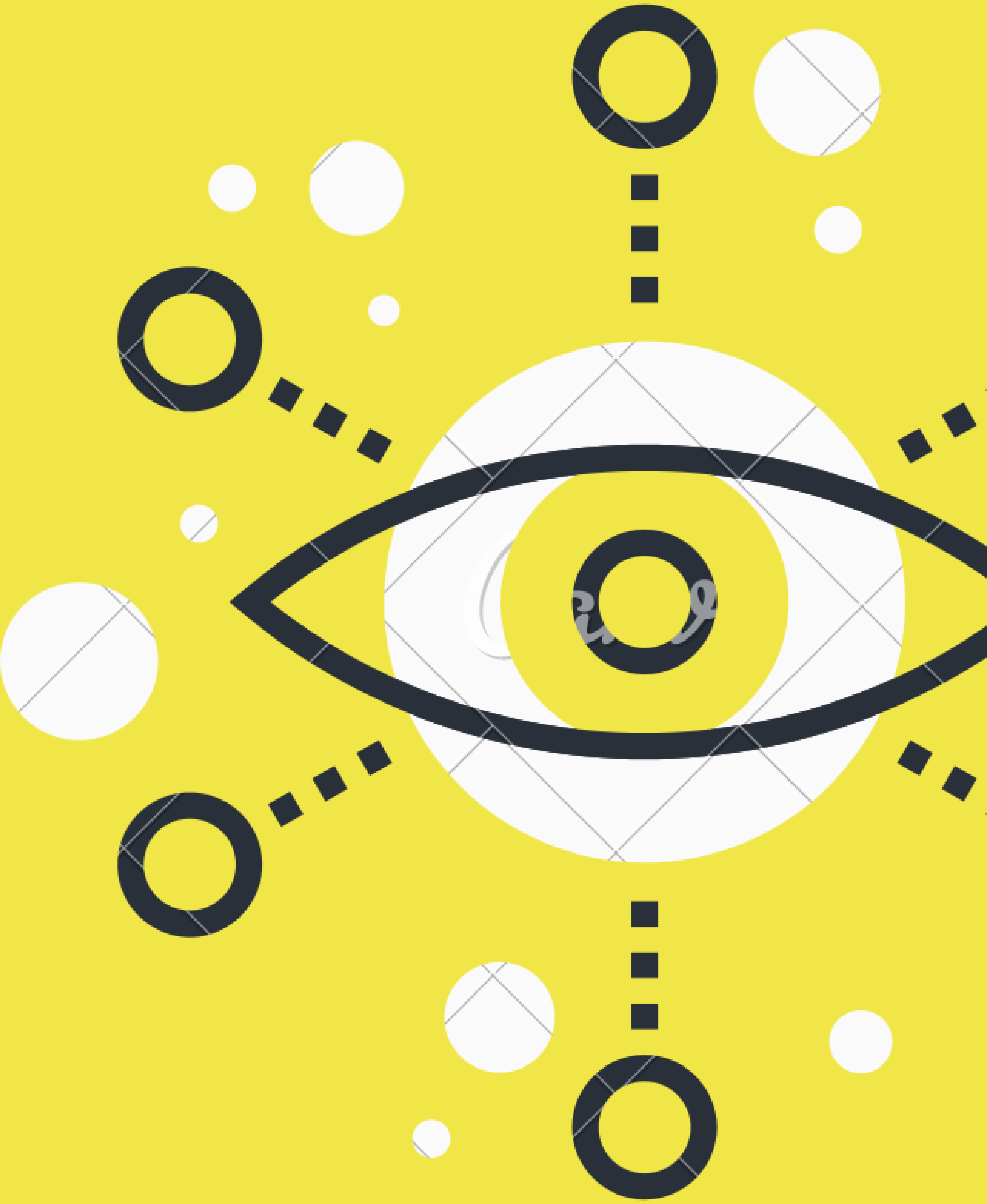


Pembahasan

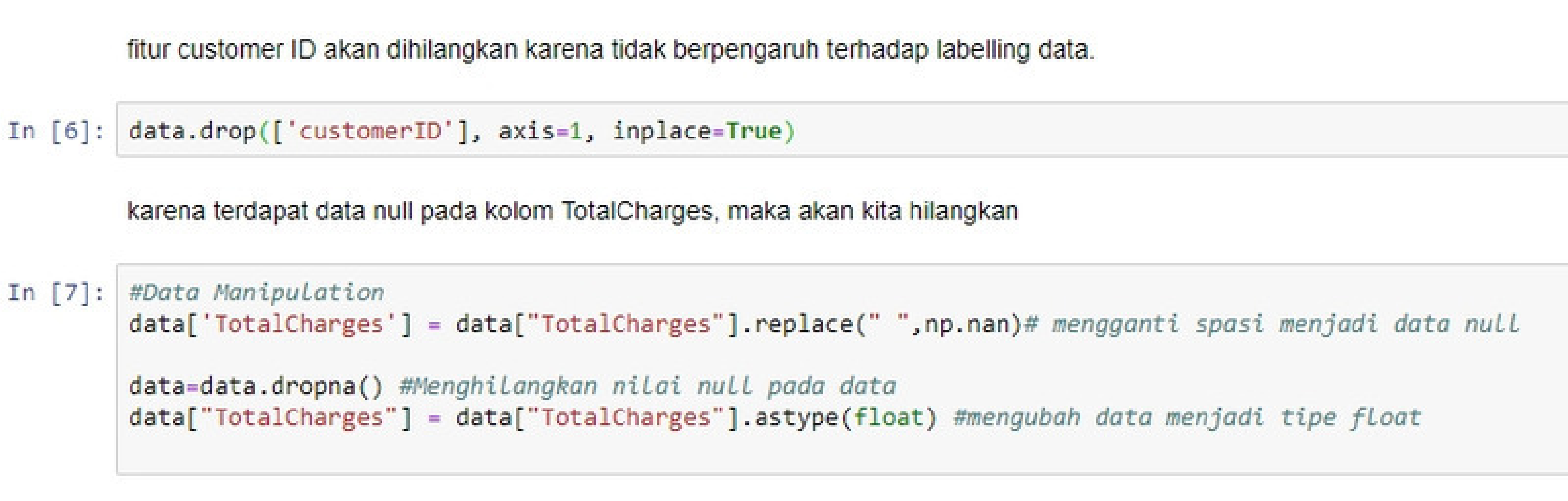
- Data Overview
- Data Cleaning
- Data Prepocessing
- Feature Selection (Chi2)
- Applied Machine Learning Model
- Model Performance

