PRINCIPAL - COMPONENT ANALYSIS

BY – SWASTIK SAMADDER

PRINCIPAL COMPONENT ANALYSIS AND LOGISTIC REGRESSION USING PYTHON PROGRAMMING

The Breast Cancer Wisconsin (Diagnostic) dataset is a widely used multivariate dataset in machine learning, primarily for classification tasks. Each instance in the dataset represents a patient, and the goal is to classify the tumor as either **malignant (cancerous)** or **benign (non-cancerous).**

The dataset contains 30 numerical features, which describe various characteristics of the cell nuclei present in the image.
They comprise **10 core measurements** each reported as a **mean, standard error, and "worst" (largest) value.** The 10 core characteristics of the cell nuclei are:
Radius, Texture, Perimeter, Area, Smoothness, Compactness, Concavity, Concave points, Symmetry, Fractal dimension.
Its clear separation into two classes and the availability of diverse, well-defined features make it an excellent resource for learning and applying various machine learning algorithms, particularly in areas like classification, dimensionality reduction (e.g., PCA), and model evaluation.

```python
import pandas as pd
import numpy as np
from sklearn.preprocessing import StandardScaler
from sklearn.decomposition import PCA
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.datasets import load_breast_cancer
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score, classification_report
```

```python
cancer = load_breast_cancer()
df = pd.DataFrame(data=cancer.data, columns=cancer.feature_names)
print(cancer.DESCR)
```

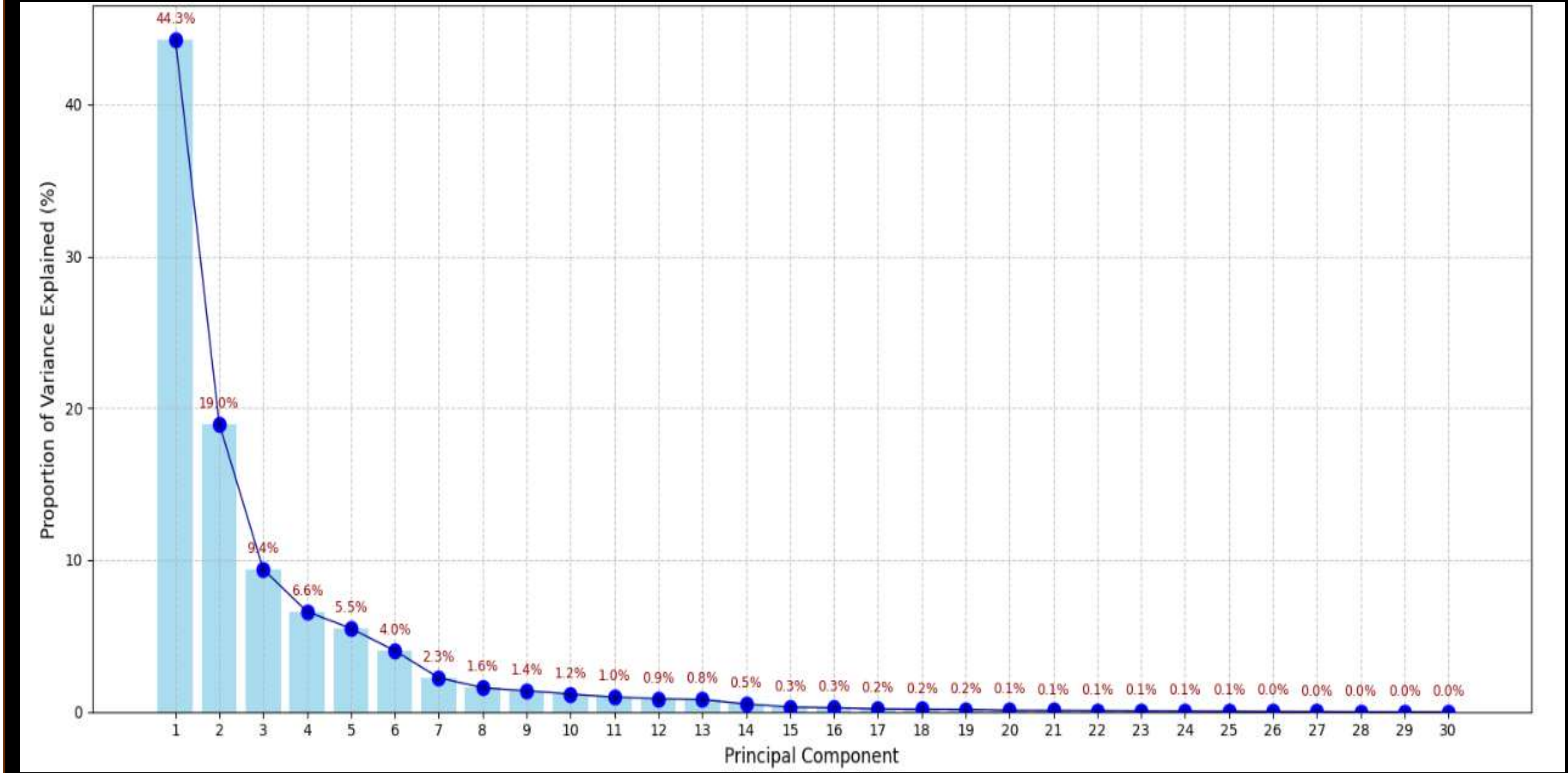
```
:Number of Instances: 569

:Number of Attributes: 30 numeric, predictive attributes and the class

:Attribute Information:
    - radius (mean of distances from center to points on the perimeter)
    - texture (standard deviation of gray-scale values)
    - perimeter
    - area
    - smoothness (local variation in radius lengths)
    - compactness (perimeter^2 / area - 1.0)
    - concavity (severity of concave portions of the contour)
    - concave points (number of concave portions of the contour)
    - symmetry
    - fractal dimension ("coastline approximation" - 1)

    The mean, standard error, and "worst" or largest (mean of the three
    worst/largest values) of these features were computed for each image,
    resulting in 30 features.  For instance, field 0 is Mean Radius, field
    10 is Radius SE, field 20 is Worst Radius.

    - class:
            - WDBC-Malignant
            - WDBC-Benign
```

```
:Summary Statistics:

===================================== ====== ======
                                        Min    Max
===================================== ====== ======
radius (mean):                        6.981  28.11
texture (mean):                       9.71   39.28
perimeter (mean):                     43.79  188.5
area (mean):                          143.5  2501.0
smoothness (mean):                    0.053  0.163
compactness (mean):                   0.019  0.345
concavity (mean):                     0.0    0.427
concave points (mean):                0.0    0.201
symmetry (mean):                      0.106  0.304
fractal dimension (mean):             0.05   0.097
radius (standard error):              0.112  2.873
texture (standard error):             0.36   4.885
perimeter (standard error):           0.757  21.98
area (standard error):                6.802  542.2
smoothness (standard error):          0.002  0.031
compactness (standard error):         0.002  0.135
concavity (standard error):           0.0    0.396
concave points (standard error):      0.0    0.053
symmetry (standard error):            0.008  0.079
fractal dimension (standard error):   0.001  0.03
radius (worst):                       7.93   36.04
texture (worst):                      12.02  49.54
perimeter (worst):                    50.41  251.2
area (worst):                         185.2  4254.0
smoothness (worst):                   0.071  0.223
compactness (worst):                  0.027  1.058
concavity (worst):                    0.0    1.252
concave points (worst):               0.0    0.291
symmetry (worst):                     0.156  0.664
fractal dimension (worst):            0.055  0.208
===================================== ====== ======
```

```python
# Step 1: Standardize the data
scaler = StandardScaler()
scaled_data = scaler.fit_transform(df)
pca = PCA()
res_pca = pca.fit(scaled_data)


explained_variance_ratio = pca.explained_variance_ratio_
standard_deviation = np.sqrt(pca.explained_variance_)
cumulative_proportion = np.cumsum(explained_variance_ratio)
pca_summary = pd.DataFrame({
    'Standard deviation': standard_deviation,
    'Proportion of Variance': explained_variance_ratio,
    'Cumulative Proportion': cumulative_proportion
}, index=[f'PC{i+1}' for i in range(len(standard_deviation))])
print("\nPCA Summary Table:")
print(pca_summary)
```

