```python
import math
!pip install ucimlrepo
!pip install matplotlib seaborn
# !pip install sklearn Already preinstalled
!pip install alibi
!pip install shap
```

```
Requirement already satisfied: ucimlrepo in
./.venv/lib/python3.12/site-packages (0.0.7)
Requirement already satisfied: pandas>=1.0.0 in
./.venv/lib/python3.12/site-packages (from ucimlrepo) (2.2.2)
Requirement already satisfied: certifi>=2020.12.5 in
./.venv/lib/python3.12/site-packages (from ucimlrepo) (2024.6.2)
Requirement already satisfied: numpy>=1.26.0 in
./.venv/lib/python3.12/site-packages (from pandas>=1.0.0->ucimlrepo)
(1.26.4)
Requirement already satisfied: python-dateutil>=2.8.2 in
./.venv/lib/python3.12/site-packages (from pandas>=1.0.0->ucimlrepo)
(2.9.0.post0)
Requirement already satisfied: pytz>=2020.1 in
./.venv/lib/python3.12/site-packages (from pandas>=1.0.0->ucimlrepo)
(2024.1)
Requirement already satisfied: tzdata>=2022.7 in
./.venv/lib/python3.12/site-packages (from pandas>=1.0.0->ucimlrepo)
(2024.1)
Requirement already satisfied: six>=1.5 in
./.venv/lib/python3.12/site-packages (from python-dateutil>=2.8.2-
>pandas>=1.0.0->ucimlrepo) (1.16.0)

[notice] A new release of pip is available: 23.2.1 -> 24.1.1
[notice] To update, run: pip install --upgrade pip
Requirement already satisfied: matplotlib in
./.venv/lib/python3.12/site-packages (3.9.0)
Requirement already satisfied: seaborn in ./.venv/lib/python3.12/site-
packages (0.13.2)
Requirement already satisfied: contourpy>=1.0.1 in
./.venv/lib/python3.12/site-packages (from matplotlib) (1.2.1)
Requirement already satisfied: cycler>=0.10 in
./.venv/lib/python3.12/site-packages (from matplotlib) (0.12.1)
Requirement already satisfied: fonttools>=4.22.0 in
./.venv/lib/python3.12/site-packages (from matplotlib) (4.53.0)
Requirement already satisfied: kiwisolver>=1.3.1 in
./.venv/lib/python3.12/site-packages (from matplotlib) (1.4.5)
Requirement already satisfied: numpy>=1.23 in
./.venv/lib/python3.12/site-packages (from matplotlib) (1.26.4)
Requirement already satisfied: packaging>=20.0 in
./.venv/lib/python3.12/site-packages (from matplotlib) (24.1)
Requirement already satisfied: pillow>=8 in
./.venv/lib/python3.12/site-packages (from matplotlib) (10.3.0)
Requirement already satisfied: pyparsing>=2.3.1 in
```

```
./.venv/lib/python3.12/site-packages (from matplotlib) (3.1.2)
Requirement already satisfied: python-dateutil>=2.7 in
./.venv/lib/python3.12/site-packages (from matplotlib) (2.9.0.post0)
Requirement already satisfied: pandas>=1.2 in
./.venv/lib/python3.12/site-packages (from seaborn) (2.2.2)
Requirement already satisfied: pytz>=2020.1 in
./.venv/lib/python3.12/site-packages (from pandas>=1.2->seaborn)
(2024.1)
Requirement already satisfied: tzdata>=2022.7 in
./.venv/lib/python3.12/site-packages (from pandas>=1.2->seaborn)
(2024.1)
Requirement already satisfied: six>=1.5 in
./.venv/lib/python3.12/site-packages (from python-dateutil>=2.7-
>matplotlib) (1.16.0)

[notice] A new release of pip is available: 23.2.1 -> 24.1.1
[notice] To update, run: pip install --upgrade pip
Requirement already satisfied: alibi in ./.venv/lib/python3.12/site-
packages (0.9.6)
Requirement already satisfied: numpy<2.0.0,>=1.16.2 in
./.venv/lib/python3.12/site-packages (from alibi) (1.26.4)
Requirement already satisfied: pandas<3.0.0,>=1.0.0 in
./.venv/lib/python3.12/site-packages (from alibi) (2.2.2)
Requirement already satisfied: scikit-learn<2.0.0,>=1.0.0 in
./.venv/lib/python3.12/site-packages (from alibi) (1.5.0)
Requirement already satisfied: spacy[lookups]<4.0.0,>=2.0.0
in ./.venv/lib/python3.12/site-packages (from alibi) (3.7.5)
Requirement already satisfied: blis<0.8.0 in
./.venv/lib/python3.12/site-packages (from alibi) (0.7.11)
Requirement already satisfied: scikit-image<0.23,>=0.17.2 in
./.venv/lib/python3.12/site-packages (from alibi) (0.22.0)
Requirement already satisfied: requests<3.0.0,>=2.21.0 in
./.venv/lib/python3.12/site-packages (from alibi) (2.32.3)
Requirement already satisfied: Pillow<11.0,>=5.4.1 in
./.venv/lib/python3.12/site-packages (from alibi) (10.3.0)
Requirement already satisfied: attrs<24.0.0,>=19.2.0 in
./.venv/lib/python3.12/site-packages (from alibi) (23.2.0)
Requirement already satisfied: scipy<2.0.0,>=1.1.0 in
./.venv/lib/python3.12/site-packages (from alibi) (1.13.1)
Requirement already satisfied: matplotlib<4.0.0,>=3.0.0 in
./.venv/lib/python3.12/site-packages (from alibi) (3.9.0)
Requirement already satisfied: typing-extensions>=3.7.4.3 in
./.venv/lib/python3.12/site-packages (from alibi) (4.12.2)
Requirement already satisfied: dill<0.4.0,>=0.3.0 in
./.venv/lib/python3.12/site-packages (from alibi) (0.3.8)
Requirement already satisfied: transformers<5.0.0,>=4.7.0 in
./.venv/lib/python3.12/site-packages (from alibi) (4.41.2)
Requirement already satisfied: tqdm<5.0.0,>=4.28.1 in
./.venv/lib/python3.12/site-packages (from alibi) (4.66.4)
```

```
Requirement already satisfied: contourpy>=1.0.1 in
./.venv/lib/python3.12/site-packages (from matplotlib<4.0.0,>=3.0.0-
>alibi) (1.2.1)
Requirement already satisfied: cycler>=0.10 in
./.venv/lib/python3.12/site-packages (from matplotlib<4.0.0,>=3.0.0-
>alibi) (0.12.1)
Requirement already satisfied: fonttools>=4.22.0 in
