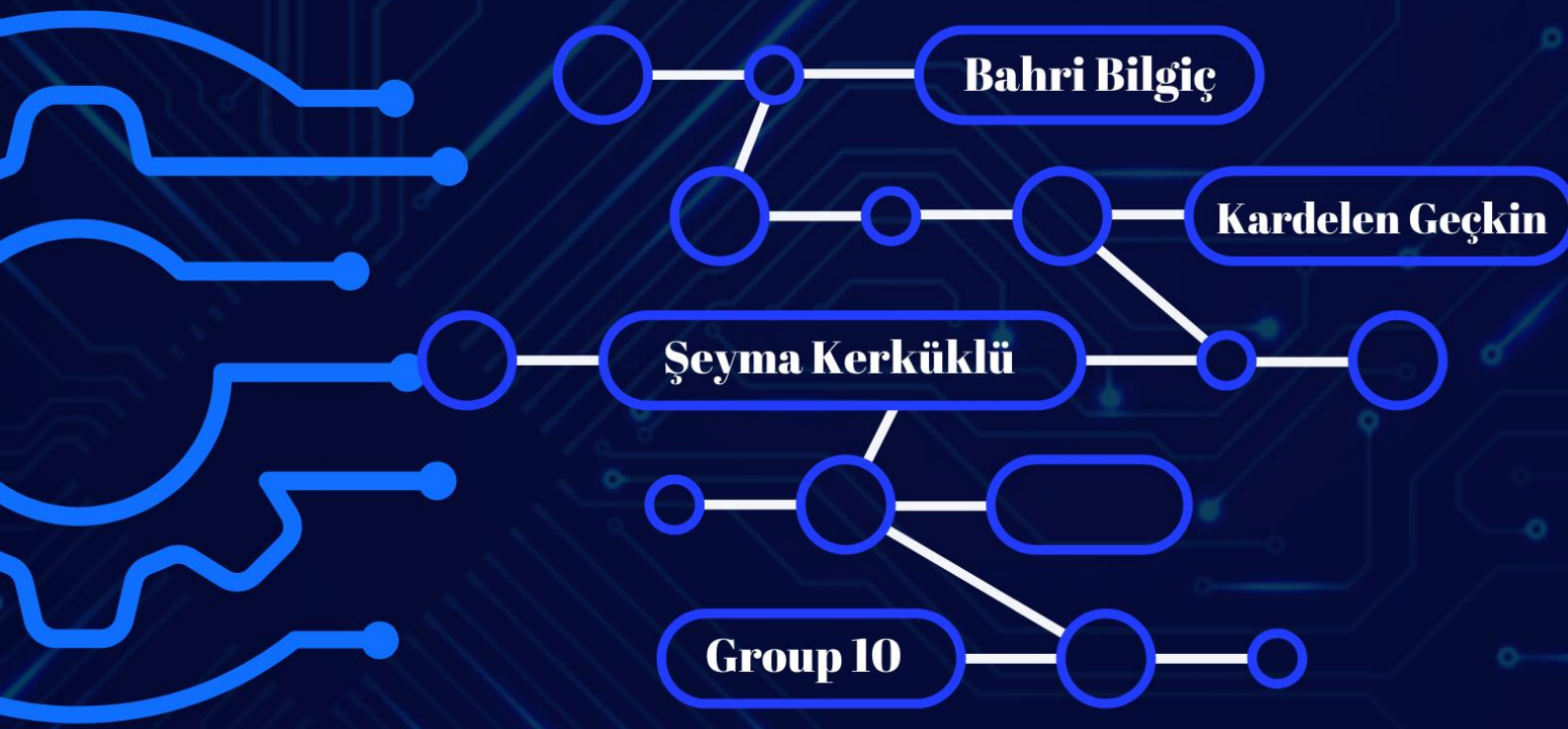
STATISTICAL LEARNING FROM DATA: APPLICATIONS IN PHYSICS

Comparative Analysis of Gaussian Naive Bayes Implementations on Breast Cancer Data



CONTENTS

1. Methods

Our code begins by loading breast cancer data and visualizing feature correlations through a heatmap. To address multicollinearity, a subset of features is selected, and a Gaussian Naive Bayes classifier, implemented in the custom `GaussianNB` class, is trained on the chosen features. The model is then evaluated on a test set, and the accuracy score is computed. The code provides a focused exploration of feature subsets using Naive Bayes classification, contributing to a concise and informative analysis of breast cancer data.

