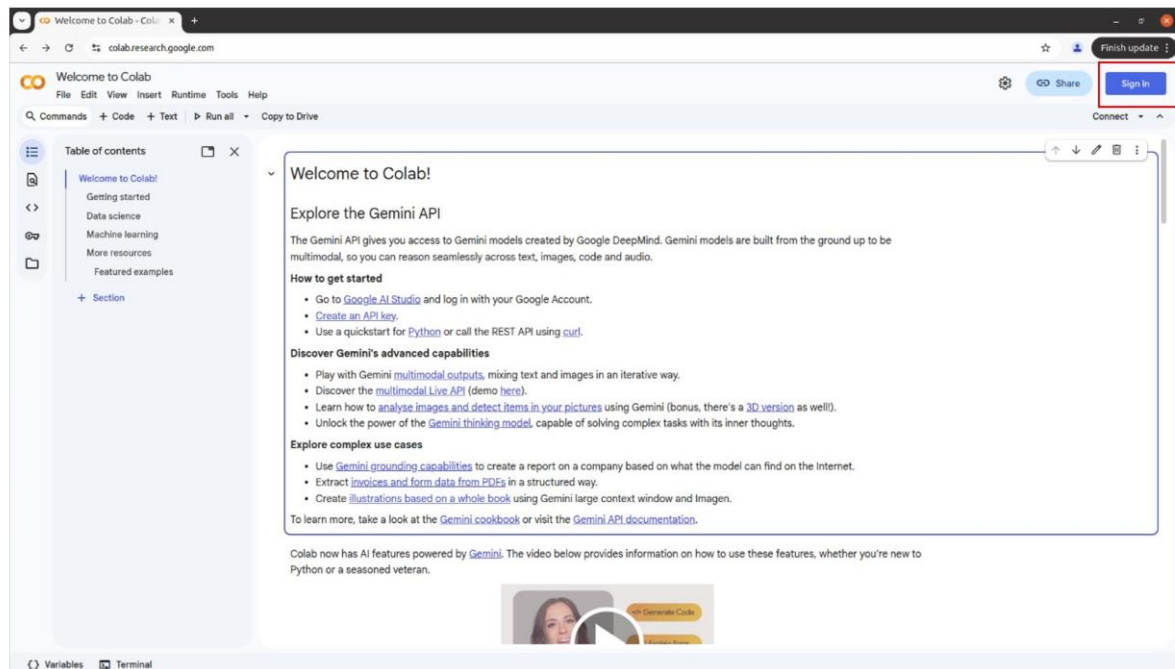


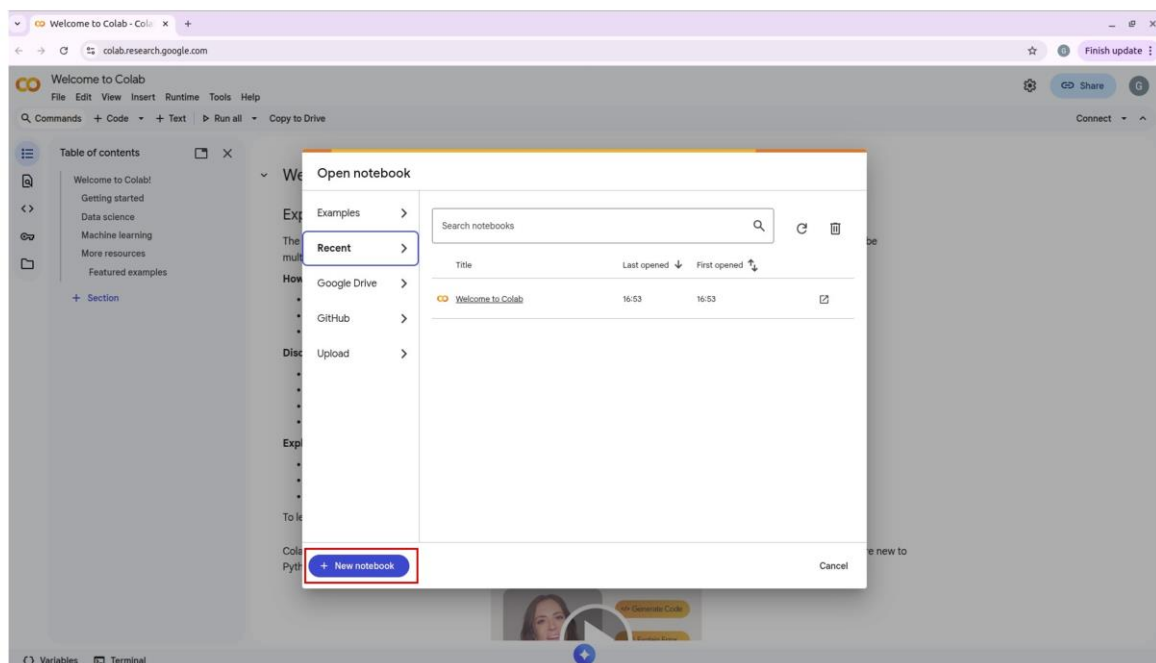
Open google colab using following link

<https://colab.research.google.com/>

-sign in via email id (on top right corner)



-click on New Notebook



import pandas as pd

Create data

```
data = {  
    'CellLine': ['CL1', 'CL2', 'CL3', 'CL4', 'CL5', 'CL6'],  
    'GeneA_Expression': [8.1, 6.5, 9.0, 7.2, None, 5.8],  
    'GeneB_Expression': [4.3, 3.0, 5.1, 2.8, 4.5, None],  
    'Drug_Response': ['Sensitive', 'Resistant', 'Sensitive', 'Resistant', 'Sensitive', 'Resistant']  
}  
df = pd.DataFrame(data)
```

Fill missing values with mean

```
df = df.fillna(df.mean(numeric_only=True))  
  
print(df)
```

Convert sensitive and resistant to binary digits

```
df['Drug_Response'] = df['Drug_Response'].map({'Sensitive': 1, 'Resistant': 0})  
  
X = df[['GeneA_Expression', 'GeneB_Expression']]  
y = df['Drug_Response']
```

Separate Data into test and training

```
from sklearn.model_selection import train_test_split  
  
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)  
  
from sklearn.linear_model import LogisticRegression  
  
model = LogisticRegression()  
model.fit(X_train, y_train)
```

```
y_pred = model.predict(X_test)
```

```
from sklearn.metrics import accuracy_score, confusion_matrix, ConfusionMatrixDisplay  
import matplotlib.pyplot as plt
```

```
print("Accuracy:", accuracy_score(y_test, y_pred))
```

Create confusion matrix

```
cm = confusion_matrix(y_test, y_pred)  
disp = ConfusionMatrixDisplay(confusion_matrix=cm, display_labels=["Resistant",  
"Sensitive"])  
disp.plot(cmap="Blues")  
plt.title("Confusion Matrix - Drug Sensitivity Prediction")  
plt.show()
```

Plot Drug Sensitivity Based on Gene Expression

```
import seaborn as sns
```

```
sns.scatterplot(data=df, x='GeneA_Expression', y='GeneB_Expression',  
                hue='Drug_Response', palette='coolwarm', s=100)  
plt.title("Drug Sensitivity Based on Gene Expression")  
plt.xlabel("GeneA Expression")  
plt.ylabel("GeneB Expression")  
plt.show()
```