./.venv/lib/python3.12/site-packages (from matplotlib<4.0.0,>=3.0.0-
>alibi) (4.53.0)
Requirement already satisfied: kiwisolver>=1.3.1 in
./.venv/lib/python3.12/site-packages (from matplotlib<4.0.0,>=3.0.0-
>alibi) (1.4.5)
Requirement already satisfied: packaging>=20.0 in
./.venv/lib/python3.12/site-packages (from matplotlib<4.0.0,>=3.0.0-
>alibi) (24.1)
Requirement already satisfied: pyparsing>=2.3.1 in
./.venv/lib/python3.12/site-packages (from matplotlib<4.0.0,>=3.0.0-
>alibi) (3.1.2)
Requirement already satisfied: python-dateutil>=2.7 in
./.venv/lib/python3.12/site-packages (from matplotlib<4.0.0,>=3.0.0-
>alibi) (2.9.0.post0)
Requirement already satisfied: pytz>=2020.1 in
./.venv/lib/python3.12/site-packages (from pandas<3.0.0,>=1.0.0-
>alibi) (2024.1)
Requirement already satisfied: tzdata>=2022.7 in
./.venv/lib/python3.12/site-packages (from pandas<3.0.0,>=1.0.0-
>alibi) (2024.1)
Requirement already satisfied: charset-normalizer<4,>=2 in
./.venv/lib/python3.12/site-packages (from requests<3.0.0,>=2.21.0-
>alibi) (3.3.2)
Requirement already satisfied: idna<4,>=2.5 in
./.venv/lib/python3.12/site-packages (from requests<3.0.0,>=2.21.0-
>alibi) (3.7)
Requirement already satisfied: urllib3<3,>=1.21.1 in
./.venv/lib/python3.12/site-packages (from requests<3.0.0,>=2.21.0-
>alibi) (2.2.2)
Requirement already satisfied: certifi>=2017.4.17 in
./.venv/lib/python3.12/site-packages (from requests<3.0.0,>=2.21.0-
>alibi) (2024.6.2)
Requirement already satisfied: networkx>=2.8 in
./.venv/lib/python3.12/site-packages (from scikit-image<0.23,>=0.17.2-
>alibi) (3.3)
Requirement already satisfied: imageio>=2.27 in
./.venv/lib/python3.12/site-packages (from scikit-image<0.23,>=0.17.2-
>alibi) (2.34.1)
Requirement already satisfied: tifffile>=2022.8.12 in
./.venv/lib/python3.12/site-packages (from scikit-image<0.23,>=0.17.2-
>alibi) (2024.6.18)
Requirement already satisfied: lazy_loader>=0.3 in
```

```
./.venv/lib/python3.12/site-packages (from scikit-image<0.23,>=0.17.2-
>alibi) (0.4)
Requirement already satisfied: joblib>=1.2.0 in
./.venv/lib/python3.12/site-packages (from scikit-learn<2.0.0,>=1.0.0-
>alibi) (1.4.2)
Requirement already satisfied: threadpoolctl>=3.1.0 in
./.venv/lib/python3.12/site-packages (from scikit-learn<2.0.0,>=1.0.0-
>alibi) (3.5.0)
Requirement already satisfied: spacy-legacy<3.1.0,>=3.0.11
in ./.venv/lib/python3.12/site-packages (from
spacy[lookups]<4.0.0,>=2.0.0->alibi) (3.0.12)
Requirement already satisfied: spacy-loggers<2.0.0,>=1.0.0
in ./.venv/lib/python3.12/site-packages (from
spacy[lookups]<4.0.0,>=2.0.0->alibi) (1.0.5)
Requirement already satisfied: murmurhash<1.1.0,>=0.28.0 in
./.venv/lib/python3.12/site-packages (from
spacy[lookups]<4.0.0,>=2.0.0->alibi) (1.0.10)
Requirement already satisfied: cymem<2.1.0,>=2.0.2 in
./.venv/lib/python3.12/site-packages (from
spacy[lookups]<4.0.0,>=2.0.0->alibi) (2.0.8)
Requirement already satisfied: preshed<3.1.0,>=3.0.2 in
./.venv/lib/python3.12/site-packages (from
spacy[lookups]<4.0.0,>=2.0.0->alibi) (3.0.9)
Requirement already satisfied: thinc<8.3.0,>=8.2.2 in
./.venv/lib/python3.12/site-packages (from
spacy[lookups]<4.0.0,>=2.0.0->alibi) (8.2.5)
Requirement already satisfied: wasabi<1.2.0,>=0.9.1 in
./.venv/lib/python3.12/site-packages (from
spacy[lookups]<4.0.0,>=2.0.0->alibi) (1.1.3)
Requirement already satisfied: srsly<3.0.0,>=2.4.3 in
./.venv/lib/python3.12/site-packages (from
spacy[lookups]<4.0.0,>=2.0.0->alibi) (2.4.8)
Requirement already satisfied: catalogue<2.1.0,>=2.0.6 in
./.venv/lib/python3.12/site-packages (from
spacy[lookups]<4.0.0,>=2.0.0->alibi) (2.0.10)
Requirement already satisfied: weasel<0.5.0,>=0.1.0 in
./.venv/lib/python3.12/site-packages (from
spacy[lookups]<4.0.0,>=2.0.0->alibi) (0.4.1)
Requirement already satisfied: typer<1.0.0,>=0.3.0 in
./.venv/lib/python3.12/site-packages (from
spacy[lookups]<4.0.0,>=2.0.0->alibi) (0.12.3)
Requirement already satisfied: pydantic!=1.8,!=1.8.1,<3.0.0,>=1.7.4 in
./.venv/lib/python3.12/site-packages (from
spacy[lookups]<4.0.0,>=2.0.0->alibi) (2.7.4)
Requirement already satisfied: jinja2 in ./.venv/lib/python3.12/site-
packages (from spacy[lookups]<4.0.0,>=2.0.0->alibi) (3.1.4)
Requirement already satisfied: setuptools in
./.venv/lib/python3.12/site-packages (from
spacy[lookups]<4.0.0,>=2.0.0->alibi) (70.1.0)
```

```
Requirement already satisfied: langcodes<4.0.0,>=3.2.0 in
./.venv/lib/python3.12/site-packages (from
spacy[lookups]<4.0.0,>=2.0.0->alibi) (3.4.0)
Requirement already satisfied: spacy-lookups-data<1.1.0,>=1.0.3
in ./.venv/lib/python3.12/site-packages (from
spacy[lookups]<4.0.0,>=2.0.0->alibi) (1.0.5)
Requirement already satisfied: filelock in
./.venv/lib/python3.12/site-packages (from transformers<5.0.0,>=4.7.0-
>alibi) (3.15.3)
Requirement already satisfied: huggingface-hub<1.0,>=0.23.0
in ./.venv/lib/python3.12/site-packages (from
transformers<5.0.0,>=4.7.0->alibi) (0.23.4)
Requirement already satisfied: pyyaml>=5.1 in
./.venv/lib/python3.12/site-packages (from transformers<5.0.0,>=4.7.0-
>alibi) (6.0.1)
Requirement already satisfied: regex!=2019.12.17 in
./.venv/lib/python3.12/site-packages (from transformers<5.0.0,>=4.7.0-
>alibi) (2024.5.15)
Requirement already satisfied: tokenizers<0.20,>=0.19 in
