```
#imports
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import sklearn as sk
import scipy as sp
import statsmodels as sm
import pyspark as ps
import os
```

```
from pyspark.sql import SparkSession
from pyspark.ml.feature import VectorAssembler

#Spark session
spark = SparkSession.builder \
    .appName("RandomForest") \
    .getOrCreate()
```

```
#data path
dataset_path = "C:\Users\Windl.DESKTOP-BF4IBNR\Downloads\oralcancer"

# Read the dataset
df = spark.read.csv(dataset_path, header=True, inferSchema=True)
print(df.head())
```

```
Row(ID=1, Country='Italy', Age=36, Gender='Female', Tobacco Use=1, Alcohol Consumption=1, HPV Infection=1, Betel Quid Use=0, Chr
```

```
#sklearn train/test split, randomforest with accuracy score, AUC, recall and precision
from sklearn.model_selection import train_test_split
from pyspark.ml.classification import RandomForestClassifier
from sklearn.metrics import accuracy_score, roc_auc_score, recall_score, precision_score
from pyspark.ml.feature import VectorAssembler
```

```
#preprocess
unique_values = df.select('Oral Cancer (Diag0sis)').distinct().collect()
print(unique_values)
```

```
[Row(Oral Cancer (Diag0sis)=1), Row(Oral Cancer (Diag0sis)=0)]
```

```
from pyspark.ml.feature import StringIndexer


df = spark.read.csv(dataset_path, header=True, inferSchema=True)

# index treatments
indexer_treatment = StringIndexer(inputCol="Treatment Type", outputCol="TreatmentType_indexed")
df = indexer_treatment.fit(df).transform(df)
indexer_diet = StringIndexer(inputCol="Diet (Fruits & Vegetables Intake)", outputCol="Diet_indexed")
df = indexer_diet.fit(df).transform(df)

# print schema
df.printSchema()
```

```
root
 |-- ID: integer (nullable = true)
 |-- Country: string (nullable = true)
 |-- Age: integer (nullable = true)
 |-- Gender: string (nullable = true)
 |-- Tobacco Use: integer (nullable = true)
 |-- Alcohol Consumption: integer (nullable = true)
 |-- HPV Infection: integer (nullable = true)
 |-- Betel Quid Use: integer (nullable = true)
 |-- Chronic Sun Exposure: integer (nullable = true)
 |-- Poor Oral Hygiene: integer (nullable = true)
 |-- Diet (Fruits & Vegetables Intake): string (nullable = true)
 |-- Family History of Cancer: integer (nullable = true)
 |-- Compromised Immune System: integer (nullable = true)
 |-- Oral Lesions: integer (nullable = true)
 |-- Unexplained Bleeding: integer (nullable = true)
 |-- Difficulty Swallowing: integer (nullable = true)
 |-- White or Red Patches in Mouth: integer (nullable = true)
```

```
 |-- Tumor Size (cm): double (nullable = true)
 |-- Cancer Stage: integer (nullable = true)
 |-- Treatment Type: string (nullable = true)
 |-- Survival Rate (5-Year, %): double (nullable = true)
 |-- Cost of Treatment (USD): double (nullable = true)
 |-- Economic Burden (Lost Workdays per Year): integer (nullable = true)
 |-- Early Diag0sis: integer (nullable = true)
 |-- Oral Cancer (Diag0sis): integer (nullable = true)
 |-- TreatmentType_indexed: double (nullable = false)
 |-- Diet_indexed: double (nullable = false)
```

```
#split train and test data
train_data, test_data = df.randomSplit([0.8, 0.2], seed=101)
```

```
# cancer present /absent
cancer_absent = df.filter(df['Oral Cancer (Diag0sis)'] == 0).count()
cancer_present = df.filter(df['Oral Cancer (Diag0sis)'] == 1).count()
```

```
# Identify feature columns (excluding ID, Country, Gender, Diet, and the label column itself)
feature_columns = [
    col for col in train_data.columns
    if col not in [
        'ID', 'Country', 'Gender', 'Diet (Fruits & Vegetables Intake)', 'Treatment Type',
        'Oral Cancer (Diag0sis)', 'Cancer Stage', 'Tumor Size (cm)'
    ]
]

# Assemble feature columns into a single vector column named 'features'
assembler = VectorAssembler(inputCols=feature_columns, outputCol='features')

# Transform the training and testing data
train_data_transformed = assembler.transform(train_data)
test_data_transformed = assembler.transform(test_data)

#run randomforest model
model = RandomForestClassifier(labelCol='Oral Cancer (Diag0sis)', featuresCol='features', numTrees=250)
model = model.fit(train_data_transformed)
```

```
#calculate accuracy, AUC, recall and precision on test data
predictions = model.transform(test_data_transformed)
y_true = [row['Oral Cancer (Diag0sis)'] for row in test_data_transformed.select('Oral Cancer (Diag0sis)').collect()]
y_pred = [row['prediction'] for row in predictions.select('prediction').collect()]
y_prob = [row['probability'][1] for row in predictions.select('probability').collect()]

accuracy = accuracy_score(y_true, y_pred)
auc = roc_auc_score(y_true, y_prob)
recall = recall_score(y_true, y_pred)
precision = precision_score(y_true, y_pred)

print("Accuracy:", accuracy)
print("AUC:", auc)
print("Recall:", recall)
print("Precision:", precision)

Accuracy: 1.0
AUC: 1.0
Recall: 1.0
Precision: 1.0
```

```
print("Correlation of 'Early Diag0sis' with 'Oral Cancer (Diag0sis)':")
display(df.stat.corr('Early Diag0sis', 'Oral Cancer (Diag0sis)'))

print("Correlation of 'Cancer Stage' with 'Oral Cancer (Diag0sis)':")
display(df.stat.corr('Cancer Stage', 'Oral Cancer (Diag0sis)'))

print("Correlation of 'Tumor Size (cm)' with 'Oral Cancer (Diag0sis)':")
display(df.stat.corr('Tumor Size (cm)', 'Oral Cancer (Diag0sis)'))

print("Correlation of 'Age' with 'Oral Cancer (Diag0sis)':")
display(df.stat.corr('Age', 'Oral Cancer (Diag0sis)'))

print("Correlation of 'TreatmentType_indexed' with 'Oral Cancer (Diag0sis)':")
display(df.stat.corr('TreatmentType_indexed', 'Oral Cancer (Diag0sis)'))
```

```
print("Correlation of 'Diet_indexed' with 'Oral Cancer (Diag0sis)':")
display(df.stat.corr('Diet_indexed', 'Oral Cancer (Diag0sis)'))
```

```
Correlation of 'Early Diag0sis' with 'Oral Cancer (Diag0sis)':
0.0007257300262715145
Correlation of 'Cancer Stage' with 'Oral Cancer (Diag0sis)':
0.8368414278511452
Correlation of 'Tumor Size (cm)' with 'Oral Cancer (Diag0sis)':
0.8637807170145679
Correlation of 'Age' with 'Oral Cancer (Diag0sis)':
0.0023082653836845883
Correlation of 'TreatmentType_indexed' with 'Oral Cancer (Diag0sis)':
0.7076404299785115
Correlation of 'Diet_indexed' with 'Oral Cancer (Diag0sis)':
-0.0018898896271861327
```