PCA Summary Table:

	Standard deviation	Proportion of Variance	Cumulative Proportion
PC1	3.647601	0.442720	0.442720
PC2	2.387755	0.189712	0.632432
PC3	1.680152	0.093932	0.726364
PC4	1.408591	0.066021	0.792385
PC5	1.285159	0.054958	0.847343
PC6	1.099765	0.040245	0.887588
PC7	0.822441	0.022507	0.910095
PC8	0.690982	0.015887	0.925983
PC9	0.646242	0.013896	0.939879
PC10	0.592715	0.011690	0.951569
PC11	0.542617	0.009797	0.961366
PC12	0.511489	0.008705	0.970071
PC13	0.491714	0.008045	0.978117
PC14	0.396593	0.005234	0.983350
PC15	0.307084	0.003138	0.986488
PC16	0.282849	0.002662	0.989150
PC17	0.243934	0.001980	0.991130
PC18	0.229590	0.001754	0.992884
PC19	0.222631	0.001649	0.994533
PC20	0.176676	0.001039	0.995572
PC21	0.173279	0.000999	0.996571
PC22	0.165794	0.000915	0.997486
PC23	0.156153	0.000811	0.998297
PC24	0.134487	0.000602	0.998899
PC25	0.124533	0.000516	0.999415
PC26	0.090510	0.000273	0.999688
PC27	0.083142	0.000230	0.999918
PC28	0.039902	0.000053	0.999971
PC29	0.027388	0.000025	0.999996
PC30	0.011545	0.000004	1.000000