./.venv/lib/python3.12/site-packages (from transformers<5.0.0,>=4.7.0-
>alibi) (0.19.1)
Requirement already satisfied: safetensors>=0.4.1 in
./.venv/lib/python3.12/site-packages (from transformers<5.0.0,>=4.7.0-
>alibi) (0.4.3)
Requirement already satisfied: fsspec>=2023.5.0 in
./.venv/lib/python3.12/site-packages (from huggingface-
hub<1.0,>=0.23.0->transformers<5.0.0,>=4.7.0->alibi) (2024.6.0)
Requirement already satisfied: language-data>=1.2 in
./.venv/lib/python3.12/site-packages (from langcodes<4.0.0,>=3.2.0-
>spacy[lookups]<4.0.0,>=2.0.0->alibi) (1.2.0)
Requirement already satisfied: annotated-types>=0.4.0 in
./.venv/lib/python3.12/site-packages (from pydantic!=1.8,!
=1.8.1,<3.0.0,>=1.7.4->spacy[lookups]<4.0.0,>=2.0.0->alibi) (0.7.0)
Requirement already satisfied: pydantic-core==2.18.4 in
./.venv/lib/python3.12/site-packages (from pydantic!=1.8,!
=1.8.1,<3.0.0,>=1.7.4->spacy[lookups]<4.0.0,>=2.0.0->alibi) (2.18.4)
Requirement already satisfied: six>=1.5 in
./.venv/lib/python3.12/site-packages (from python-dateutil>=2.7-
>matplotlib<4.0.0,>=3.0.0->alibi) (1.16.0)
Requirement already satisfied: confection<1.0.0,>=0.0.1 in
./.venv/lib/python3.12/site-packages (from thinc<8.3.0,>=8.2.2-
>spacy[lookups]<4.0.0,>=2.0.0->alibi) (0.1.5)
Requirement already satisfied: click>=8.0.0 in
./.venv/lib/python3.12/site-packages (from typer<1.0.0,>=0.3.0-
>spacy[lookups]<4.0.0,>=2.0.0->alibi) (8.1.7)
Requirement already satisfied: shellingham>=1.3.0 in
./.venv/lib/python3.12/site-packages (from typer<1.0.0,>=0.3.0-
>spacy[lookups]<4.0.0,>=2.0.0->alibi) (1.5.4)
Requirement already satisfied: rich>=10.11.0 in
```

```
./.venv/lib/python3.12/site-packages (from typer<1.0.0,>=0.3.0-
>spacy[lookups]<4.0.0,>=2.0.0->alibi) (13.7.1)
Requirement already satisfied: cloudpathlib<1.0.0,>=0.7.0 in
./.venv/lib/python3.12/site-packages (from weasel<0.5.0,>=0.1.0-
>spacy[lookups]<4.0.0,>=2.0.0->alibi) (0.18.1)
Requirement already satisfied: smart-open<8.0.0,>=5.2.1 in
./.venv/lib/python3.12/site-packages (from weasel<0.5.0,>=0.1.0-
>spacy[lookups]<4.0.0,>=2.0.0->alibi) (7.0.4)
Requirement already satisfied: MarkupSafe>=2.0 in
./.venv/lib/python3.12/site-packages (from jinja2-
>spacy[lookups]<4.0.0,>=2.0.0->alibi) (2.1.5)
Requirement already satisfied: marisa-trie>=0.7.7 in
./.venv/lib/python3.12/site-packages (from language-data>=1.2-
>langcodes<4.0.0,>=3.2.0->spacy[lookups]<4.0.0,>=2.0.0->alibi) (1.2.0)

Requirement already satisfied: markdown-it-py>=2.2.0 in
./.venv/lib/python3.12/site-packages (from rich>=10.11.0-
>typer<1.0.0,>=0.3.0->spacy[lookups]<4.0.0,>=2.0.0->alibi) (3.0.0)
Requirement already satisfied: pygments<3.0.0,>=2.13.0 in
./.venv/lib/python3.12/site-packages (from rich>=10.11.0-
>typer<1.0.0,>=0.3.0->spacy[lookups]<4.0.0,>=2.0.0->alibi) (2.18.0)
Requirement already satisfied: wrapt in ./.venv/lib/python3.12/site-
packages (from smart-open<8.0.0,>=5.2.1->weasel<0.5.0,>=0.1.0-
>spacy[lookups]<4.0.0,>=2.0.0->alibi) (1.16.0)
Requirement already satisfied: mdurl~=0.1 in
./.venv/lib/python3.12/site-packages (from markdown-it-py>=2.2.0-
>rich>=10.11.0->typer<1.0.0,>=0.3.0->spacy[lookups]<4.0.0,>=2.0.0-
>alibi) (0.1.2)

[notice] A new release of pip is available: 23.2.1 -> 24.1.1
[notice] To update, run: pip install --upgrade pip
Requirement already satisfied: shap in ./.venv/lib/python3.12/site-
packages (0.45.1)
Requirement already satisfied: numpy in ./.venv/lib/python3.12/site-
packages (from shap) (1.26.4)
Requirement already satisfied: scipy in ./.venv/lib/python3.12/site-
packages (from shap) (1.13.1)
Requirement already satisfied: scikit-learn in
./.venv/lib/python3.12/site-packages (from shap) (1.5.0)
Requirement already satisfied: pandas in ./.venv/lib/python3.12/site-
packages (from shap) (2.2.2)
Requirement already satisfied: tqdm>=4.27.0 in
./.venv/lib/python3.12/site-packages (from shap) (4.66.4)
Requirement already satisfied: packaging>20.9 in
./.venv/lib/python3.12/site-packages (from shap) (24.1)
Requirement already satisfied: slicer==0.0.8 in
./.venv/lib/python3.12/site-packages (from shap) (0.0.8)
Requirement already satisfied: numba in ./.venv/lib/python3.12/site-
packages (from shap) (0.60.0)
```

```
Requirement already satisfied: cloudpickle in
./.venv/lib/python3.12/site-packages (from shap) (3.0.0)
Requirement already satisfied: llvmlite<0.44,>=0.43.0dev0 in
./.venv/lib/python3.12/site-packages (from numba->shap) (0.43.0)
Requirement already satisfied: python-dateutil>=2.8.2 in
./.venv/lib/python3.12/site-packages (from pandas->shap) (2.9.0.post0)

Requirement already satisfied: pytz>=2020.1 in
./.venv/lib/python3.12/site-packages (from pandas->shap) (2024.1)
Requirement already satisfied: tzdata>=2022.7 in
./.venv/lib/python3.12/site-packages (from pandas->shap) (2024.1)
Requirement already satisfied: joblib>=1.2.0 in
./.venv/lib/python3.12/site-packages (from scikit-learn->shap) (1.4.2)

Requirement already satisfied: threadpoolctl>=3.1.0 in
./.venv/lib/python3.12/site-packages (from scikit-learn->shap) (3.5.0)

Requirement already satisfied: six>=1.5 in
./.venv/lib/python3.12/site-packages (from python-dateutil>=2.8.2-
>pandas->shap) (1.16.0)

[notice] A new release of pip is available: 23.2.1 -> 24.1.1
[notice] To update, run: pip install --upgrade pip
```

