Double-click (or enter) to edit

```
import pandas as pd
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import train_test_split
from sklearn.metrics import classification_report, confusion_matrix

df=pd.read_csv('/kaggle/input/creditcardfraud/creditcard.csv')
print(df['Class'].value_counts())

X=df.drop('Class',axis=1)
y=df['Class']

X_train,X_test,y_train,y_test=train_test_split(X,y,test_size=0.3,random_state=42,stratify=y)

model=RandomForestClassifier(n_estimators=100,random_state=42)
model.fit(X_train,y_train)

y_pred=model.predict(X_test)

print(confusion_matrix(y_test,y_pred))
print(classification_report(y_test,y_pred))

feat_importances=pd.Series(model.feature_importances_,index=X.columns)
feat_importances.nlargest(10).plot(kind='barh')
plt.title("Top 10 important Features")
import pandas as pd
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import train_test_split
from sklearn.metrics import classification_report, confusion_matrix

df=pd.read_csv('/kaggle/input/creditcardfraud/creditcard.csv')
print(df['Class'].value_counts())

X=df.drop('Class',axis=1)
y=df['Class']

X_train,X_test,y_train,y_test=train_test_split(X,y,test_size=0.3,random_state=42,stratify=y)

model=RandomForestClassifier(n_estimators=100,random_state=42)
model.fit(X_train,y_train)

y_pred=model.predict(X_test)

print(confusion_matrix(y_test,y_pred))
print(classification_report(y_test,y_pred))

feat_importances=pd.Series(model.feature_importances_,index=X.columns)
feat_importances.nlargest(10).plot(kind='barh')
plt.title("Top 10 important Features")
plt.show()

from sklearn.metrics import classification_report, confusion_matrix

df=pd.read_csv('/kaggle/input/creditcardfraud/creditcard.csv')
print(df['Class'].value_counts())

X=df.drop('Class',axis=1)
y=df['Class']

X_train,X_test,y_train,y_test=train_test_split(X,y,test_size=0.3,random_state=42,stratify=y)

model=RandomForestClassifier(n_estimators=100,random_state=42)
model.fit(X_train,y_train)

y_pred=model.predict(X_test)

print(confusion_matrix(y_test,y_pred))
print(classification_report(y_test,y_pred))

feat_importances=pd.Series(model.feature_importances_,index=X.columns)
feat_importances.nlargest(10).plot(kind='barh')
plt.title("Top 10 important Features")
plt.show()
import pandas as pd
```

```
import pandas as pd
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import train_test_split
from sklearn.metrics import classification_report, confusion_matrix

df=pd.read_csv("/kaggle/input/creditcardfraud/creditcard.csv")
print(df['Class'].value_counts())

X=df.drop('Class',axis=1)
y=df['Class']

X_train,X_test,y_train,y_test=train_test_split(X,y,test_size=0.3,random_state=42,stratify=y)

model=RandomForestClassifier(n_estimators=100,random_state=42)
model.fit(X_train,y_train)

y_pred=model.predict(X_test)

print(confusion_matrix(y_test,y_pred))
print(classification_report(y_test,y_pred))

feat_importances=pd.Series(model.feature_importances_,index=X.columns)
feat_importances.nlargest(10).plot(kind='barh')
plt.title("Top 10 important Features")
import pandas as pd
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import train_test_split
from sklearn.metrics import classification_report, confusion_matrix

df=pd.read_csv('/kaggle/input/creditcardfraud/creditcard.csv')
print(df['Class'].value_counts())

X=df.drop('Class',axis=1)
y=df['Class']

X_train,X_test,y_train,y_test=train_test_split(X,y,test_size=0.3,random_state=42,stratify=y)

model=RandomForestClassifier(n_estimators=100,random_state=42)
model.fit(X_train,y_train)

y_pred=model.predict(X_test)

print(confusion_matrix(y_test,y_pred))
print(classification_report(y_test,y_pred))

feat_importances=pd.Series(model.feature_importances_,index=X.columns)
feat_importances.nlargest(10).plot(kind='barh')
plt.title("Top 10 important Features")
plt.show()

from sklearn.metrics import classification_report, confusion_matrix

df=pd.read_csv('/kaggle/input/creditcardfraud/creditcard.csv')
print(df['Class'].value_counts())

X=df.drop('Class',axis=1)
y=df['Class']

X_train,X_test,y_train,y_test=train_test_split(X,y,test_size=0.3,random_state=42,stratify=y)

model=RandomForestClassifier(n_estimators=100,random_state=42)
model.fit(X_train,y_train)

y_pred=model.predict(X_test)

print(confusion_matrix(y_test,y_pred))
print(classification_report(y_test,y_pred))

feat_importances=pd.Series(model.feature_importances_,index=X.columns)
feat_importances.nlargest(10).plot(kind='barh')
plt.title("Top 10 important Features")
plt.show()


df=pd.read_csv('/kaggle/input/creditcardfraud/creditcard.csv')
print(df['Class'].value_counts())
```

```
X=df.drop('Class',axis=1)
y=df['Class']

X_train,X_test,y_train,y_test=train_test_split(X,y,test_size=0.3,random_state=42,stratify=y)

model=RandomForestClassifier(n_estimators=100,random_state=42)
model.fit(X_train,y_train)

y_pred=model.predict(X_test)

print(confusion_matrix(y_test,y_pred))
print(classification_report(y_test,y_pred))

feat_importances=pd.Series(model.feature_importances_,index=X.columns)
feat_importances.nlargest(10).plot(kind='barh')
plt.title("Top 10 important Features")
plt.show()
X=df.drop('Class',axis=1)
y=df['Class']

X_train,X_test,y_train,y_test=train_test_split(X,y,test_size=0.3,random_state=42,syratify=y)

model=RandomForestClassifier(n_estimators=100,random_state=42)
model.fit(X_train,y_train)

y_pred=model.predict(X_test)

print(confusion_matrix(y_test,y_pred))
print(classification_report(y_test,y_pred))

feat_importances=pd.Series(model.feature_importances_,index=X.columns)
feat_importances.nlargest(10).plot(kind='barh')
plt.title("Top 10 important Features")
plt.show()
```

```
Class
0    284315
1       492
Name: count, dtype: int64
[[85290     5]
 [   36   112]]
              precision    recall  f1-score   support

           0       1.00      1.00      1.00     85295
           1       0.96      0.76      0.85       148

    accuracy                           1.00     85443
   macro avg       0.98      0.88      0.92     85443
weighted avg       1.00      1.00      1.00     85443

Class
0    284315
1       492
Name: count, dtype: int64
[[85290     5]
 [   36   112]]
              precision    recall  f1-score   support

           0       1.00      1.00      1.00     85295
           1       0.96      0.76      0.85       148

    accuracy                           1.00     85443
   macro avg       0.98      0.88      0.92     85443
weighted avg       1.00      1.00      1.00     85443
```