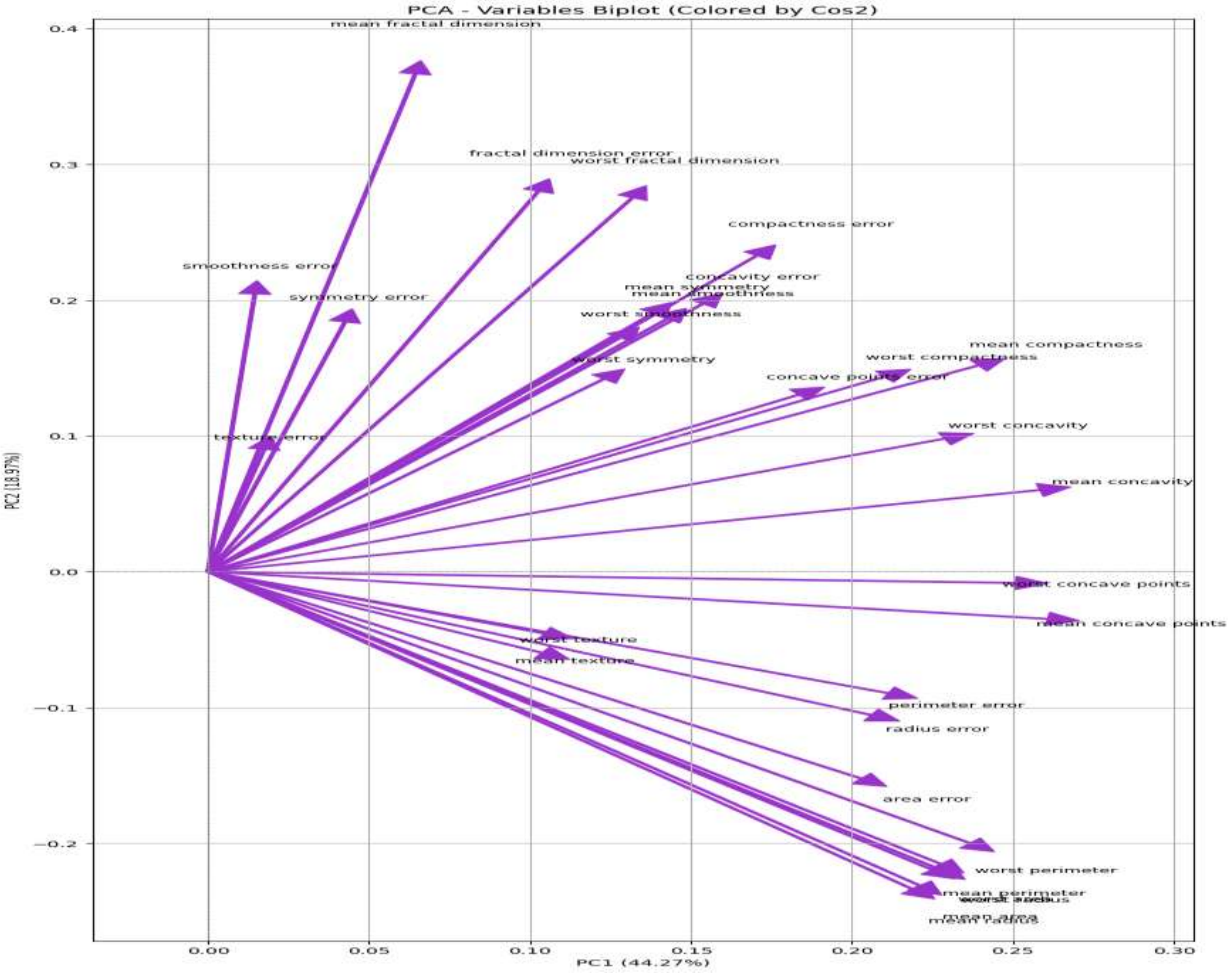


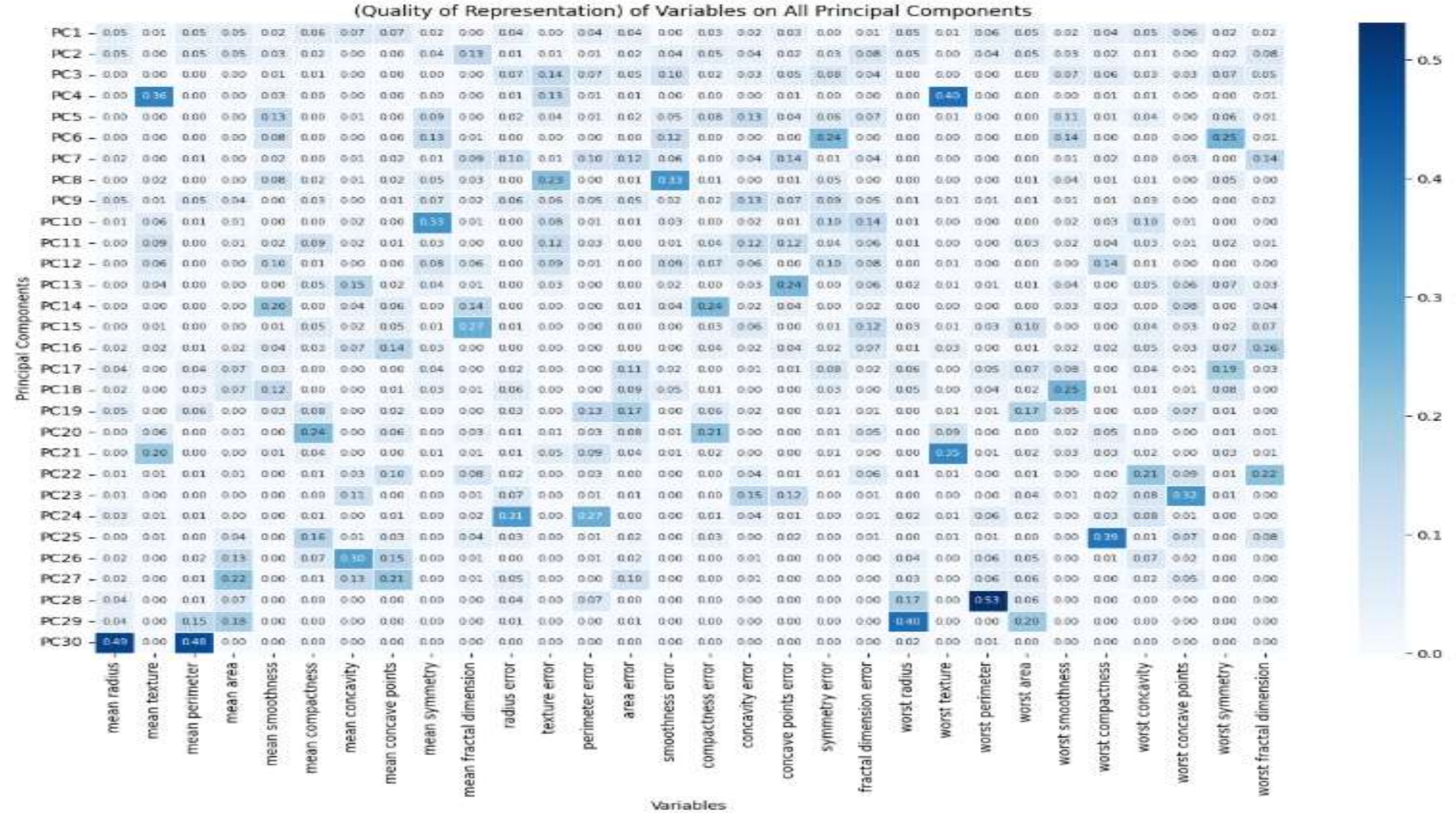
44.3%
19.0%
9.4%
6.6%
5.5%
4.0%
2.3%
1.6%
1.4%
1.2%
1.0%
0.9%
0.8%
0.5%
0.3%
0.3%
0.2%
0.2%
0.2%
0.1%
0.1%
0.1%
0.1%
0.1%
0.1%
0.0%
0.0%
0.0%
0.0%
0.0%
Proportion of Variance Explained (%)
Principal Component

