

```
!pip install pandas numpy matplotlib seaborn scikit-learn
```

Requirement already satisfied: pandas in /usr/local/lib/python3.11/dist-packages (2.2.2)
Requirement already satisfied: numpy in /usr/local/lib/python3.11/dist-packages (2.0.2)
Requirement already satisfied: matplotlib in /usr/local/lib/python3.11/dist-packages (3.10.0)
Requirement already satisfied: seaborn in /usr/local/lib/python3.11/dist-packages (0.13.2)
Requirement already satisfied: scikit-learn in /usr/local/lib/python3.11/dist-packages (1.6.1)
Requirement already satisfied: python-dateutil>=2.8.2 in /usr/local/lib/python3.11/dist-packages (from pandas) (2.9.0.post0)
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.11/dist-packages (from pandas) (2025.2)
Requirement already satisfied: tzdata>=2022.7 in /usr/local/lib/python3.11/dist-packages (from pandas) (2025.2)
Requirement already satisfied: contourpy>=1.0.1 in /usr/local/lib/python3.11/dist-packages (from matplotlib) (1.3.2)
Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.11/dist-packages (from matplotlib) (0.12.1)
Requirement already satisfied: fonttools>=4.22.0 in /usr/local/lib/python3.11/dist-packages (from matplotlib) (4.58.0)
Requirement already satisfied: kiwisolver>=1.3.1 in /usr/local/lib/python3.11/dist-packages (from matplotlib) (1.4.8)
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.11/dist-packages (from matplotlib) (24.2)
Requirement already satisfied: pillow>=8 in /usr/local/lib/python3.11/dist-packages (from matplotlib) (11.2.1)
Requirement already satisfied: pyparsing>=2.3.1 in /usr/local/lib/python3.11/dist-packages (from matplotlib) (3.2.3)
Requirement already satisfied: scipy>=1.6.0 in /usr/local/lib/python3.11/dist-packages (from scikit-learn) (1.15.3)
Requirement already satisfied: joblib>=1.2.0 in /usr/local/lib/python3.11/dist-packages (from scikit-learn) (1.5.0)
Requirement already satisfied: threadpoolctl>=3.1.0 in /usr/local/lib/python3.11/dist-packages (from scikit-learn) (3.6.0)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.11/dist-packages (from python-dateutil>=2.8.2->pandas) (1.17.0)

```
from google.colab import files
uploaded = files.upload()
```

Choose Files StudentsPerformance.csv

• StudentsPerformance.csv(text/csv) - 72036 bytes, last modified: 6/2/2025 - 100% done

Saving StudentsPerformance.csv to StudentsPerformance.csv

```
import pandas as pd

# Load the CSV file into a DataFrame
df = pd.read_csv('StudentsPerformance.csv')
df.head()
```

	gender	race/ethnicity	parental level of education	lunch	test preparation course	math score	reading score	writing score
0	female	group B	bachelor's degree	standard	none	72	72	74
1	female	group C	some college	standard	completed	69	90	88
2	female	group B	master's degree	standard	none	90	95	93
3	male	group A	associate's degree	free/reduced	none	47	57	44

Next steps: [Generate code with df](#) [View recommended plots](#) [New interactive sheet](#)

```
from sklearn.preprocessing import LabelEncoder

# Convert gender, race/ethnicity, parental level of education, etc.
label_encoders = {}
for column in df.select_dtypes(include='object').columns:
    le = LabelEncoder()
    df[column] = le.fit_transform(df[column])
    label_encoders[column] = le

# Create a binary target variable
df['average_score'] = df[['math score', 'reading score', 'writing score']].mean(axis=1)
df['pass'] = (df['average_score'] >= 60).astype(int)
df = df.drop(columns=['average_score'])

from sklearn.model_selection import train_test_split

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

X_train, X_test, y_train, y_test =
```

What can I help you build?

+ >

```
from sklearn.linear_model import LogisticRegression
```

```
lr_model = LogisticRegression()  
lr_model.fit(X_train, y_train)  
y_pred_lr = lr_model.predict(X_test)
```

➡ /usr/local/lib/python3.11/dist-packages/sklearn/linear_model/_logistic.py:465: ConvergenceWarning: lbfgs failed to converge (status=1):
STOP: TOTAL NO. OF ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:

<https://scikit-learn.org/stable/modules/preprocessing.html>

Please also refer to the documentation for alternative solver options:

https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression

```
n_iter_i = _check_optimize_result(
```

```
from sklearn.neighbors import KNeighborsClassifier
```

```
knn_model = KNeighborsClassifier(n_neighbors=5)  
knn_model.fit(X_train, y_train)  
y_pred_knn = knn_model.predict(X_test)
```

```
from sklearn.tree import DecisionTreeClassifier
```

```
dt_model = DecisionTreeClassifier(random_state=42)  
dt_model.fit(X_train, y_train)  
y_pred_dt = dt_model.predict(X_test)
```

```
from sklearn.metrics import classification_report
```

```
print("Logistic Regression:\n", classification_report(y_test, y_pred_lr))  
print("k-NN:\n", classification_report(y_test, y_pred_knn))  
print("Decision Tree:\n", classification_report(y_test, y_pred_dt))
```

➡ Logistic Regression:

	precision	recall	f1-score	support
0	0.94	0.98	0.96	62
1	0.99	0.97	0.98	138
accuracy			0.97	200
macro avg	0.97	0.98	0.97	200
weighted avg	0.98	0.97	0.98	200

k-NN:

	precision	recall	f1-score	support
0	0.95	0.98	0.97	62
1	0.99	0.98	0.99	138
accuracy			0.98	200
macro avg	0.97	0.98	0.98	200
weighted avg	0.98	0.98	0.98	200

Decision Tree:

	precision	recall	f1-score	support
0	0.92	0.95	0.94	62
1	0.98	0.96	0.97	138
accuracy			0.96	200
macro avg	0.95	0.96	0.95	200
weighted avg	0.96	0.96	0.96	200