DATA OVERVIEW





DATA CLEANING



fitur customer ID akan dihilangkan karena tidak berpengaruh terhadap labelling data.

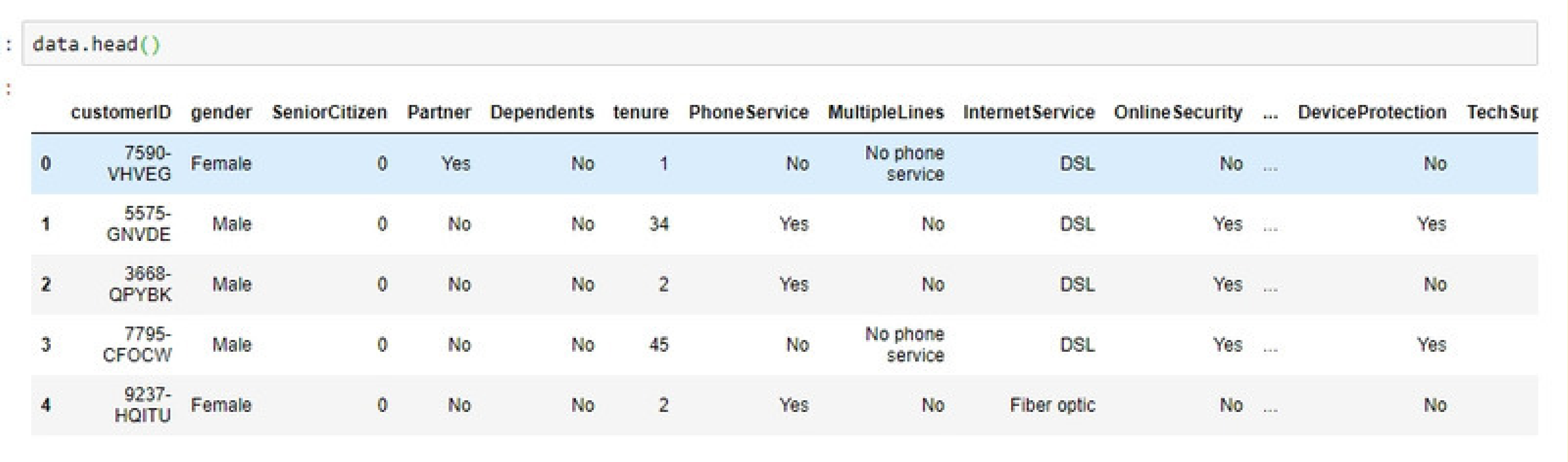
```
In [6]: data.drop(['customerID'], axis=1, inplace=True)
```

karena terdapat data null pada kolom TotalCharges, maka akan kita hilangkan

```
In [7]: #Data Manipulation
        data['TotalCharges'] = data["TotalCharges"].replace(" ",np.nan)# mengganti spasi menjadi data null

        data=data.dropna() #Menghilangkan nilai null pada data
        data["TotalCharges"] = data["TotalCharges"].astype(float) #mengubah data menjadi tipe float
```