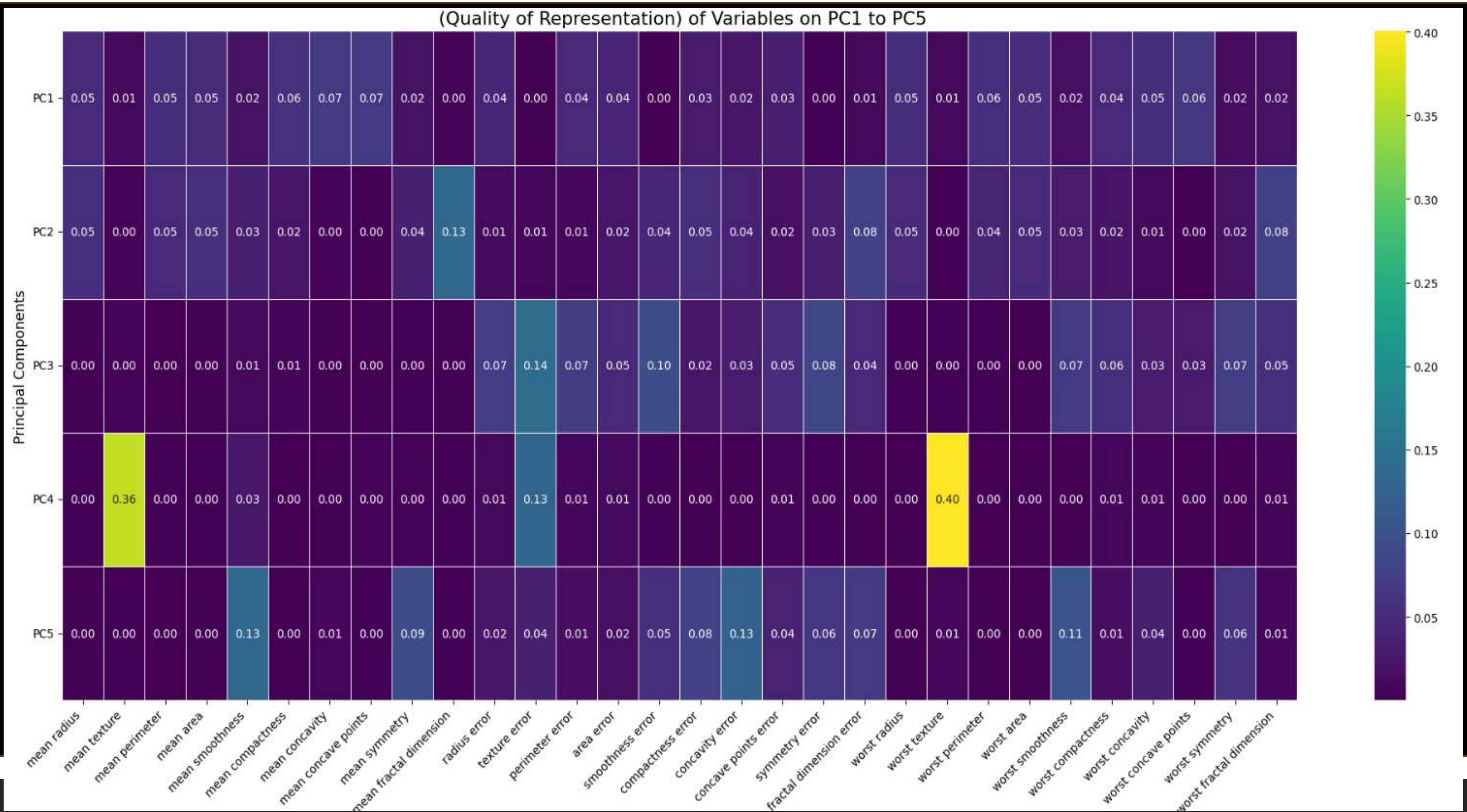
```python
plt.figure(figsize=(15,7))
# 1. Bar Plot for Explained Variance
plt.bar(range(1, len(explained_variance_ratio) + 1), explained_variance_ratio * 100,
        color='skyblue', alpha=0.7, label='Explained Variance')
# 2. Scatter Plot (on top of bars) for emphasis
plt.scatter(range(1, len(explained_variance_ratio) + 1), explained_variance_ratio * 100,
            marker='o', color='blue', s=80, zorder=5) # zorder to ensure dots are on top
plt.title('Scree Plot (Variance Explained by Principal Components)', fontsize=16)
plt.xlabel('Principal Component', fontsize=12)
plt.ylabel('Proportion of Variance Explained (%)', fontsize=12)
plt.xticks(range(1, len(explained_variance_ratio) + 1)) # Ensure all PC numbers are shown
plt.plot(range(1, len(explained_variance_ratio) + 1), explained_variance_ratio * 100,
         color='darkblue',  # Choose a color for the line
         linestyle='-',     # Solid line
         linewidth=1,       # Line thickness
         marker='o',        # Add circular markers at each point
         markersize=4,      # Size of the markers
         markerfacecolor='darkblue', # Fill color of the markers
         markeredgecolor='darkblue', # Edge color of the markers
         zorder=10)
# Annotate with percentage values
for i, txt in enumerate(explained_variance_ratio):
    plt.annotate(f'{txt * 100:.1f}%', # Format as percentage with one decimal place
                 (i + 1, txt * 100),
                 textcoords="offset points",
                 xytext=(0, 10), # Offset text slightly above the point
                 ha='center',
                 fontsize=9,
                 color='darkred') # Make annotation color stand out

plt.grid(axis='y', linestyle='--', alpha=0.7) # Grid lines only on y-axis for clarity
plt.grid(axis='x', linestyle='--', alpha=0.7) # Grid lines only on x-axis for clarity

plt.tight_layout() # Adjust layout to prevent labels from overlapping
plt.show()
```

```python
# Perform PCA on the standardized features (no n_components specified to get all for full Cos2 calculation)
pca = PCA()
pca.fit(X_scaled_df)

# Calculate Cos2 (Quality of Representation)
components_df = pd.DataFrame(pca.components_, columns=X_scaled_df.columns, index=[f'PC{i+1}' for i in range(pca.n_components_)])
cos2_matrix = components_df**2
cos2_matrix_sum = cos2_matrix.sum(axis=0)
cos2_normalized = cos2_matrix.divide(cos2_matrix_sum, axis=1)

# Ensure the index for PC selection is correct
pc_indices_to_plot = [f'PC{i+1}' for i in range(5)] # Select PC1 to PC5

# Generate the heatmap for the first 5 PCs
plt.figure(figsize=(20, 10)) # Adjusted figsize for better readability with 5 PCs
sns.heatmap(cos2_normalized.loc[pc_indices_to_plot],
            annot=True,
            cmap='viridis',
            fmt=".2f", # Format to two decimal places
            linewidths=.5)
plt.title(' (Quality of Representation) of Variables on PC1 to PC5', fontsize=16)
plt.xlabel('Variables', fontsize=12)
plt.ylabel('Principal Components', fontsize=12)
plt.xticks(rotation=45, ha='right') # Rotate x-axis labels for better readability
plt.yticks(rotation=0) # Keep y-axis labels horizontal
plt.tight_layout()
plt.show()
```