Cervical cancer risk factor prediction

We consider the cervical cancer dataset contains indicators and risk factors for predicting whether a woman will get cervical cancer. The features include demographic data (such as age), lifestyle, and medical history. The objectives are: 1) to train a risk predictor (i.e., binary classifier) of cervical cancer, given the input features. 2) to identify the most important risk factors.

The features are:

1. Age in years
2. Number of sexual partners
3. First sexual intercourse (age in years)
4. Number of pregnancies
5. Smoking (in years)
6. Hormonal contraceptives (in years)
7. Number of years with an intrauterine device (IUD)
8. Has patient ever had a sexually transmitted disease (STD) yes or no
9. Number of STD diagnoses
10. Time since first STD diagnosis
11. Time since last STD diagnosis
12. hPV diagnostic

The target is the biopsy results: “Healthy” or “Cancer”.

Download dataset, partition into train/test

```python
from ucimlrepo import fetch_ucirepo

cervical_cancer = fetch_ucirepo(name='Cervical Cancer')

X = cervical_cancer.data.features
y = X['Biopsy']  # Ground truth diagnosis: Biopsy result

# access metadata
print('Number of instances', cervical_cancer.metadata.num_instances)
print('Summary', cervical_cancer.metadata.additional_info.summary)

# access variable info in tabular format
print('All variables', cervical_cancer.variables)

# Retain only a fraction of the features:

included_features = ['Age',
                     'Number of sexual partners',
                     'First sexual intercourse',
                     'Num of pregnancies',
                     'Smokes (years)',
                     'Hormonal Contraceptives (years)',
                     'IUD (years)',
                     'STDs',
                     'STDs: Number of diagnosis',
                     'STDs: Time since first diagnosis',
                     'STDs: Time since last diagnosis',
                     'Dx:HPV']

X = X[included_features]
```

```
Number of instances 858
Summary The dataset was collected at 'Hospital Universitario de
Caracas' in Caracas, Venezuela. The dataset comprises demographic
information, habits, and historic medical records of 858 patients.
Several patients decided not to answer some of the questions because
of privacy concerns (missing values).
All variables                                   name     role
type demographic  \
0                                  Age  Feature     Integer
Age
1            Number of sexual partners  Feature  Continuous
Other
2             First sexual intercourse  Feature  Continuous
None
3                   Num of pregnancies  Feature  Continuous
None
4                               Smokes  Feature  Continuous
```

```
None
5                         Smokes (years)  Feature  Continuous
None
6                    Smokes (packs/year)  Feature  Continuous
None
7                Hormonal Contraceptives  Feature  Continuous
None
8        Hormonal Contraceptives (years)  Feature  Continuous
None
9                                    IUD  Feature  Continuous
None
10                           IUD (years)  Feature  Continuous
None
11                                  STDs  Feature  Continuous
None
12                         STDs (number)  Feature  Continuous
None
13                   STDs:condylomatosis  Feature  Continuous
None
14          STDs:cervical condylomatosis  Feature  Continuous
None
15           STDs:vaginal condylomatosis  Feature  Continuous
None
16    STDs:vulvo-perineal condylomatosis  Feature  Continuous
None
17                         STDs:syphilis  Feature  Continuous
None
18      STDs:pelvic inflammatory disease  Feature  Continuous
None
19                   STDs:genital herpes  Feature  Continuous
None
20            STDs:molluscum contagiosum  Feature  Continuous
None
21                             STDs:AIDS  Feature  Continuous
None
22                              STDs:HIV  Feature  Continuous
None
23                      STDs:Hepatitis B  Feature  Continuous
None
24                              STDs:HPV  Feature  Continuous
None
25             STDs: Number of diagnosis  Feature     Integer
None
26      STDs: Time since first diagnosis  Feature  Continuous
None
27       STDs: Time since last diagnosis  Feature  Continuous
None
28                             Dx:Cancer  Feature     Integer
None
```

```
29                              Dx:CIN  Feature     Integer
None
30                              Dx:HPV  Feature     Integer
None
31                                  Dx  Feature     Integer
None
32                          Hinselmann  Feature     Integer
None
33                            Schiller  Feature     Integer
None
34                            Citology  Feature     Integer
None
35                              Biopsy  Feature     Integer
None

   description units missing_values
0         None  None             no
1         None  None            yes
2         None  None            yes
3         None  None            yes
4         None  None            yes
5         None  None            yes
6         None  None            yes
7         None  None            yes
8         None  None            yes
9         None  None            yes
10        None  None            yes
11        None  None            yes
12        None  None            yes
13        None  None            yes
14        None  None            yes
15        None  None            yes
16        None  None            yes
17        None  None            yes
18        None  None            yes
19        None  None            yes
20        None  None            yes
21        None  None            yes
22        None  None            yes
23        None  None            yes
24        None  None            yes
25        None  None             no
26        None  None            yes
27        None  None            yes
28        None  None             no
29        None  None             no
30        None  None             no
31        None  None             no
32        None  None             no
```

```
33        None  None             no
34        None  None             no
35        None  None             no
```

Data cleaning.

Here, we display summary statistics and identify potential issues. Here, we find no aberrant values, but some features are missing for some instances; we replace missing feature values by the median of the dataset.

```
import pandas as pd
from sklearn.impute import SimpleImputer

print('Before imputation')
print(X.describe())

imputer = SimpleImputer(strategy="median")
X = pd.DataFrame(imputer.fit_transform(X), columns=X.columns)
print('after imputation')
print(X.describe())
```

```
Before imputation
              Age  Number of sexual partners  First sexual intercourse  
\
count  858.000000                 832.000000                851.000000   

mean    26.820513                   2.527644                 16.995300   

std      8.497948                   1.667760                  2.803355   

min     13.000000                   1.000000                 10.000000   

25%     20.000000                   2.000000                 15.000000   

50%     25.000000                   2.000000                 17.000000   

75%     32.000000                   3.000000                 18.000000   

max     84.000000                  28.000000                 32.000000   


       Num of pregnancies  Smokes (years)  Hormonal Contraceptives 
(years)  \
count          802.000000      845.000000                        
750.000000   
mean             2.275561        1.219721                        
2.256419   
std              1.447414        4.089017                        
3.764254   
min              0.000000        0.000000                        
```

```
0.000000
25%              1.000000        0.000000
0.000000
50%              2.000000        0.000000
0.500000
75%              3.000000        0.000000
3.000000
max             11.000000       37.000000
30.000000

       IUD (years)        STDs  STDs: Number of diagnosis  \
count   741.000000  753.000000                 858.000000
mean      0.514804    0.104914                   0.087413
std       1.943089    0.306646                   0.302545
min       0.000000    0.000000                   0.000000
25%       0.000000    0.000000                   0.000000
50%       0.000000    0.000000                   0.000000
75%       0.000000    0.000000                   0.000000
max      19.000000    1.000000                   3.000000

       STDs: Time since first diagnosis  STDs: Time since last
diagnosis  \
count                         71.000000
71.000000
mean                           6.140845
5.816901
std                            5.895024
5.755271
min                            1.000000
1.000000
25%                            2.000000
2.000000
50%                            4.000000
3.000000
75%                            8.000000
7.500000
max                           22.000000
22.000000

           Dx:HPV
count  858.000000
mean     0.020979
std      0.143398
min      0.000000
25%      0.000000
50%      0.000000
75%      0.000000
max      1.000000
after imputation
              Age  Number of sexual partners  First sexual intercourse
```

```
\
count  858.000000                          858.000000                          858.000000

mean    26.820513                            2.511655                           16.995338

std      8.497948                            1.644759                            2.791883

min     13.000000                            1.000000                           10.000000

25%     20.000000                            2.000000                           15.000000

50%     25.000000                            2.000000                           17.000000

75%     32.000000                            3.000000                           18.000000

max     84.000000                           28.000000                           32.000000


       Num of pregnancies  Smokes (years)  Hormonal Contraceptives
(years)  \
count          858.000000      858.000000
858.000000
mean             2.257576        1.201241
2.035331
std              1.400981        4.060623
3.567040
min              0.000000        0.000000
0.000000
25%              1.000000        0.000000
0.000000
50%              2.000000        0.000000
0.500000
75%              3.000000        0.000000
2.000000
max             11.000000       37.000000
30.000000

       IUD (years)        STDs  STDs: Number of diagnosis  \
count   858.000000  858.000000                 858.000000
mean      0.444604    0.092075                   0.087413
std       1.814218    0.289300                   0.302545
min       0.000000    0.000000                   0.000000
25%       0.000000    0.000000                   0.000000
50%       0.000000    0.000000                   0.000000
75%       0.000000    0.000000                   0.000000
max      19.000000    1.000000                   3.000000

       STDs: Time since first diagnosis  STDs: Time since last
diagnosis  \
count                        858.000000
```

```
858.000000
mean                           4.177156
3.233100
std                            1.785156
1.818927
min                            1.000000
1.000000
25%                            4.000000
3.000000
50%                            4.000000
3.000000
75%                            4.000000
3.000000
max                           22.000000
22.000000

           Dx:HPV
count  858.000000
mean     0.020979
std      0.143398
min      0.000000
25%      0.000000
50%      0.000000
75%      0.000000
max      1.000000
```

Data Partition

```
from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(X, y,
random_state=0, stratify=y)

y_train = y_train.astype(int)
y_test = y_test.astype(int)

print(X_train.shape, X_test.shape, y_train.shape, y_test.shape)

(643, 12) (215, 12) (643,) (215,)
```

B. Logistic Regression

```
from sklearn import set_config

set_config(display='diagram')

from sklearn.linear_model import LogisticRegressionCV
```

1. Train an L2-regularized logistic regression model on the training data set with an optimal using the AUCROC metric. You can use the sklearn.linear_model.LogisticRegressionCV function to automatically adjust the value of the L2 penalty.

```
from sklearn.preprocessing import StandardScaler
from sklearn.pipeline import Pipeline

logistic_model = Pipeline(
    [
        ('Scaler', StandardScaler()),
        ('classifier', LogisticRegressionCV(
            scoring='roc_auc', max_iter=1000
        ))
    ]
)

logistic_model.fit(X_train, y_train)
```

```
Pipeline(steps=[('Scaler', StandardScaler()),
                ('classifier',
                 LogisticRegressionCV(max_iter=1000,
scoring='roc_auc'))])
```