DATA PREPOCESSING

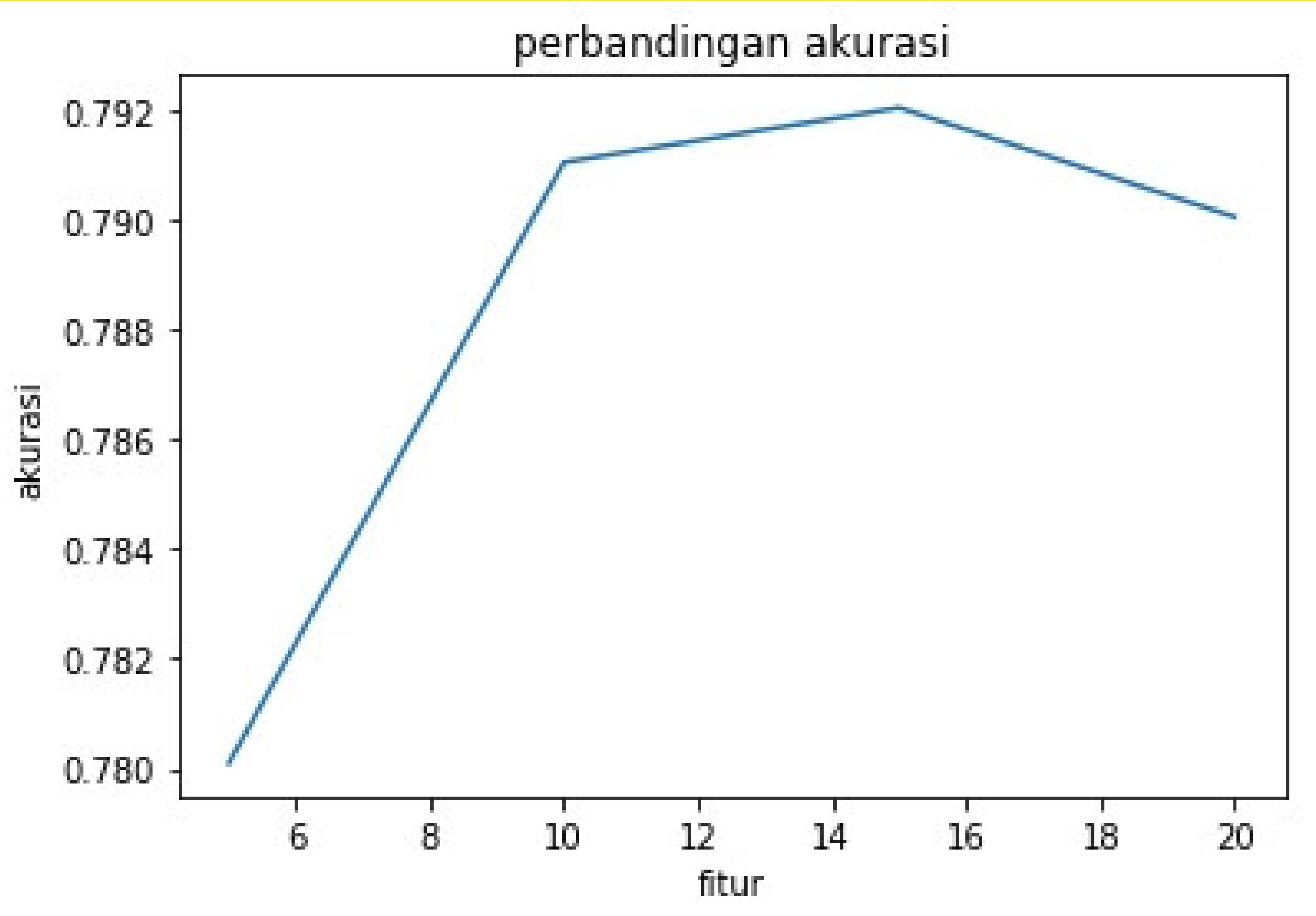


```
data.head()
```

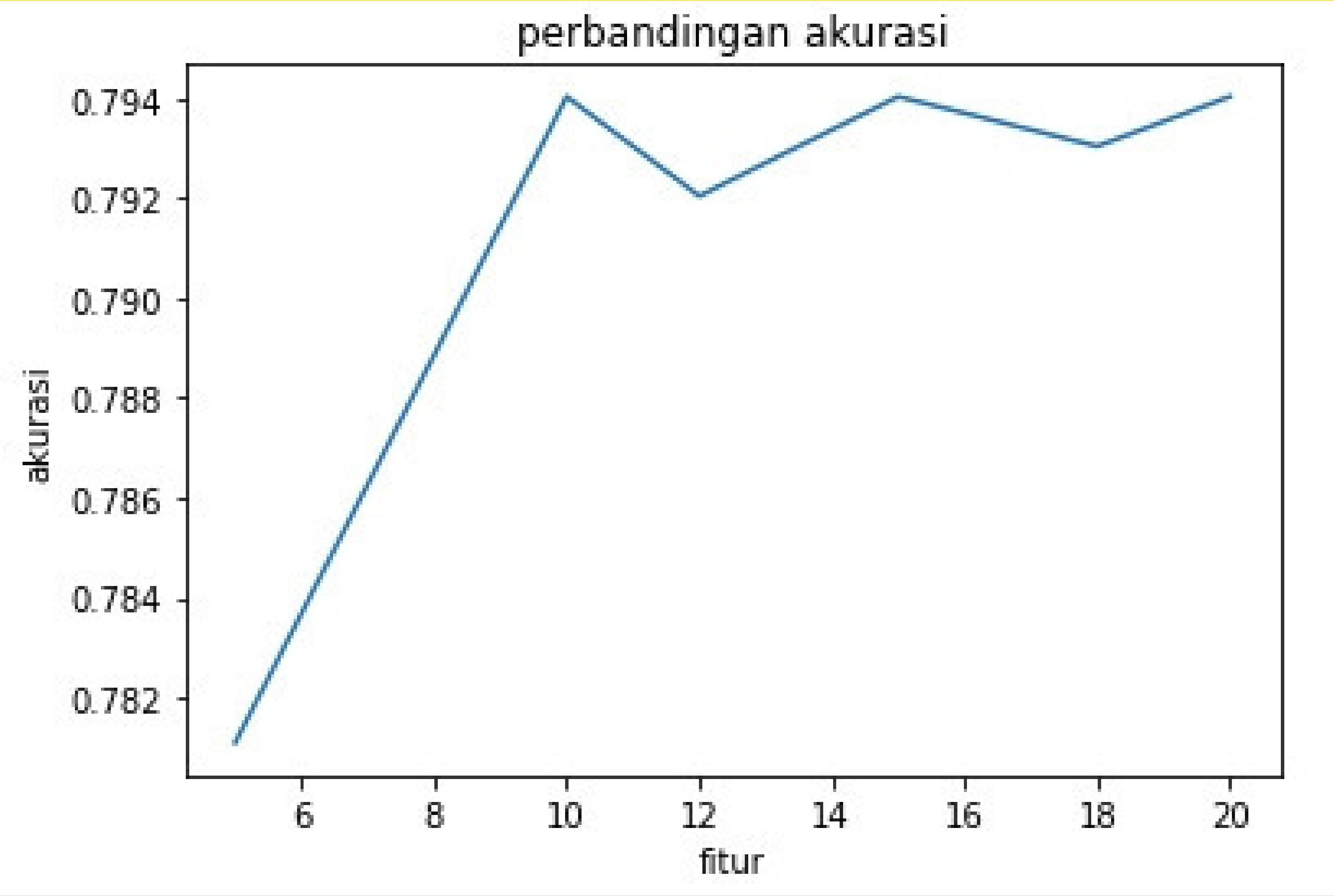
	customerID	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	MultipleLines	InternetService	OnlineSecurity	...	DeviceProtection	TechSup
0	7590-VHVEG	Female	0	Yes	No	1	No	No phone service	DSL	No	...	No	
1	5575-GNVDE	Male	0	No	No	34	Yes	No	DSL	Yes	...	Yes	
2	3668-QPYBK	Male	0	No	No	2	Yes	No	DSL	Yes	...	No	
3	7795-CFOCW	Male	0	No	No	45	No	No phone service	DSL	Yes	...	Yes	
4	9237-HQITU	Female	0	No	No	2	Yes	No	Fiber optic	No	...	No	