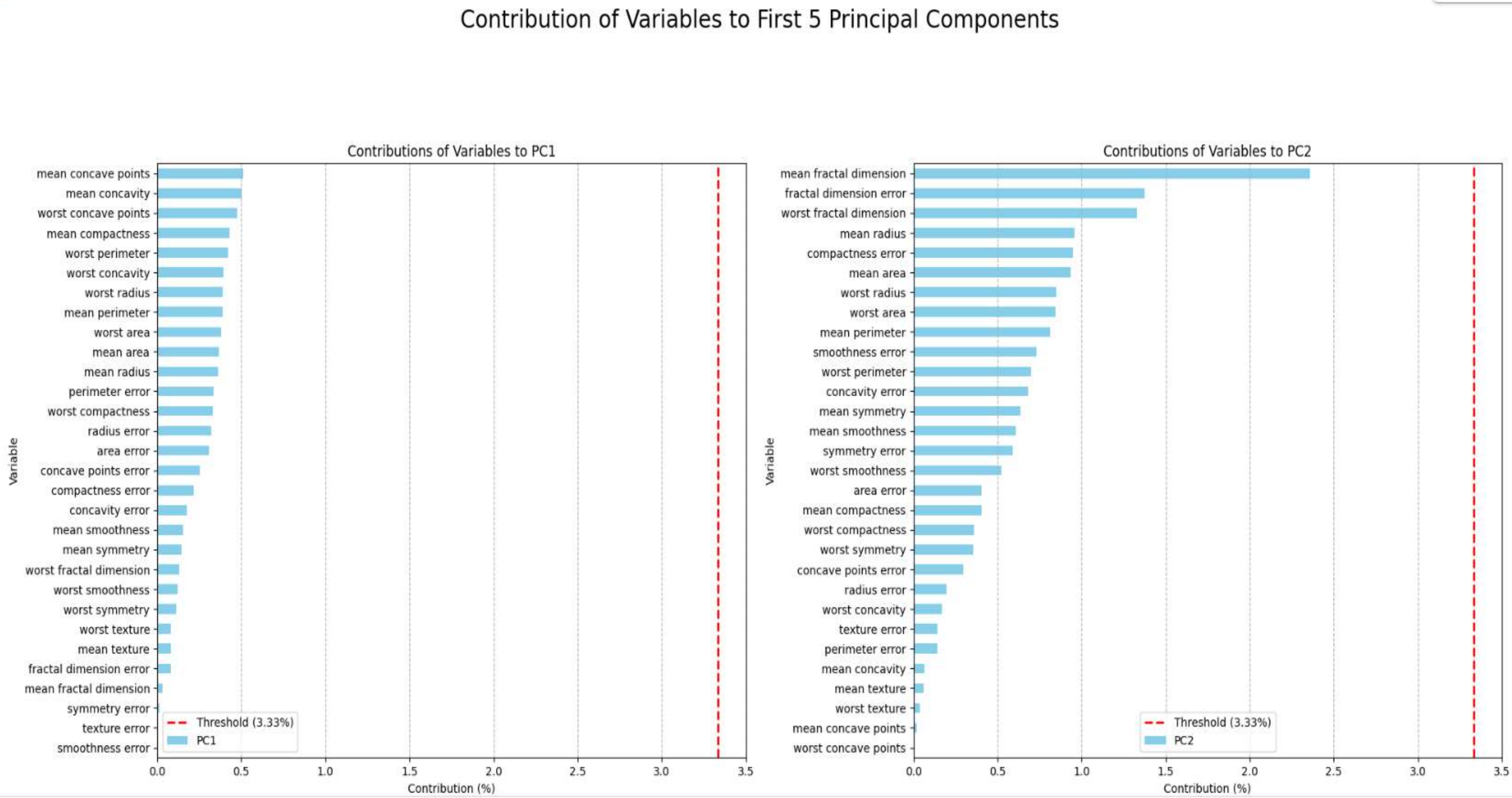
```python
plt.figure(figsize=(16, 10)) # Increased size for better readability with more features
sns.heatmap(cos2_normalized, annot=True, cmap='Blues', fmt=".2f", linewidths=.5, annot_kws={"size": 7})
plt.title(' (Quality of Representation) of Variables on All Principal Components')
plt.xlabel('Variables')
plt.ylabel('Principal Components')
plt.show()
```

