```
i Star notebook in Google Drive
import numpy as np

from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder

from sklearn.linear_model import LogisticRegression
from sklearn.tree import DecisionTreeClassifier
from sklearn.neighbors import KNeighborsClassifier

from sklearn.metrics import accuracy_score, precision_score, f1_score
```

```
df = pd.read_csv("/content/Titanic-Dataset.csv")
df
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN	S
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85	C
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	S
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123	S
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	NaN	S
...	...	...	...	...	...	...	...	...	...	...	...	...
886	887	0	2	Montvila, Rev. Juozas	male	27.0	0	0	211536	13.0000	NaN	S
887	888	1	1	Graham, Miss. Margaret Edith	female	19.0	0	0	112053	30.0000	B42	S
888	889	0	3	Johnston, Miss. Catherine Helen "Carrie"	female	NaN	1	2	W./C. 6607	23.4500	NaN	S

```
df['Age'] = df['Age'].fillna(df['Age'].mean())
```

```
from sklearn.preprocessing import LabelEncoder

le = LabelEncoder()
df['Sex'] = le.fit_transform(df['Sex'])
```

```
df = df[['Survived', 'Pclass', 'Sex', 'Age', 'Fare']]
df.head()
```

	Survived	Pclass	Sex	Age	Fare
0	0	3	1	22.0	7.2500
1	1	1	0	38.0	71.2833
2	1	3	0	26.0	7.9250
3	1	1	0	35.0	53.1000
4	0	3	1	35.0	8.0500

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

```
from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.2, random_state=42
)
```

```
from sklearn.linear_model import LogisticRegression

lr = LogisticRegression(max_iter=1000)
lr.fit(X_train, y_train)
lr_pred = lr.predict(X_test)
```

```
from sklearn.tree import DecisionTreeClassifier

d Star notebook in Google Drive ier()
dt.fit(X_train, y_train)
dt_pred = dt.predict(X_test)
```

```
from sklearn.neighbors import KNeighborsClassifier

knn = KNeighborsClassifier(n_neighbors=5)
knn.fit(X_train, y_train)
knn_pred = knn.predict(X_test)
```

```
from sklearn.metrics import accuracy_score, precision_score, f1_score
```

Logistic Regression

```
lr_acc = accuracy_score(y_test, lr_pred)
lr_prec = precision_score(y_test, lr_pred)
lr_f1 = f1_score(y_test, lr_pred)
```

Decision Tree

```
dt_acc = accuracy_score(y_test, dt_pred)
dt_prec = precision_score(y_test, dt_pred)
dt_f1 = f1_score(y_test, dt_pred)
```

KNN

```
knn_acc = accuracy_score(y_test, knn_pred)
knn_prec = precision_score(y_test, knn_pred)
knn_f1 = f1_score(y_test, knn_pred)
```

Final Table

```
results = pd.DataFrame({
    'Model': ['Logistic Regression', 'Decision Tree', 'KNN'],
    'Accuracy': [lr_acc, dt_acc, knn_acc],
    'Precision': [lr_prec, dt_prec, knn_prec],
    'F1-score': [lr_f1, dt_f1, knn_f1]
})

results
```

	Model	Accuracy	Precision	F1-score
0	Logistic Regression	0.798883	0.771429	0.750000
1	Decision Tree	0.743017	0.684211	0.693333
2	KNN	0.698324	0.678571	0.584615

```
Start coding or generate with AI.
```