FEATURE SELECTION

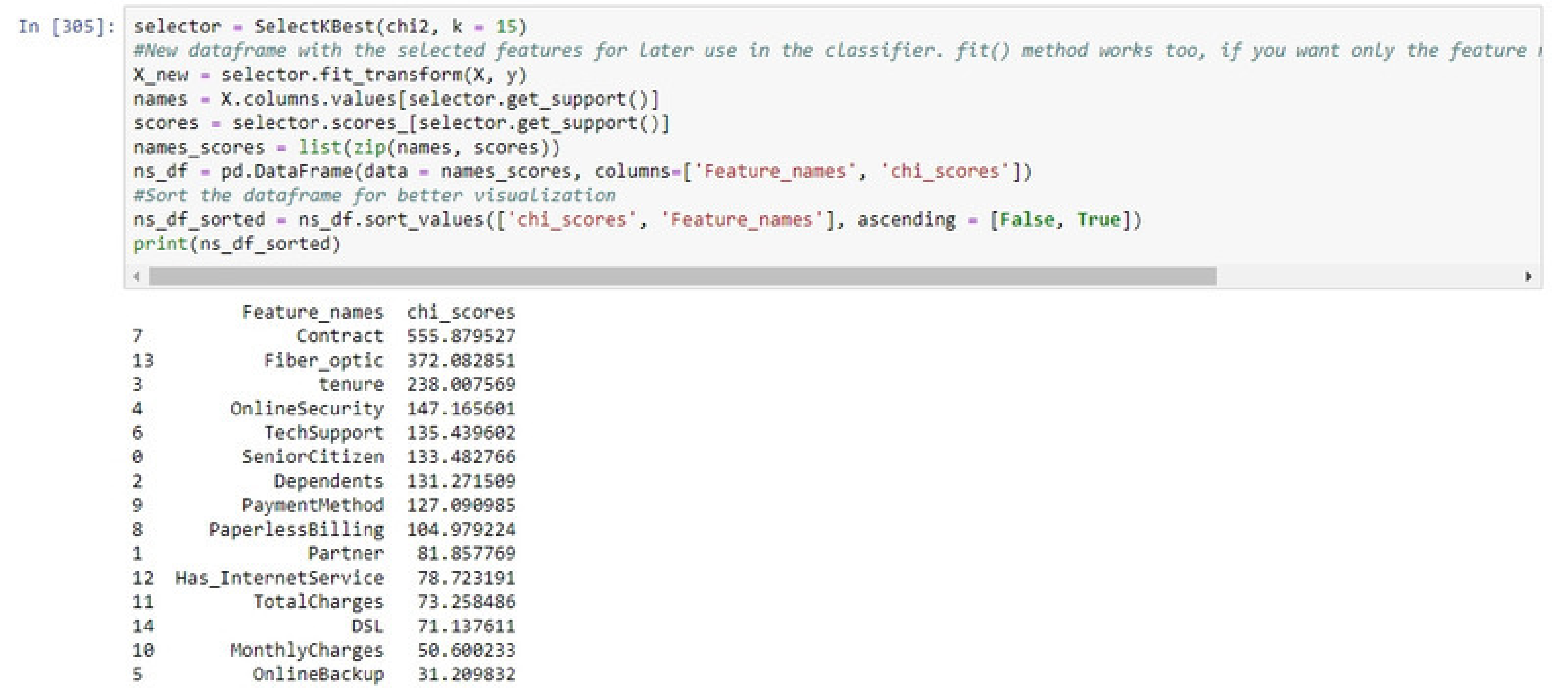
Regression Linier



SVM (Support Vector Machine)