2. Report the classification performance on the train and test set (accuracy, AUCROC, and negative log-likelihood).

```
from sklearn.metrics import accuracy_score, log_loss,
classification_report, roc_auc_score


def report_performance(
        model,
        X_train=X_train, X_test=X_test,
        y_train=y_train, y_test=y_test,
        with_classification_report=False
):
    for name, X, y in zip(
            ['Train', 'Test'],
            [X_train, X_test],
            [y_train, y_test],
    ):
        print(name)
        y_pred = model.predict(X)

        if with_classification_report:
            # show y value counts
            print('Value counts:')
            print(y.value_counts())
            print(pd.Series(y_pred).value_counts())

            print('Classification report:')
            print(classification_report(y, y_pred,
zero_division=np.nan))
        else:
            accuracy = accuracy_score(y, y_pred) * 100
            print(f'Accuracy: {accuracy:.1f}%')

        aucroc = roc_auc_score(y, model.predict_proba(X)[:, 1])
        print(f'AUCROC: {aucroc:.3f}')
        neg_log_likelihood = log_loss(y, model.predict_proba(X))
        print(f'Negative log-likelihood: {neg_log_likelihood:.3f}')

report_performance(logistic_model)
```

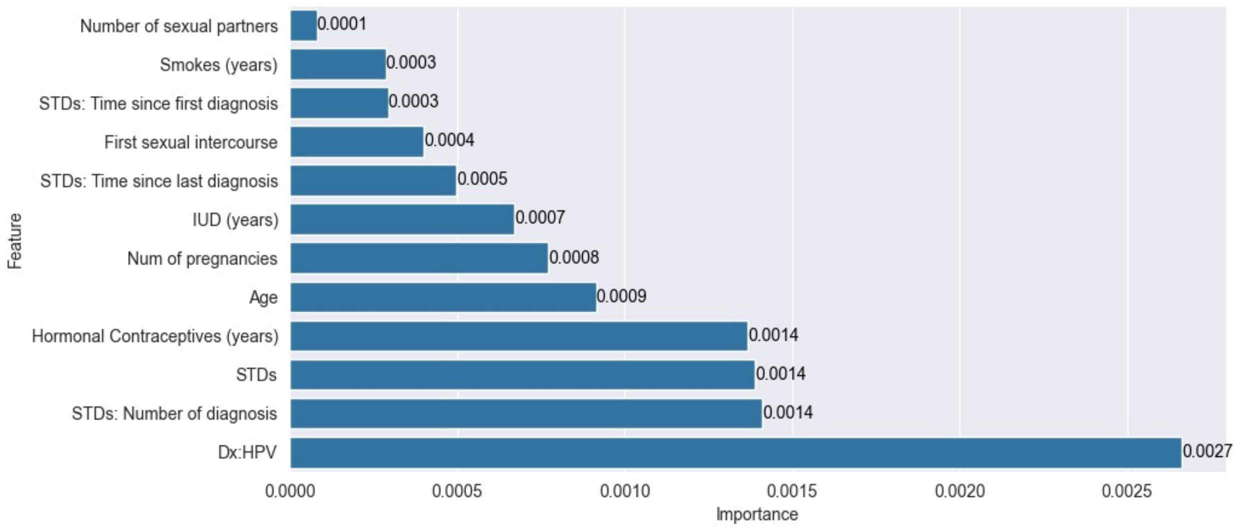
```
Train
Accuracy: 93.6%
AUCROC: 0.654
Negative log-likelihood: 0.237
Test
Accuracy: 93.5%
AUCROC: 0.725
Negative log-likelihood: 0.241
```

3. Calculate the feature importance (defined as the standard deviation of the feature effects) and visualize them as a bar plot.

4. What are the most and least important features?

```
from sklearn.pipeline import Pipeline
def calculate_feature_effects(model, X_train=X_train):
    if isinstance(model, Pipeline):
        preprocessor = model[:-1]
        model = model[-1]
        transformed_features = preprocessor.transform(X_train)
        transformed_feature_names =
preprocessor.get_feature_names_out()
    else:
        transformed_features = X_train.to_numpy()
        transformed_feature_names = X_train.columns

    transformed_features_effects = transformed_features * model.coef_

    feature_effects = np.array([
        transformed_features_effects[:,
        [
            i
            for i, transformed_feature_name
            in enumerate(transformed_feature_names)
            if (

transformed_feature_name.startswith(f"Bspline__{feature_name}_sp_")
                or
transformed_feature_name.startswith(f"remainder__{feature_name}")
                or  feature_name == transformed_feature_name
            )
        ]
        ].sum(-1)
        for feature_name in X_train.columns
    ]).T

    return pd.DataFrame(feature_effects, columns=X_train.columns)

import seaborn as sns

def plot_feature_importance(feature_effects):
    feature_importance = feature_effects.std()
    feature_importance = pd.Series(feature_importance,
index=X_train.columns).sort_values()

    print(f'Most important: {feature_importance.idxmax()}
({feature_importance.max():.4f})')
    print(f'Least important: {feature_importance.idxmin()}
({feature_importance.min():.4f})')

    plt.figure(figsize=(10, 5))
    sns.barplot(x=feature_importance, y=feature_importance.index)
    # show values
    for i, v in enumerate(feature_importance):
        plt.text(v, i, f'{v:.4f}', color='black', va='center')
    plt.xlabel('Importance')
    plt.ylabel('Feature')
    plt.show()

plot_feature_importance(calculate_feature_effects(logistic_model))
```

```
Most important: Dx:HPV (0.0027)
Least important: Number of sexual partners (0.0001)
```



C. Generalized Additive Model

The log-odds ratio of having cervical cancer is not expected to be linearly related to the numerical features. Hence, a Generalized Additive Model could be more accurate.

1. Build and train a Generalized Additive Model, where the numerical features have a trainable, non-linear effect, and the others have a linear effect.

Your model should be implemented as a scikit-learn Pipeline (sklearn.pipeline.Pipeline), where numerical features are transformed via B-splines (cubic order, 5 knots, constant extrapolation; use sklearn.preprocessing.SplineTransformer) while other features are not transformed (use sklearn.compose.ColumnTransformer), followed by an L2-regularized logistic regression model (use sklearn.linear_model.LogisticRegressionCV).

```
def get_numerical_features(data):
    return [
        feature
        for feature in data.columns
        if (
                data[feature].dtype in ['int64', 'float64']
                and (not set(data[feature].unique()) == {0, 1})
        )
    ]


numerical_features = get_numerical_features(X)

from sklearn.preprocessing import StandardScaler
from sklearn.preprocessing import SplineTransformer
from sklearn.compose import ColumnTransformer
from sklearn.linear_model import LogisticRegressionCV

gam_model = Pipeline(
    [
        ('preprocessor', ColumnTransformer(
            [
                ('Bspline', SplineTransformer(
                    n_knots=5, extrapolation='constant', order='C'  # Explicit defaults
                ), numerical_features),
            ],
            remainder='passthrough',
            force_int_remainder_cols=False
        )),
        ('Scaler', StandardScaler()),  # not asked specifically but added for better practice
        ('classifier', LogisticRegressionCV(
            max_iter=1000,
            scoring='roc_auc'  # TODO: with or without this?
        ))
    ]
)

gam_model.fit(X_train, y_train)
```

```
Pipeline(steps=[('preprocessor',
                 ColumnTransformer(force_int_remainder_cols=False,
                                   remainder='passthrough',
                                   transformers=[('Bspline',
                                                  SplineTransformer(),
                                                  ['Age',
                                                   'Number of sexual partners',
                                                   'First sexual intercourse',
                                                   'Num of pregnancies',
                                                   'Smokes (years)',
                                                   'Hormonal Contraceptives '
                                                   '(years)',
                                                   'IUD (years)',
                                                   'STDs: Number of diagnosis',
                                                   'STDs: Time since first '
                                                   'diagnosis',
                                                   'STDs: Time since last '
                                                   'diagnosis'])])),
                ('Scaler', StandardScaler()),
                ('classifier',
                 LogisticRegressionCV(max_iter=1000,
scoring='roc_auc'))])
```