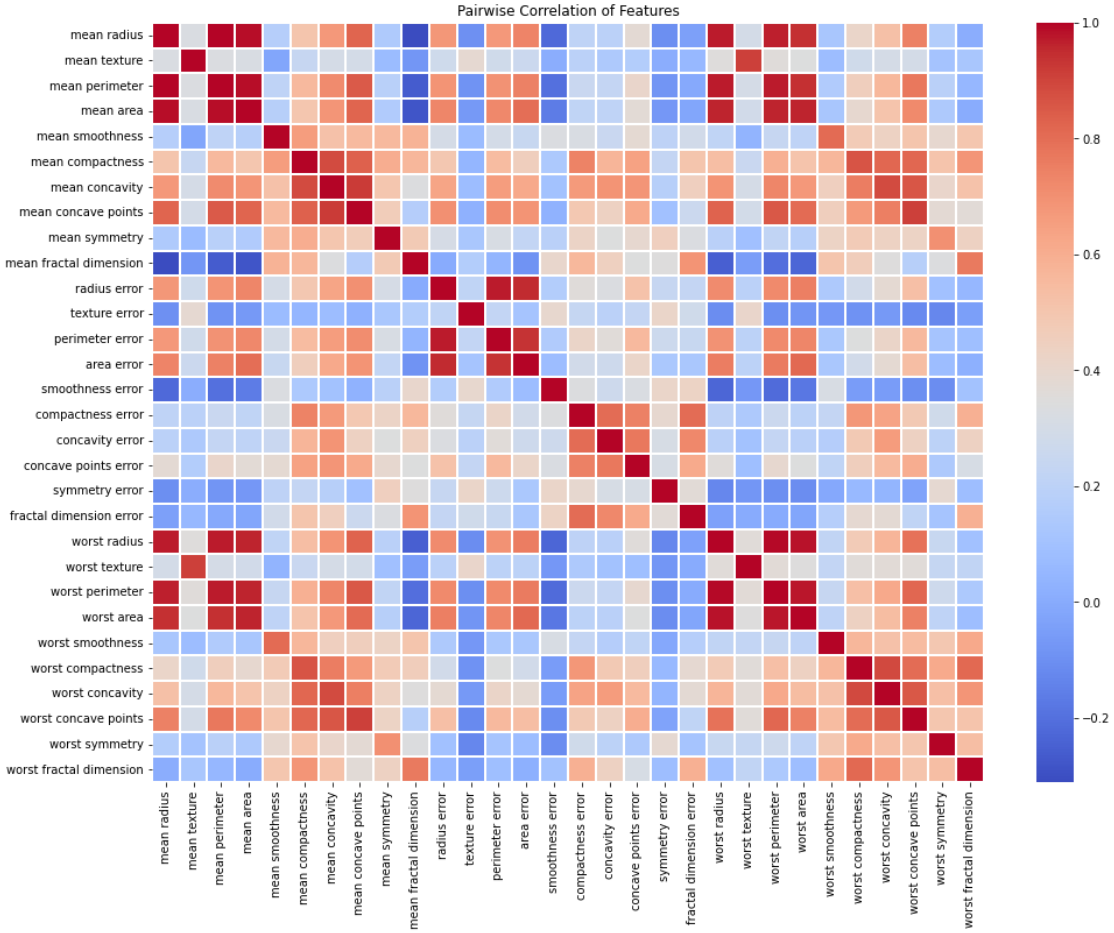
2. Homework 4

2.1 Application

```
import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
from sklearn.datasets import load_breast_cancer
from nv_hw3 import GaussianNB
import matplotlib.pyplot as plt
import seaborn as sns

# Load breast cancer data using load_breast_cancer
cancer = load_breast_cancer()



# Get the data and target
X, y = cancer.data, cancer.target

# Get feature names
feature_names = cancer.feature_names
print ( feature_names )



# Create a pandas dataframe
```

```
df_cancer = pd.DataFrame(data=cancer.data, columns=cancer.feature_names)

# Compute pairwise correlation of features
corr = df_cancer.corr(method="pearson")

# Plot correlations using seaborn or any other plotting tool .
plt.figure(figsize=(16, 12))
sns.heatmap(corr, cmap='coolwarm', fmt=".2f", linewidths=0.1)
plt.title("Pairwise Correlation of Features")
plt.show()


# Can you use all of the features ? Remember the fundamental assumption of
# We explain it out reports(app_naive_bayes).

# There are many features . Select some of them ( maybe four or five ) and get
# predictions for your test samples using GaussianNB class that you have
# implemented in the previous homework .
selected_features = ['mean symmetry', 'mean texture', 'mean perimeter', 'mean
concave points', 'mean concavity']


# Your selected features
X_selected = df_cancer[selected_features].to_numpy()

# Labels of your selected features
df_cancer['target'] = y
y_selected = df_cancer['target'].values

# Split the data using train_test_split
X_train, X_test, y_train, y_test = train_test_split(X_selected, y, test_size=0.1,
random_state=42)


clf = GaussianNB()
clf.fit(X_train, y_train)
predictions = clf.predict(X_test)
accuracy = accuracy_score(y_test, predictions)
print("Accuracy score:", accuracy)
```

2.2 Figures



3. Question

3.1 Can you use all of the features? Remember the fundamental assumption of naive bayes. Explain your thinking.

The fundamental assumption of Naive Bayes, whether Gaussian or another variant, is the independence assumption. Specifically, it assumes that all features are conditionally independent given the class label. This independence assumption simplifies the computation of probabilities. In the code, the feature selection step has chosen a subset of features with low pairwise correlations for analysis. However, since Naive Bayes assumes feature independence, using all features does not violate this assumption. In fact, using all features might capture more information and potentially lead to better model performance.