Model Metrics

**Final model** =

X = df[['Genre', 'Platform', 'Publisher']]

Y = df['NA\_Sales']

X = pd.get\_dummies(data=X, drop\_first=True)

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, Y, test\_size = .20, random\_state = 42)

prac\_model = RandomForestRegressor()

prac\_model = RandomizedSearchCV(estimator = prac\_model, param\_distributions = random\_grid, n\_iter = 100, cv = 3, verbose=2, random\_state=42, n\_jobs = -1)

prac\_model.fit(X\_train, y\_train)

prac\_model.best\_params\_

model1 = RandomForestRegressor(n\_estimators = 70, min\_samples\_split = 10, min\_samples\_leaf = 2, max\_features = 'sqrt', max\_depth = 80, bootstrap = True)model1.fit(X\_train, y\_train)

y\_pred = model1.predict(X\_test)

print('Mean Absolute Error:', metrics.mean\_absolute\_error(y\_test, y\_pred))

print('Mean Squared Error:', metrics.mean\_squared\_error(y\_test, y\_pred))

print('Root Mean Squared Error:', np.sqrt(metrics.mean\_squared\_error(y\_test, y\_pred)))

**Features =**

X = df[['Genre', 'Platform', 'Publisher']]

Y = df['NA\_Sales']

X = pd.get\_dummies(data=X, drop\_first=True)

**Parameters =**

'n\_estimators': 70,

'min\_samples\_split': 10,

'min\_samples\_leaf': 2,

'max\_features': 'sqrt',

'max\_depth': 80,

'bootstrap': True

**Performance Metrics =**

Mean Absolute Error: 0.34266039607172505

Mean Squared Error: 1.4257775904054613

Root Mean Squared Error: 1.1940592909924788