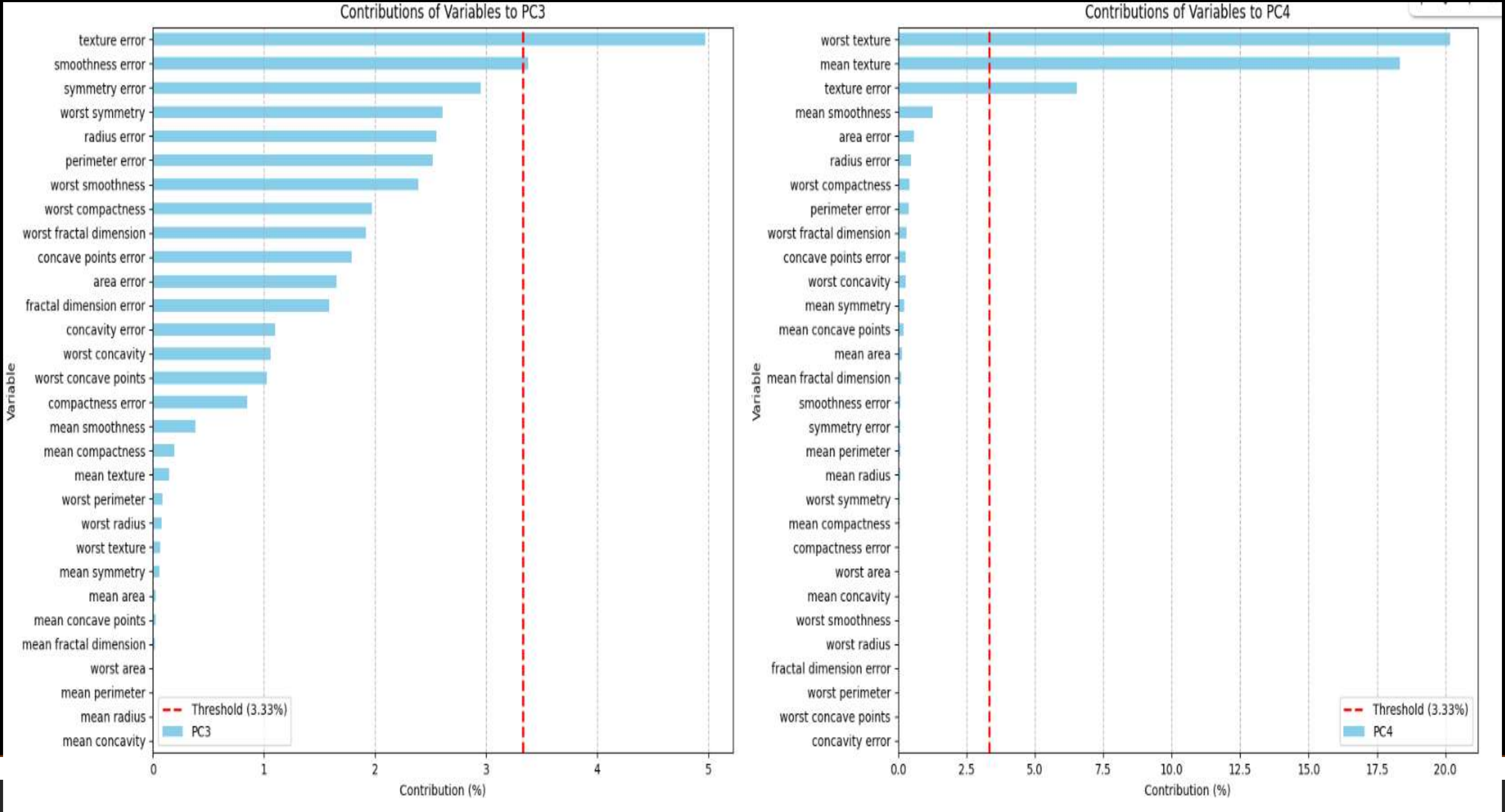

PCA - Variables Biplot (Colored by Cos2)
PC1 (44.27%)
PC2 (18.97%)
0.00
0.05
0.10
0.15
0.20
0.25
0.30
0.4
0.3
0.2
0.1
0.0
−0.1
−0.2
mean fractal dimension
fractal dimension error
worst fractal dimension
compactness error
smoothness error
symmetry error
concavity error
mean symmetry
mean smoothness
worst smoothness
mean compactness
worst compactness
worst symmetry
concave points error
worst concavity
texture error
mean concavity
worst concave points
mean concave points
worst texture
mean texture
perimeter error
radius error
area error
worst perimeter
mean perimeter
worst radius
worst area
mean area
mean radius