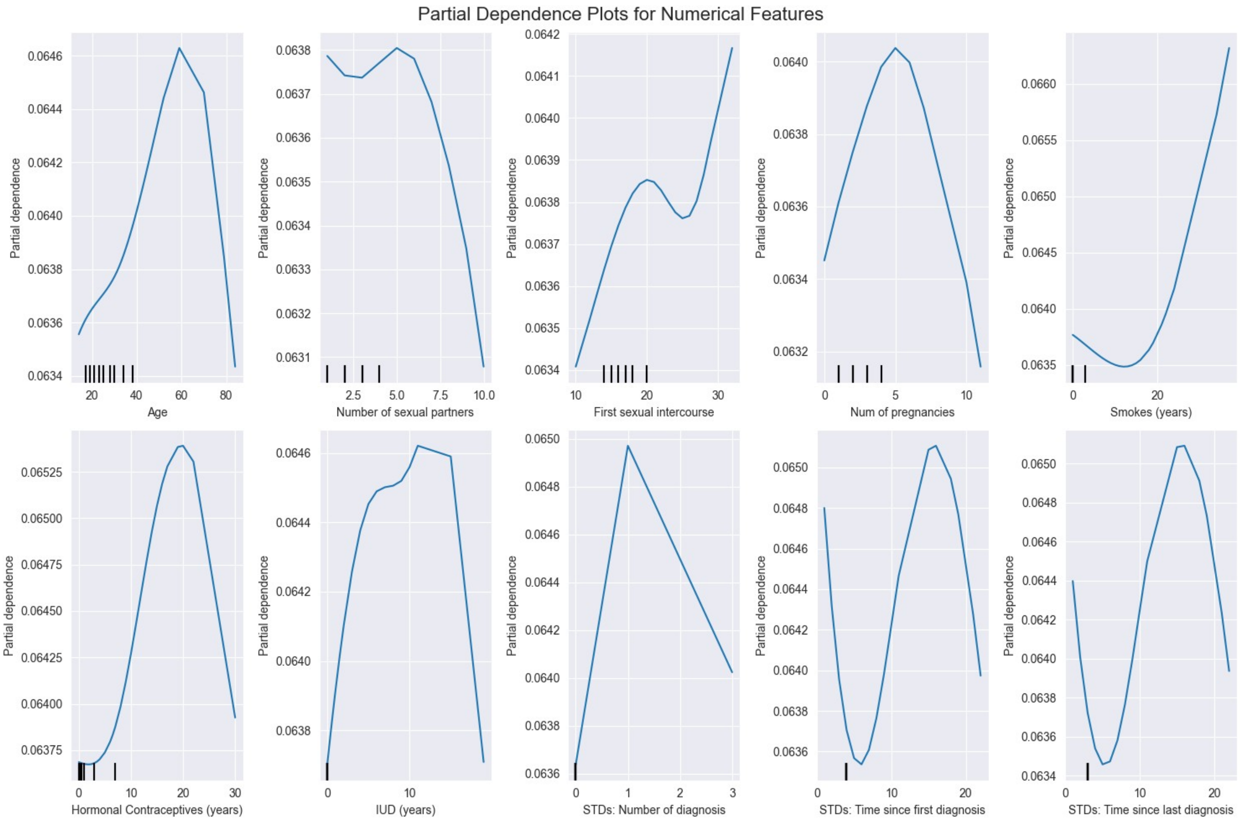
2. Report the classification performance on the train and test set (accuracy, AUROC, and negative log-likelihood) and compare with the performance of the logistic regression model.

```
report_performance(gam_model)

Train
Accuracy: 93.6%
AUCROC: 0.678
Negative log-likelihood: 0.236
Test
Accuracy: 93.5%
AUCROC: 0.676
Negative log-likelihood: 0.240
```

3. Visualize the learnt non-linearity (use sklearn.inspection.partial_dependence).

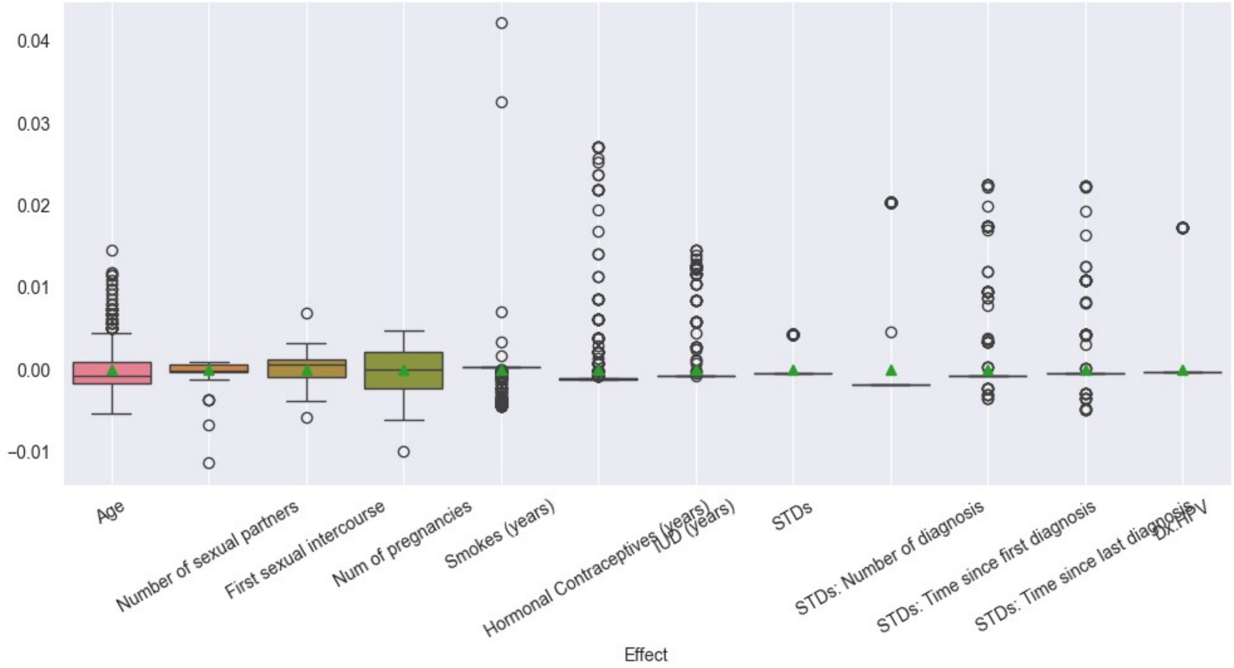
```
from sklearn.inspection import PartialDependenceDisplay
import math


def plot_partial_dependence(model, X=X_train, cols=5,
allow_different_y_axis_range=False,
                            numerical_features=numerical_features):
    number_of_rows = math.ceil(len(numerical_features) / cols)
    fig, ax = plt.subplots(number_of_rows, cols, figsize=(15, 5 *
number_of_rows))

    ax = ax.flatten()

    if allow_different_y_axis_range:
        # Using for allows to have different y-axis range for each
feature
        for i, feature in enumerate(numerical_features):
            PartialDependenceDisplay.from_estimator(
                model, X, [feature], ax=ax[i], percentiles=(0.01,
0.99)
            )
    else:
        PartialDependenceDisplay.from_estimator(
            model, X, numerical_features, percentiles=(0.01, 0.99),
ax=ax, n_cols=cols
```

```
        )

    plt.tight_layout()
    plt.subplots_adjust(top=0.95)
    plt.suptitle('Partial Dependence Plots for Numerical Features',
fontsize=16)
    plt.show()


plot_partial_dependence(gam_model, allow_different_y_axis_range=True)
```



