

Success Metrics : FindDefault-Project

1 # Success Metrics : Hyperparameter Tuning

```
In [5]: 1 import pandas as pd
2 from sklearn.model_selection import train_test_split, GridSearchCV
3 from sklearn.linear_model import LogisticRegression
4
5 # Load the CSV file into a DataFrame
6 data = pd.read_csv("creditcard.csv")
7
8 # Split the data into features (X) and target variable (y)
9 X = data.drop(columns=['Class'])
10 y = data['Class']
11
12 # Split the data into training and testing sets
13 X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
14
15 # Define the Logistic regression model with increased max_iter
16 model = LogisticRegression(max_iter=100000) # Increase max_iter from default value
17
18 # Fit the model to the training data
19 model.fit(X_train, y_train)
20
21 # Evaluate the model
22 accuracy = model.score(X_test, y_test)
23 print("Accuracy:", accuracy)
24
```

Accuracy: 0.9991222218320986

1 # Model Evaluation

```
In [11]: 1 import pandas as pd
2 from sklearn.metrics import accuracy_score, classification_report
3
4 # Load the CSV file into a DataFrame
5 data = pd.read_csv("creditcard.csv")
6
7 # Split the data into features (X) and target variable (y)
8 X = data.drop(columns=['Class'])
9 y = data['Class']
10
11 # Predict on the test set
12 y_pred = best_model.predict(X_test)
13
14 # Calculate accuracy
15 accuracy = accuracy_score(y_test, y_pred)
16
17 # Perform model validation
18 print("Classification Report:")
19 print(classification_report(y_test, y_pred))
20
21 # Print the accuracy
22 print("Accuracy:", accuracy)
23
```

Classification Report:

	precision	recall	f1-score	support
0	1.00	1.00	1.00	56864
1	0.61	0.56	0.59	98
accuracy			1.00	56962
macro avg	0.81	0.78	0.79	56962
weighted avg	1.00	1.00	1.00	56962

Accuracy: 0.9986306660580738