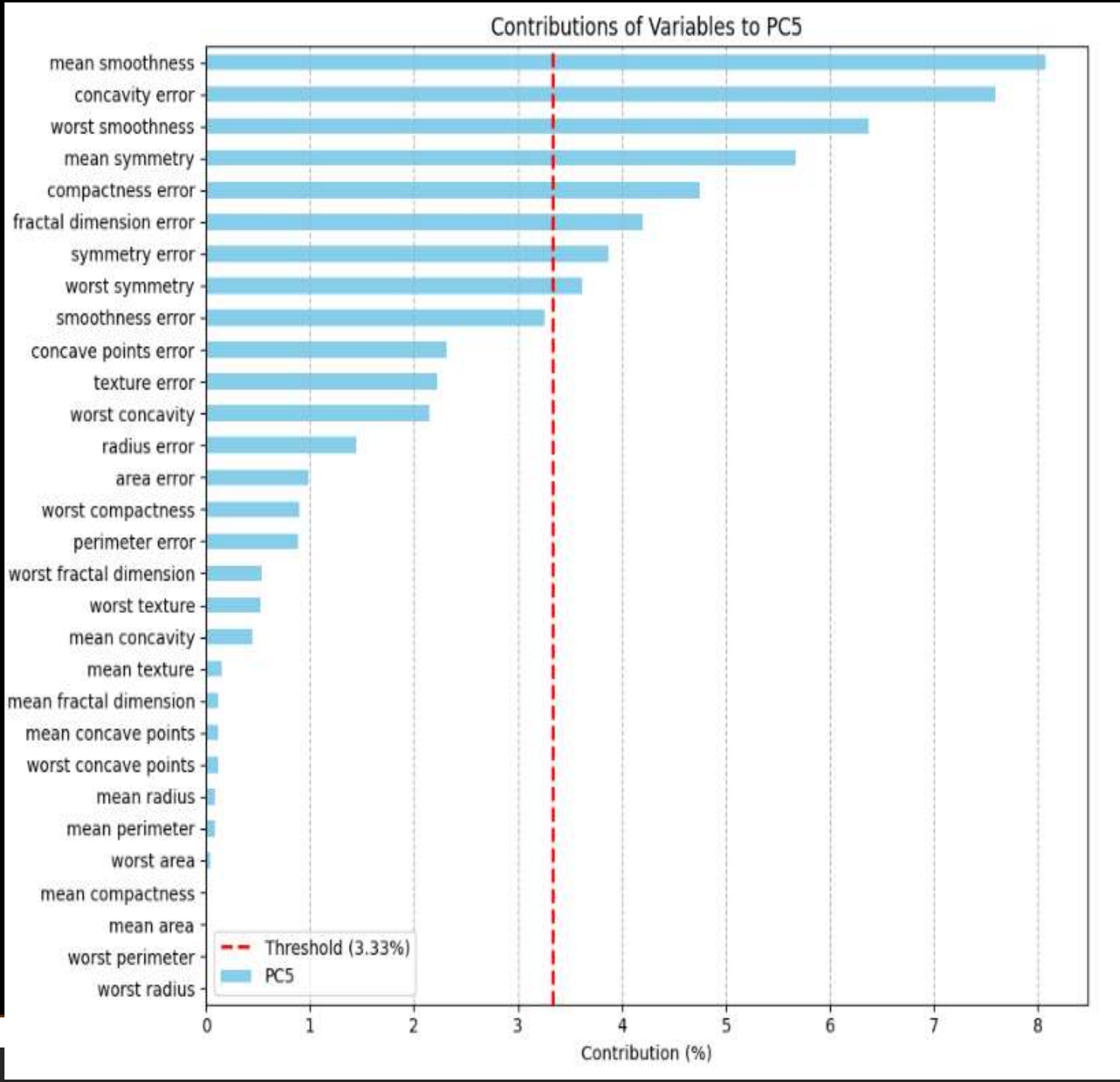

(Quality of Representation) of Variables on All Principal Components
Principal Components
PC1
PC2
PC3
PC4
PC5
PC6
PC7
PC8
PC9
PC10
PC11
PC12
PC13
PC14
PC15
PC16
PC17
PC18
PC19
PC20
PC21
PC22
PC23
PC24
PC25
PC26
PC27
PC28
PC29
PC30
mean radius
mean texture
mean perimeter
mean area
mean smoothness
mean compactness
mean concavity
mean concave points
mean symmetry
mean fractal dimension
radius error
texture error
perimeter error
area error
smoothness error
compactness error
concavity error
concave points error
symmetry error
fractal dimension error
worst radius
worst texture
worst perimeter
worst area
worst smoothness
worst compactness
worst concavity
worst concave points
worst symmetry
worst fractal dimension
Variables
0.5
0.4
0.3
0.2
0.1
0.0