HASIL FEATURE SELECTION

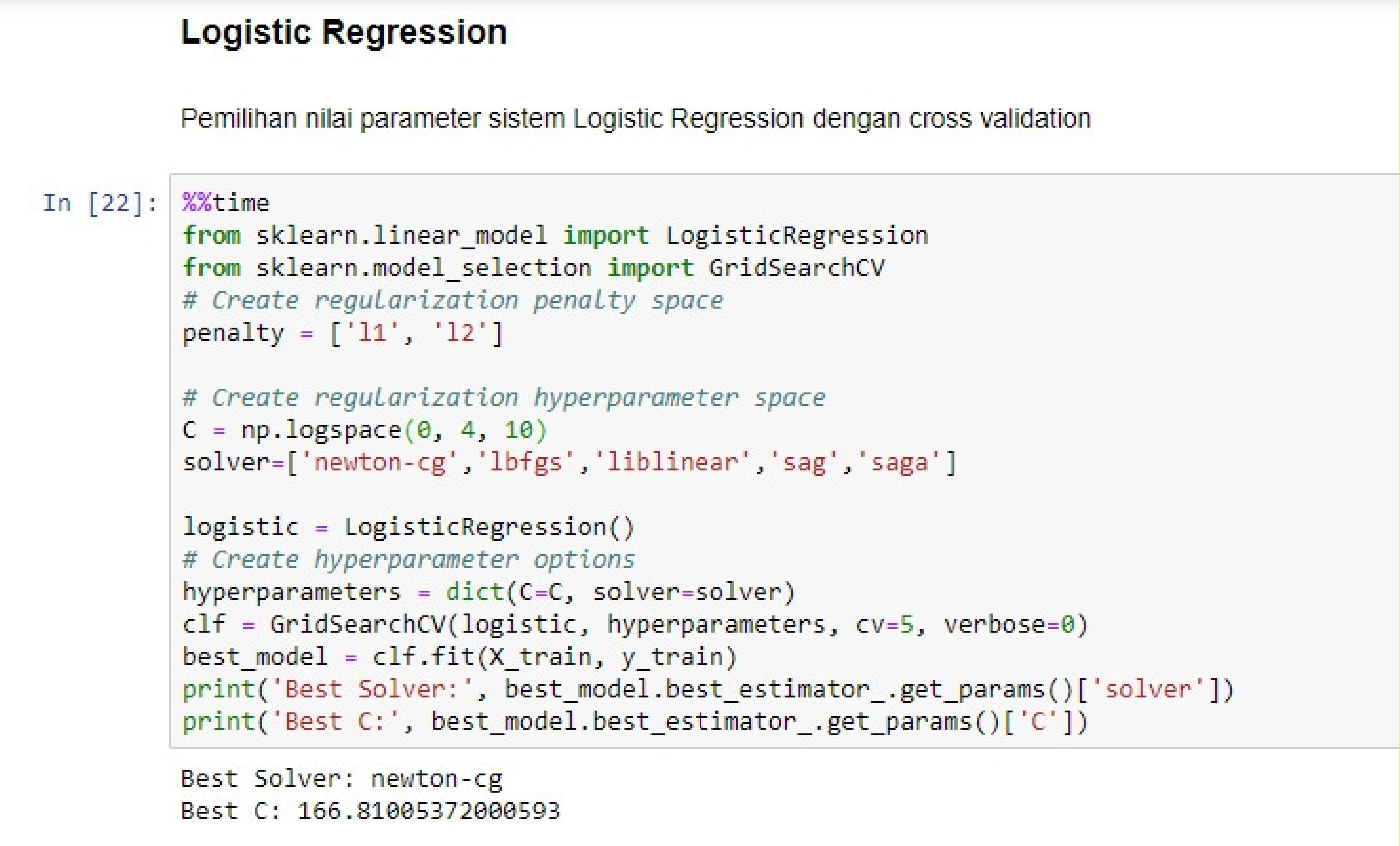


In [305]:

```
selector = SelectKBest(chi2, k = 15)
#New dataframe with the selected features for later use in the classifier. fit() method works too, if you want only the feature
X_new = selector.fit_transform(X, y)
names = X.columns.values[selector.get_support()]
scores = selector.scores_[selector.get_support()]
names_scores = list(zip(names, scores))
ns_df = pd.DataFrame(data = names_scores, columns=['Feature_names', 'chi_scores'])
#Sort the dataframe for better visualization
ns_df_sorted = ns_df.sort_values(['chi_scores', 'Feature_names'], ascending = [False, True])
print(ns_df_sorted)
```

```
         Feature_names  chi_scores
7             Contract  555.879527
13         Fiber_optic  372.082851
3               tenure  238.007569
4       OnlineSecurity  147.165601
6          TechSupport  135.439602
0        SeniorCitizen  133.482766
2           Dependents  131.271509
9        PaymentMethod  127.090985
8     PaperlessBilling  104.979224
1              Partner   81.857769
12 Has_InternetService   78.723191
11        TotalCharges   73.258486
14                 DSL   71.137611
10      MonthlyCharges   50.600233
5         OnlineBackup   31.209832
```

LOGISTIC REGRESSION



Logistic Regression

Pemilihan nilai parameter sistem Logistic Regression dengan cross validation

In [22]:

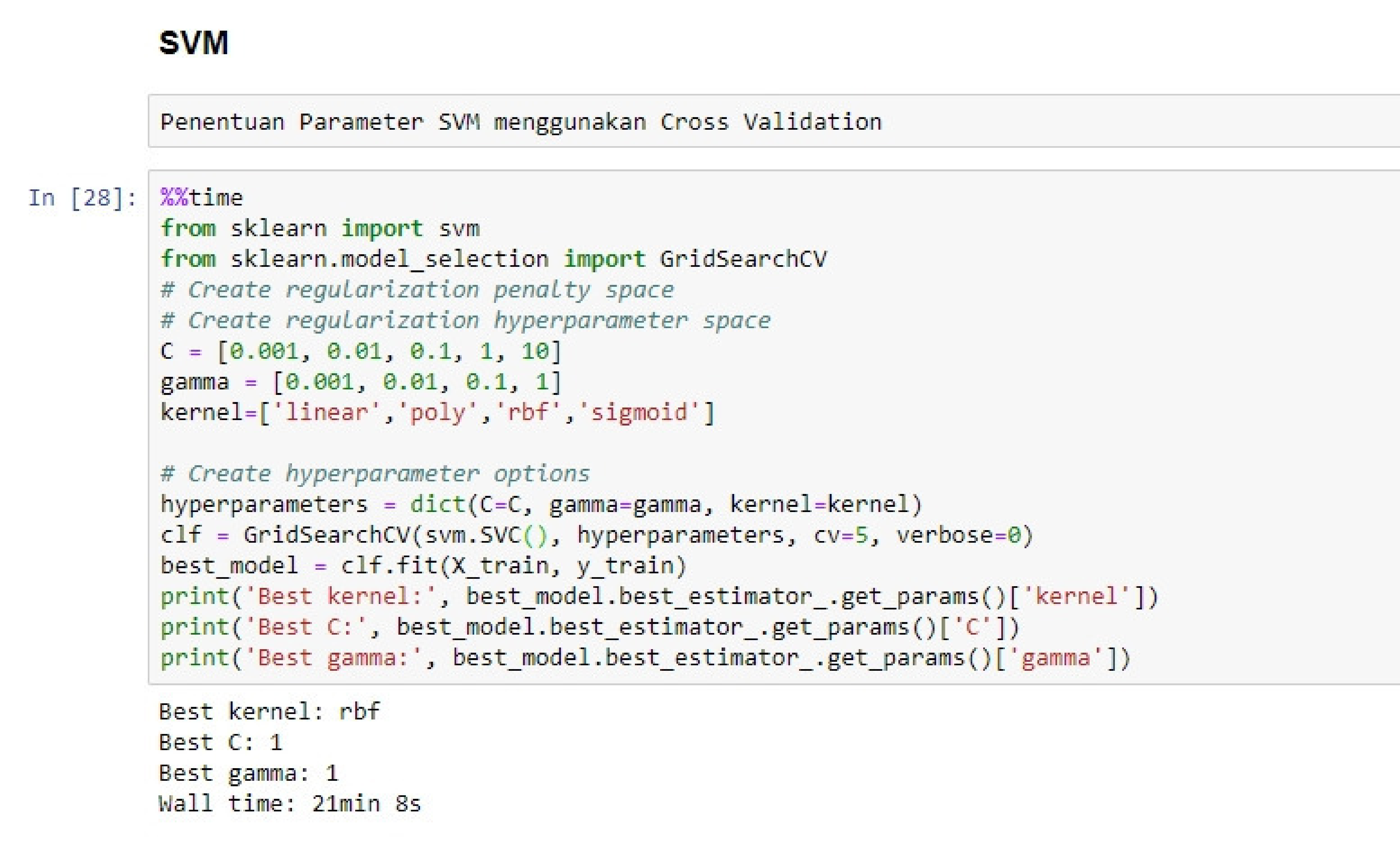
```
%%time
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import GridSearchCV
# Create regularization penalty space
penalty = ['l1', 'l2']

# Create regularization hyperparameter space
C = np.logspace(0, 4, 10)
solver=['newton-cg','lbfgs','liblinear','sag','saga']

logistic = LogisticRegression()
# Create hyperparameter options
hyperparameters = dict(C=C, solver=solver)
clf = GridSearchCV(logistic, hyperparameters, cv=5, verbose=0)
best_model = clf.fit(X_train, y_train)
print('Best Solver:', best_model.best_estimator_.get_params()['solver'])
print('Best C:', best_model.best_estimator_.get_params()['C'])
```

```
Best Solver: newton-cg
Best C: 166.81005372000593
```

SVM (SUPPORT VECTOR MACHINE)



SVM

Penentuan Parameter SVM menggunakan Cross Validation

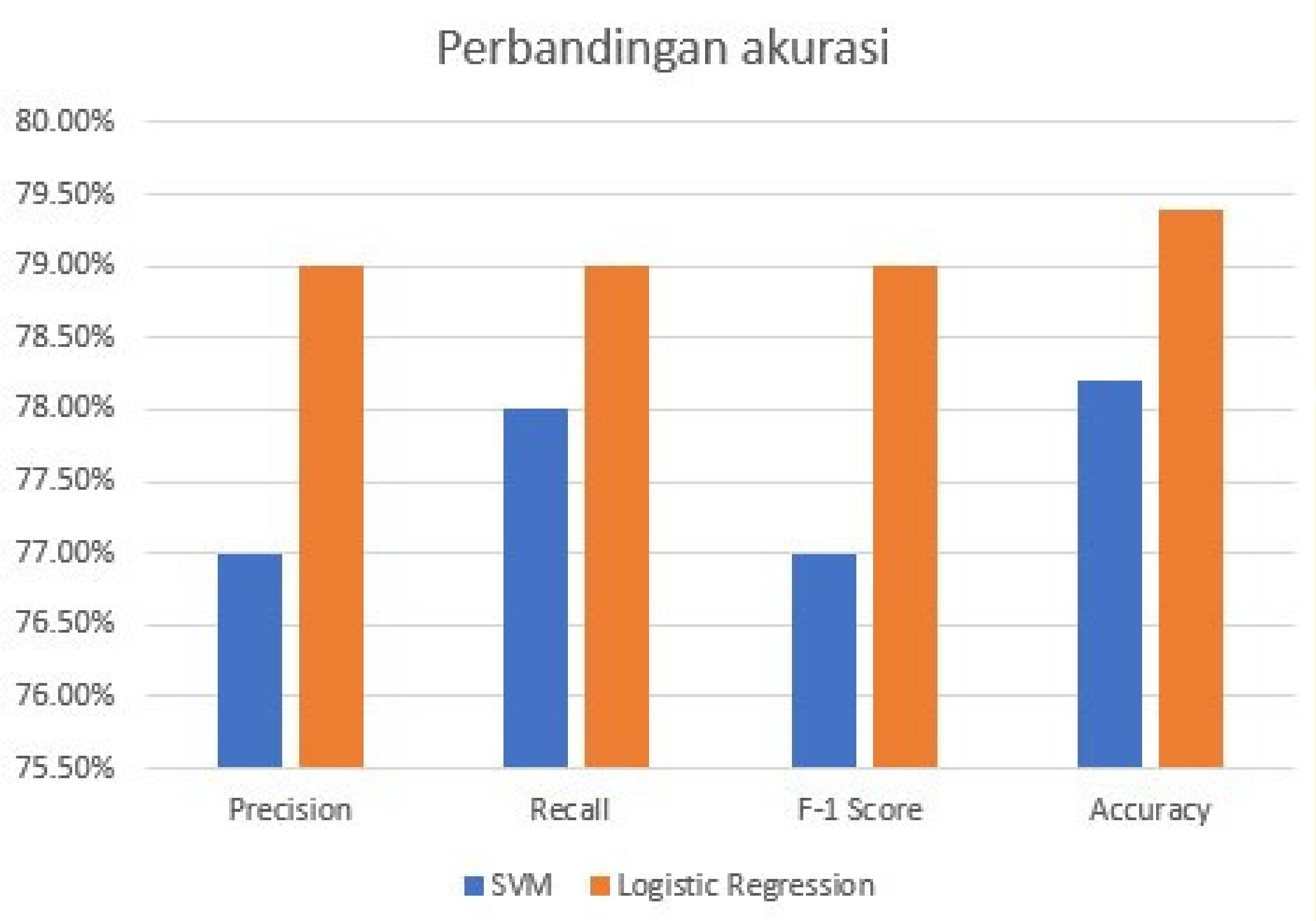
In [28]:

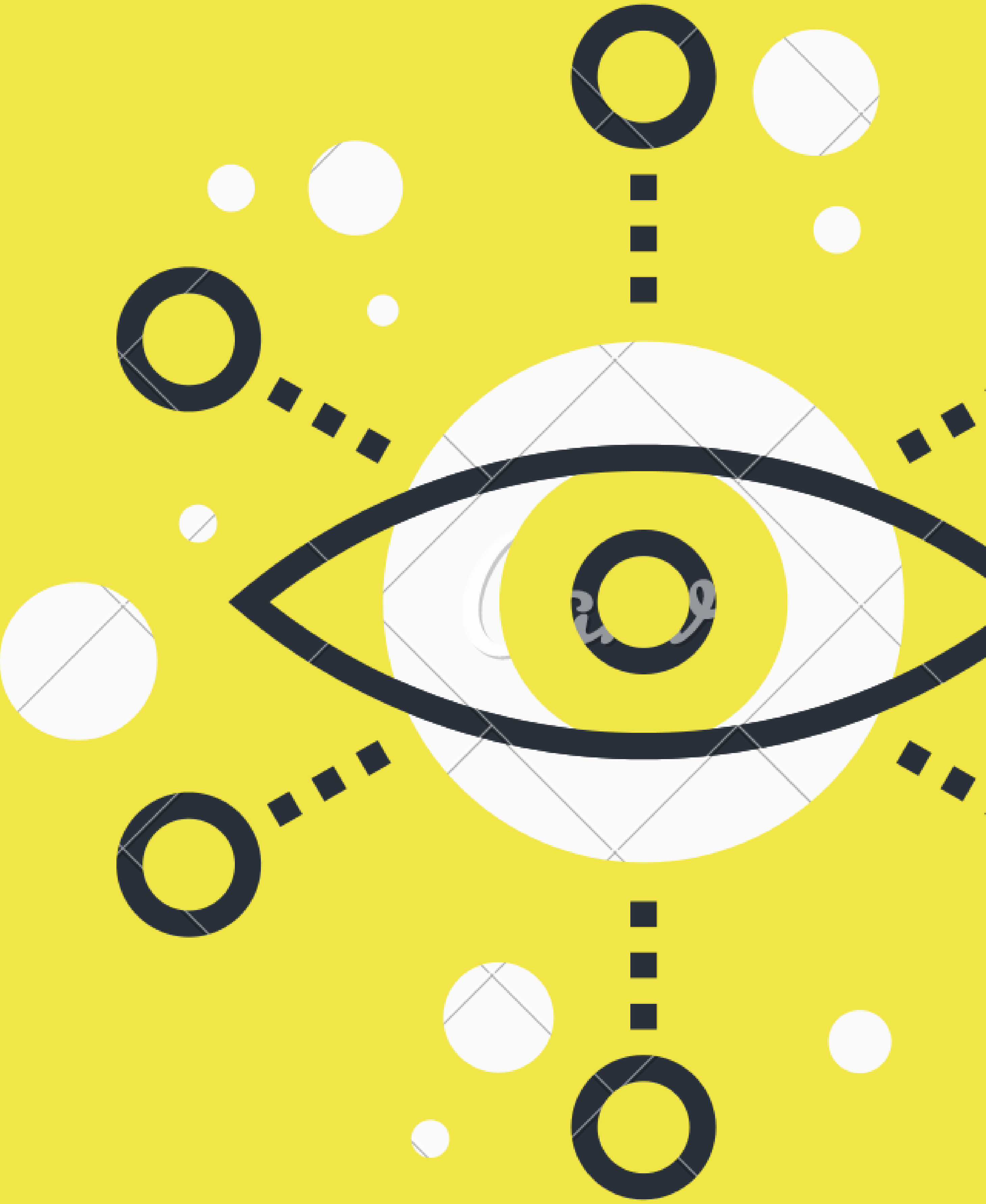
```
%%time
from sklearn import svm
from sklearn.model_selection import GridSearchCV
# Create regularization penalty space
# Create regularization hyperparameter space
C = [0.001, 0.01, 0.1, 1, 10]
gamma = [0.001, 0.01, 0.1, 1]
kernel=['linear','poly','rbf','sigmoid']

# Create hyperparameter options
hyperparameters = dict(C=C, gamma=gamma, kernel=kernel)
clf = GridSearchCV(svm.SVC(), hyperparameters, cv=5, verbose=0)
best_model = clf.fit(X_train, y_train)
print('Best kernel:', best_model.best_estimator_.get_params()['kernel'])
print('Best C:', best_model.best_estimator_.get_params()['C'])
print('Best gamma:', best_model.best_estimator_.get_params()['gamma'])
```