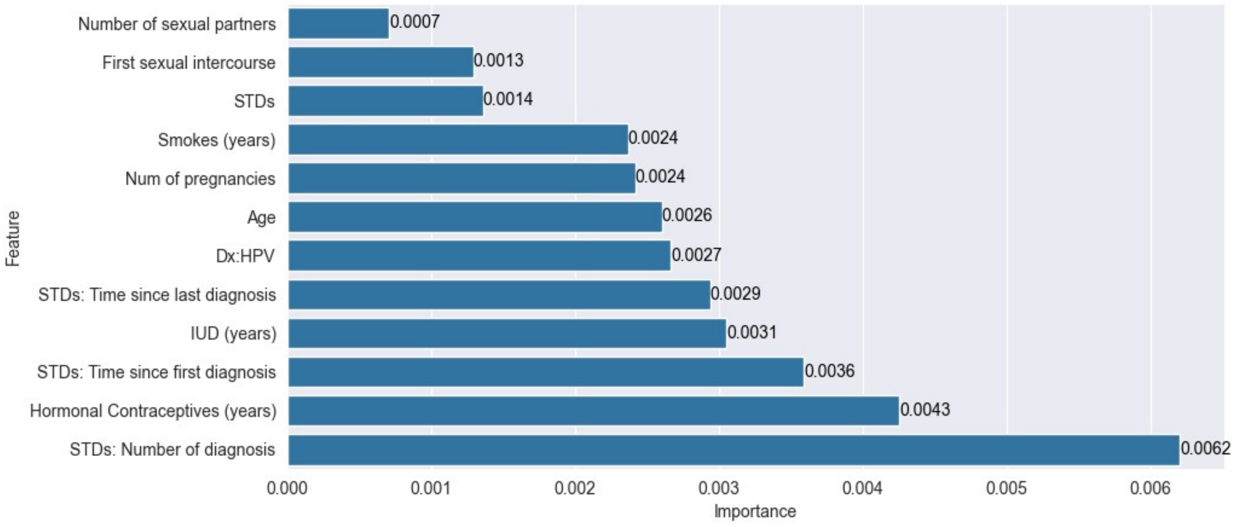
4. Calculate the feature effects (you can adapt the code snippet from the tutorial).

```
def plot_feature_effects(feature_effects):
    fig, ax = plt.subplots(figsize=(12, 5))
    sns.boxplot(data=feature_effects, showmeans=True)
    plt.xticks(rotation=30)
    plt.xlabel('Effect')
    plt.grid()

plot_feature_effects(calculate_feature_effects(gam_model))
```



5. What are the most and least important features? Comment on the differences.

```
plot_feature_importance(calculate_feature_effects(gam_model))

Most important: STDs: Number of diagnosis (0.0062)
Least important: Number of sexual partners (0.0007)
```



When we examine the disagreements between the two models importance plots, we can see that while `DX:HPV` was a high importance feature in the logistic regression, it's not as high in the gam model. While other features for example `STDs: Time since first diagnosis` and `STDs: Time since last diagnosis` are significantly higher.

The PDP plots explains that phenomena since we see that these "Time Since ..." features have strong non-linear effects, that the GAM model is able to capture.

D. Black-box Classifier Model

A model taking into account interactions between features could yield better predictive performance at the cost of reduced interpretability. We will build such a black-box model and use a-posteriori interpretation/explanation methods.

1. Train a Random Forest classifier (sklearn.ensemble.RandomForestClassifier) with n_estimators=200 trees.

Optimize the min_samples_leaf hyperparameter (from 1 to 100) using cross-validation over the train set (use sklearn.model_selection.GridSearchCV and sklearn.model_selection.KFold).

```
import numpy as np
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import GridSearchCV, KFold
seed = 42
min_samples_leaf = np.linspace(1, 100, 100, dtype=int)
forest = GridSearchCV(
    RandomForestClassifier(random_state=seed, n_estimators=200),
    param_grid={'min_samples_leaf': min_samples_leaf},
    cv=KFold(random_state=seed, n_splits=5, shuffle=True),
    scoring='roc_auc',
    n_jobs=-1
).fit(X_train, y_train)
print(forest.best_params_)
```

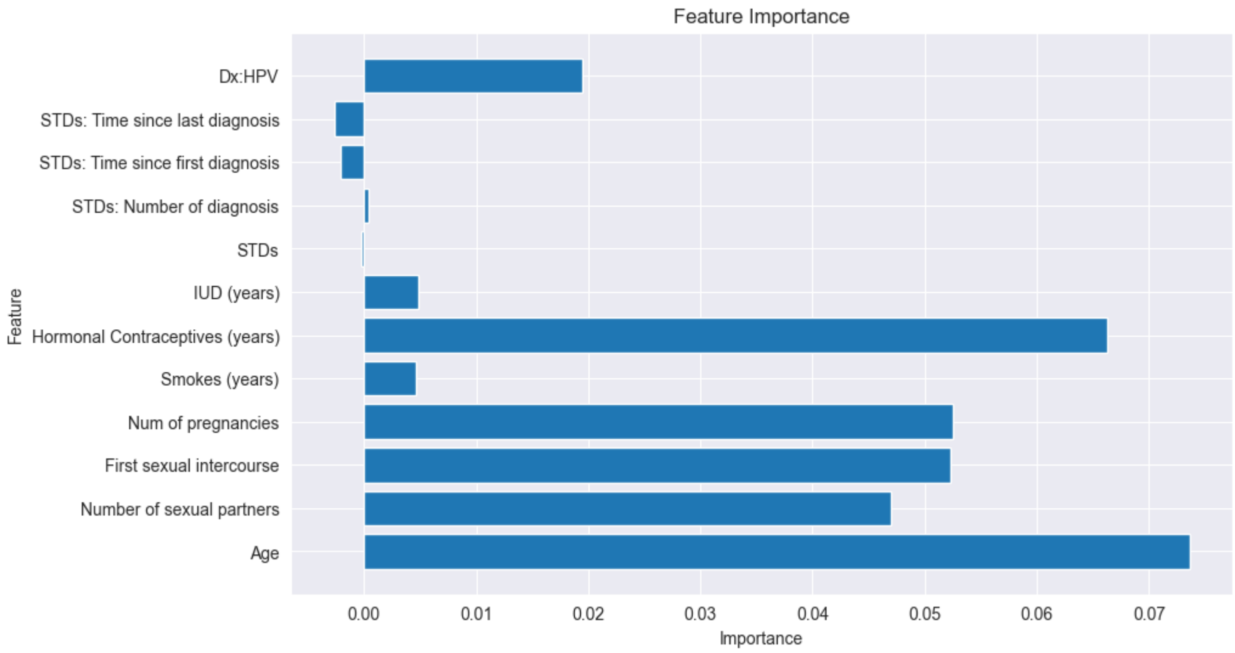
```
{'min_samples_leaf': 5}
```

```
best_params = {'min_samples_leaf': 5}
forest = RandomForestClassifier(random_state=42, n_estimators=200,
min_samples_leaf=best_params['min_samples_leaf']).fit(X_train,
y_train)
report_performance(forest)
```