(Quality of Representation) of Variables on PC1 to PC5
Principal Components
PC1
PC2
PC3
PC4
PC5
0.05 0.01 0.05 0.05 0.02 0.06 0.07 0.07 0.02 0.00 0.04 0.00 0.04 0.04 0.00 0.03 0.02 0.03 0.00 0.01 0.05 0.01 0.06 0.05 0.02 0.04 0.05 0.06 0.02 0.02
0.05 0.00 0.05 0.05 0.03 0.02 0.00 0.00 0.04 0.13 0.01 0.01 0.01 0.02 0.04 0.05 0.04 0.02 0.03 0.08 0.05 0.00 0.04 0.05 0.03 0.02 0.01 0.00 0.02 0.08
0.00 0.00 0.00 0.00 0.01 0.01 0.00 0.00 0.00 0.00 0.07 0.14 0.07 0.05 0.10 0.02 0.03 0.05 0.08 0.04 0.00 0.00 0.00 0.00 0.07 0.06 0.03 0.03 0.07 0.05
0.00 0.36 0.00 0.00 0.03 0.00 0.00 0.00 0.00 0.00 0.01 0.13 0.01 0.01 0.00 0.00 0.00 0.01 0.00 0.00 0.00 0.40 0.00 0.00 0.00 0.01 0.01 0.00 0.00 0.01
0.00 0.00 0.00 0.00 0.13 0.00 0.01 0.00 0.09 0.00 0.02 0.04 0.01 0.02 0.05 0.08 0.13 0.04 0.06 0.07 0.00 0.01 0.00 0.00 0.11 0.01 0.04 0.00 0.06 0.01
mean radius
mean texture
mean perimeter
mean area
mean smoothness
mean compactness
mean concavity
mean concave points
mean symmetry
mean fractal dimension
radius error
texture error
perimeter error
area error
smoothness error
compactness error
concavity error
concave points error
symmetry error
fractal dimension error
worst radius
worst texture
worst perimeter
worst area
worst smoothness
worst compactness
worst concavity
worst concave points
worst symmetry
worst fractal dimension
0.40
0.35
0.30
0.25
0.20
0.15
0.10
0.05

Threshold for equal contribution: 3.33%




Contributions of Variables to PC3
texture error
smoothness error
symmetry error
worst symmetry
radius error
perimeter error
worst smoothness
worst compactness
worst fractal dimension
concave points error
area error
fractal dimension error
concavity error
worst concavity
worst concave points
compactness error
mean smoothness
mean compactness
mean texture
worst perimeter
worst radius
worst texture
mean symmetry
mean area
mean concave points
mean fractal dimension
worst area
mean perimeter
mean radius
mean concavity
Variable
Threshold (3.33%)
PC3
0
1
2
3
4
5
Contribution (%)
Contributions of Variables to PC4
worst texture
mean texture
texture error
mean smoothness
area error
radius error
worst compactness
perimeter error
worst fractal dimension
concave points error
worst concavity
mean symmetry
mean concave points
mean area
mean fractal dimension
smoothness error
symmetry error
mean perimeter
mean radius
worst symmetry
mean compactness
compactness error
worst area
mean concavity
worst smoothness
worst radius
fractal dimension error
worst perimeter
worst concave points
concavity error
Variable
Threshold (3.33%)
PC4
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
20.0
Contribution (%)


Contributions of Variables to PC5
mean smoothness
concavity error
worst smoothness
mean symmetry
compactness error
fractal dimension error
symmetry error
worst symmetry
smoothness error
concave points error
texture error
worst concavity
radius error
area error
worst compactness
perimeter error
worst fractal dimension
worst texture
mean concavity
mean texture
mean fractal dimension
mean concave points
worst concave points
mean radius
mean perimeter
worst area
mean compactness
mean area
worst perimeter
worst radius
Threshold (3.33%)
PC5
0
1
2
3
4
5
6
7
8
Contribution (%)

```python
cancer = load_breast_cancer()
X = pd.DataFrame(data=cancer.data, columns=cancer.feature_names)
y = pd.Series(cancer.target, name='target')
print("Shape of original features (X):", X.shape)
print("Shape of target (y):", y.shape)
```

```
Shape of original features (X): (569, 30)
Shape of target (y): (569,)
```

```python
X_train, X_test, y_train, y_test = train_test_split(X_pca_selected, y, test_size=0.3, random_state=42, stratify=y)

print(f"\nTraining set shape - Features: {X_train.shape}, Target: {y_train.shape}")
print(f"Testing set shape - Features: {X_test.shape}, Target: {y_test.shape}")
```

```
Training set shape - Features: (398, 5), Target: (398,)
Testing set shape - Features: (171, 5), Target: (171,)
```

```python
# 7. Conduct Logistic Regression
logistic_model = LogisticRegression(random_state=42, max_iter=200)
logistic_model.fit(X_train, y_train)
```

```
LogisticRegression
LogisticRegression(max_iter=200, random_state=42)
```

```python
# Make predictions on the test set
y_pred = logistic_model.predict(X_test)
```

```python
# 8. Evaluate the model
accuracy = accuracy_score(y_test, y_pred)
class_report = classification_report(y_test, y_pred)
print(f"\nLogistic Regression Model Accuracy: {accuracy:.4f}")
print("\nClassification Report:")
print(class_report)
```

```
Logistic Regression Model Accuracy: 0.9766

Classification Report:
              precision    recall  f1-score   support

           0       0.97      0.97      0.97        64
           1       0.98      0.98      0.98       107

    accuracy                           0.98       171
   macro avg       0.98      0.98      0.98       171
weighted avg       0.98      0.98      0.98       171
```