Generate | randomly select 5 items from a list | Close

```
import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score, classification_report


df = pd.read_csv('/content/gender_submission.csv')


X = df.drop(columns=['Survived'])
y = df['PassengerId']


# Split data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)


# Standardize the features
scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)


# Train logistic regression model
model = LogisticRegression()
model.fit(X_train, y_train)
```

```
▾ LogisticRegression
LogisticRegression()
```

```
# Make predictions
y_pred = model.predict(X_test)


# Evaluate the model
accuracy = accuracy_score(y_test, y_pred)
print(f'Accuracy: {accuracy:.2f}')
print('Classification Report:')
print(classification_report(y_test, y_pred))
```

```
Accuracy: 0.00
Classification Report:
              precision    recall  f1-score   support

         892       0.00      0.00      0.00       1.0
         893       0.00      0.00      0.00       0.0
         901       0.00      0.00      0.00       1.0
         907       0.00      0.00      0.00       1.0
         909       0.00      0.00      0.00       1.0
         910       0.00      0.00      0.00       0.0
         911       0.00      0.00      0.00       1.0
         914       0.00      0.00      0.00       1.0
         916       0.00      0.00      0.00       1.0
         917       0.00      0.00      0.00       1.0
         922       0.00      0.00      0.00       1.0
         925       0.00      0.00      0.00       1.0
         931       0.00      0.00      0.00       1.0
         934       0.00      0.00      0.00       1.0
         938       0.00      0.00      0.00       1.0
         941       0.00      0.00      0.00       0.0
         947       0.00      0.00      0.00       1.0
         948       0.00      0.00      0.00       1.0
         949       0.00      0.00      0.00       1.0
         951       0.00      0.00      0.00       0.0
         958       0.00      0.00      0.00       1.0
         962       0.00      0.00      0.00       1.0
         964       0.00      0.00      0.00       1.0
         965       0.00      0.00      0.00       1.0
         966       0.00      0.00      0.00       0.0
         967       0.00      0.00      0.00       1.0
         968       0.00      0.00      0.00       1.0
         969       0.00      0.00      0.00       1.0
         970       0.00      0.00      0.00       1.0
         971       0.00      0.00      0.00       1.0
```

974	0.00	0.00	0.00	1.0
981	0.00	0.00	0.00	0.0
982	0.00	0.00	0.00	1.0
985	0.00	0.00	0.00	1.0
986	0.00	0.00	0.00	0.0
996	0.00	0.00	0.00	1.0
1002	0.00	0.00	0.00	1.0
1005	0.00	0.00	0.00	1.0
1009	0.00	0.00	0.00	0.0
1016	0.00	0.00	0.00	0.0
1018	0.00	0.00	0.00	1.0
1029	0.00	0.00	0.00	1.0
1031	0.00	0.00	0.00	0.0
1033	0.00	0.00	0.00	1.0
1040	0.00	0.00	0.00	1.0
1042	0.00	0.00	0.00	0.0
1045	0.00	0.00	0.00	1.0
1047	0.00	0.00	0.00	1.0
1057	0.00	0.00	0.00	1.0
1060	0.00	0.00	0.00	1.0
1067	0.00	0.00	0.00	1.0
1072	0.00	0.00	0.00	1.0
1091	0.00	0.00	0.00	0.0