

```
In [21]: ▶ import pandas as pd

# Load dataset
ds = pd.read_csv("Ant19Data.csv")
ds.head()
```

Out[21]:

		COMP	LOC	WMC	DIT	NOC
0	org.apache.tools.ant.util.regexp.JakartaRegexp...	-0.455390	-0.644382	-0.660476	-0.141926	
1	org.apache.tools.ant.taskdefs.GUnzip.java	-0.403774	-0.569526	-0.660476	-0.141926	
2	org.apache.tools.ant.taskdefs.condition.Equals...	-0.201800	-0.045537	0.654215	-0.141926	
3	org.apache.tools.ant.taskdefs.optional.ccm.CCM...	-0.628190	-0.794093	-0.660476	-0.141926	
4	org.apache.tools.ant.listener.Log4jListener.java	-0.107546	-0.270104	0.654215	-0.141926	

```
In [32]: ▶ #ant16_input_features=["MaX_CC", "CE", "MOA", "LCOM", "MFA", "CBM", "DAM", "CA"]
#optimal_ant16_input_features=["MaX_CC", "CE"]#optimal

#ant18_input_features=["MFA", "CAM", "MOA", "LCOM", "AMC", "CE", "DAM", "NOC", "MFA"]
#optimal_ant18_input_features=["CAM", "MOA", "LCOM"]#optimal

#ant19_input_features=["CA", "AMC", "MOA", "CAM", "MaX_CC", "DIT", "CE", "NOC", "MFA"]
#optimal_ant19_input_features=["CA", "AMC", "MOA", "CAM"] #optimal

X = ds[ant19_input_featuresExcept_MFA] # Features
y = ds.Sum_Churn # Target variable
```

```
In [33]: ▶ from sklearn.inspection import permutation_importance
from sklearn.ensemble import RandomForestRegressor

# Create a random forest classifier
clf = RandomForestRegressor(random_state=42)

# Train the classifier on the data
clf.fit(X, y)
```

Out[33]: RandomForestRegressor(random_state=42)

```
In [34]: ▶ #permutation FI
# Calculate the permutation importance of each feature
result = permutation_importance(clf, X, y, n_repeats=10, random_state=42)

# Print the feature importance scores
for i in range(len(result.importances_mean)):
    print(f"Feature {i}: {result.importances_mean[i]}")
```

```
Feature 0: 0.30438493580737075
Feature 1: 0.410695708809279
Feature 2: 0.057865844246599706
Feature 3: 0.5769285139558541
Feature 4: 0.3258902820898531
Feature 5: 0.04144056316197568
Feature 6: 0.1651031643960872
Feature 7: 0.03022389718261167
Feature 8: 0.2607769860898778
Feature 9: 0.08846967701139488
```

```
In [35]: ▶ feature_importance = pd.DataFrame({'Feature': X.columns,
                                             'Importance': result.importances_mean,
                                             'Standard Deviation': result.importances_std})
feature_importance = feature_importance.sort_values('Importance', ascending=False)
print(feature_importance)
```

	Feature	Importance	Standard Deviation
7	NOC	0.030224	0.001422
5	DIT	0.041441	0.002921
2	MOA	0.057866	0.003136
9	DAM	0.088470	0.007599
6	CE	0.165103	0.006191
8	LCOM	0.260777	0.009308
0	CA	0.304385	0.021287
4	MaX_CC	0.325890	0.017184
1	AMC	0.410696	0.012160
3	CAM	0.576929	0.017662

```
In [36]: ▶ # #impurity FP
# feature_importances = clf.feature_importances_
# #print(feature_importances)

# #Print the feature importance scores
# for i in range(len(feature_importances)):
#     print(f"Feature {i}: {feature_importances[i]}")
```

In [37]: ▶

```
# import numpy as np

# from sklearn.ensemble import RandomForestClassifier
# import matplotlib.pyplot as plt

# # Sort indices in descending order of feature importance
# indices = np.argsort(feature_importances[::-1])

# # Plot the feature importances
# plt.figure(figsize=(8, 6))
# plt.title("Feature Importance - Random Forest")
# plt.bar(range(X.shape[1]), feature_importances[indices], align="center")
# plt.xticks(range(X.shape[1]), np.array(X)[indices])
# plt.ylabel("Normalized Importance")
# plt.show()
```

In []: ▶

In []: ▶