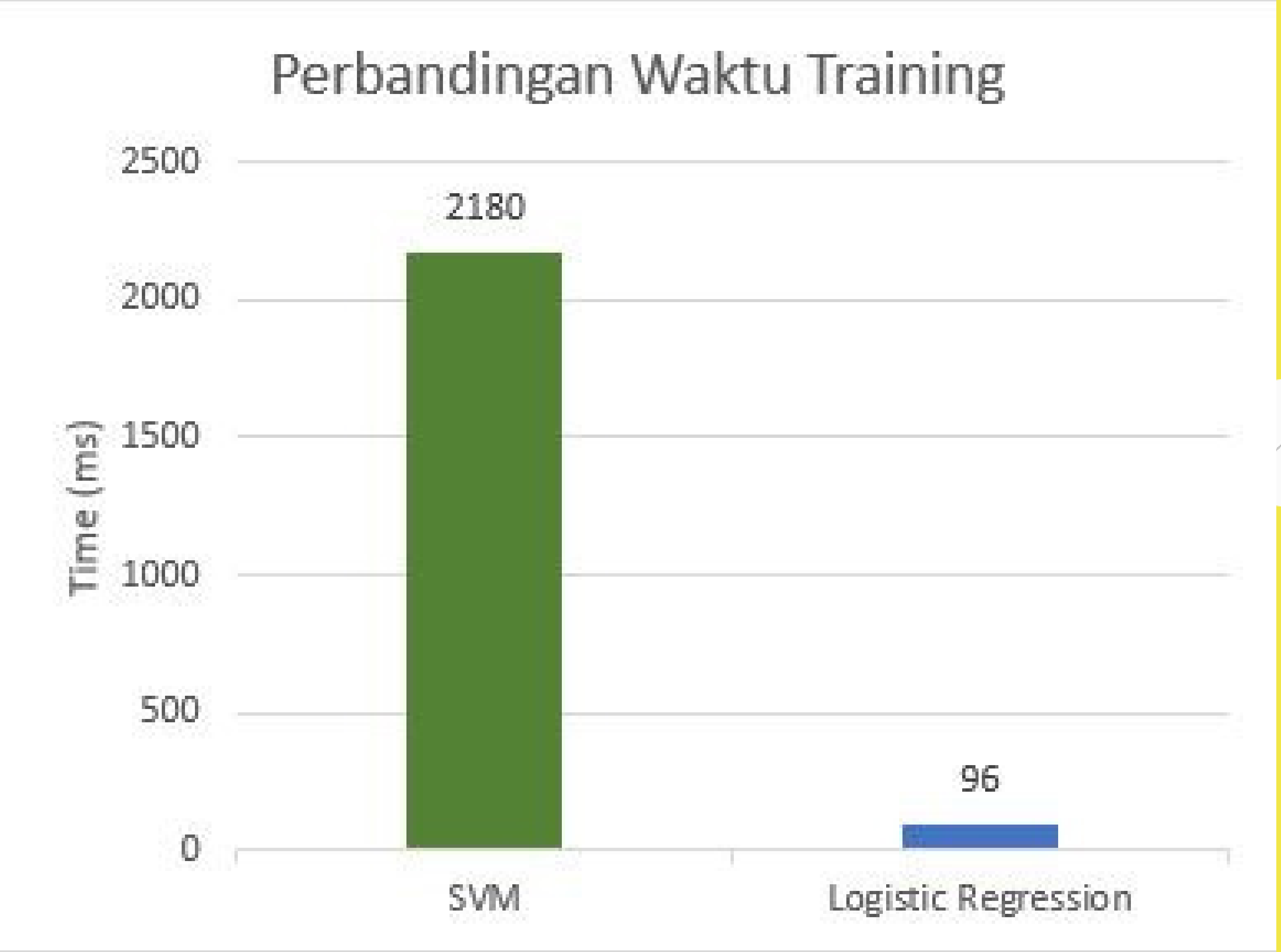
```
Best kernel: rbf
Best C: 1
Best gamma: 1
Wall time: 21min 8s
```

Analisis


Perbandingan akurasi
80.00%
79.50%
79.00%
78.50%
78.00%
77.50%
77.00%
76.50%
76.00%
75.50%
Precision
Recall
F-1 Score
Accuracy
SVM
Logistic Regression



Analisis


Perbandingan Waktu Training
Time (ms)
2500
2000
1500
1000
500
0
2180
96
SVM
Logistic Regression

Kesimpulan

1

LOGISTIC REGRESSION LEBIH BAIK

2

DIBUTUHKAN LEBIH BANYAK DATA SUPAYA AKURASI MEMBAIK

Terima Kasih