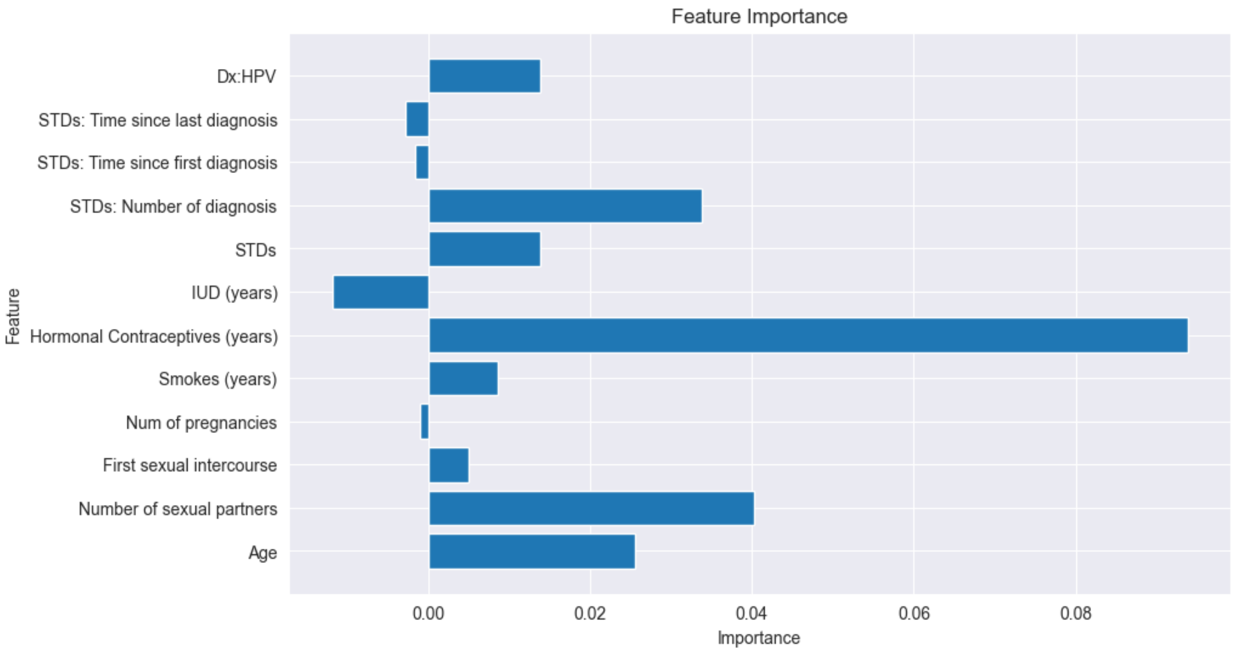
```
Train
Accuracy: 93.6%
AUCROC: 0.951
Negative log-likelihood: 0.161
Test
Accuracy: 93.5%
AUCROC: 0.730
Negative log-likelihood: 0.222
```

2. Determine the feature importance using the permutation importance metric on the train and test set and visualize them (sklearn.inspection.permutation_importance).

```python
from sklearn.inspection import permutation_importance
# train_importance = permutation_importance(forest, X_train, y_train)
train_importance = permutation_importance(forest, X_train, y_train,
scoring='roc_auc', random_state=seed)
import matplotlib.pyplot as plt

feature_names = X_train.columns
train_importance = train_importance.importances_mean

plt.figure(figsize=(10, 6))
plt.barh(feature_names, train_importance)
plt.xlabel('Importance')
plt.ylabel('Feature')
plt.title('Feature Importance')
plt.show()
```



```python
from sklearn.inspection import permutation_importance
test_importance = permutation_importance(forest, X_test, y_test,
scoring='roc_auc', random_state=seed)
import matplotlib.pyplot as plt

feature_names = X_test.columns
test_importance = test_importance.importances_mean

plt.figure(figsize=(10, 6))
plt.barh(feature_names, test_importance)
plt.xlabel('Importance')
plt.ylabel('Feature')
plt.title('Feature Importance')
plt.show()
```



3. Conclude on the most important features, and on the features for which overfitting occurs. Are the conclusions different from previously? Why?

Most important: `Hormonal Contraceptives`, `STDs: Number of diagnosis`, `Number of sexual partners` (Strongest importance on test set)

Overfitting: `Age`, `Number of pregnancies`, `First sexual intercourse` (Significantly stronger effect on training set compared to test set)

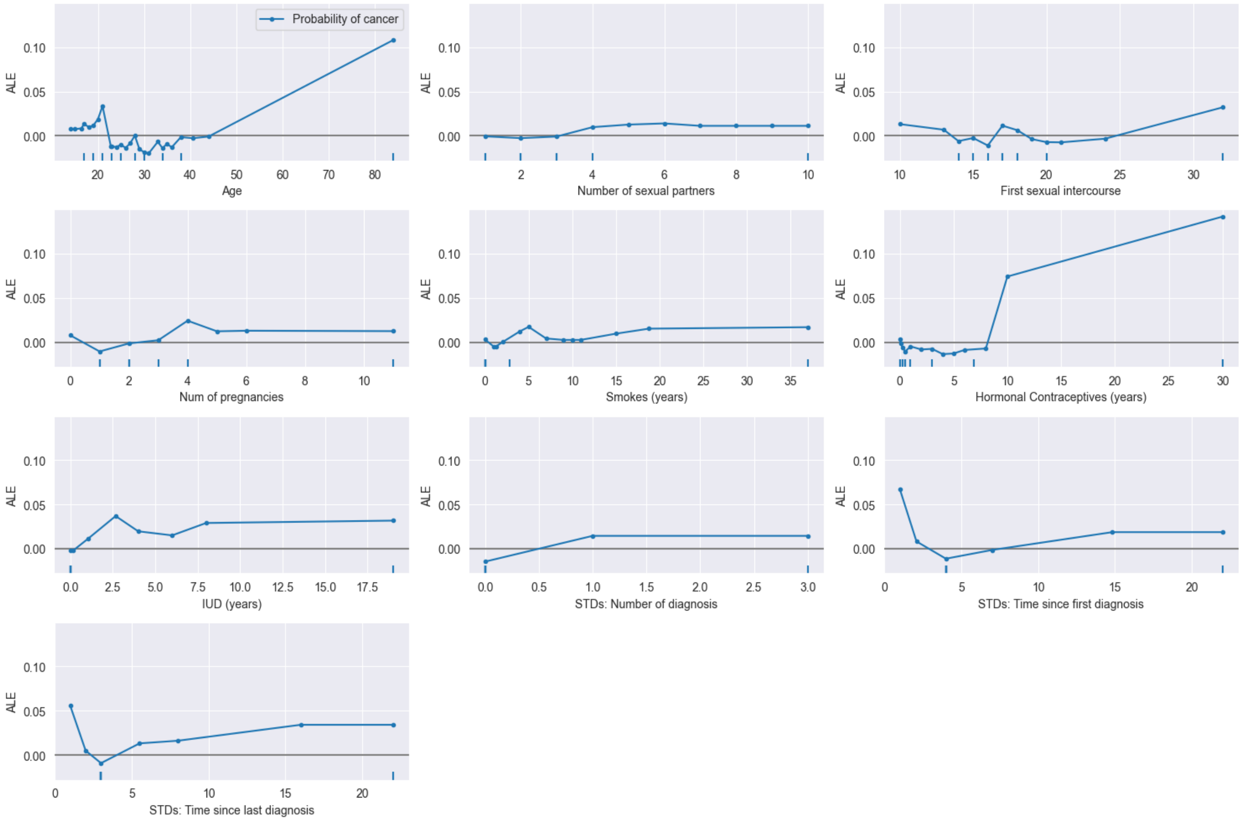
4. Using the alibi package (Accumulated Local Effects (ALE)), plot the Accumulated Local Effects for all numerical features and report them. How do the ALE plots compare with the partial dependence plots of the GAM model?

```python
def get_numerical_features(data):
    return [
        feature
        for feature in data.columns
        if (
                data[feature].dtype in ['int64', 'float64']
                and (not set(data[feature].unique()) == {0, 1})
        )
    ]

from alibi.explainers import ALE, plot_ale
prob = lambda X: forest.predict_proba(X)[:, 1] # the probability of
positive

ale = ALE(prob, feature_names=X_train.columns,
target_names=['Probability of cancer'])
exp = ale.explain(X_train.values)
fig, ax = plt.subplots(figsize=(15, 10))  # Increase figure size
plot_ale(exp, features=get_numerical_features(X_train), ax=ax)
```

```
/Users/nirendy/school-repo/biology-hw1/.venv/lib/python3.12/site-
packages/tqdm/auto.py:21: TqdmWarning: IProgress not found. Please
update jupyter and ipywidgets. See
https://ipywidgets.readthedocs.io/en/stable/user_install.html
  from .autonotebook import tqdm as notebook_tqdm
/Users/nirendy/school-repo/biology-hw1/.venv/lib/python3.12/site-
packages/sklearn/base.py:493: UserWarning: X does not have valid
feature names, but RandomForestClassifier was fitted with feature
names
  warnings.warn(
/Users/nirendy/school-repo/biology-hw1/.venv/lib/python3.12/site-
packages/sklearn/base.py:493: UserWarning: X does not have valid
feature names, but RandomForestClassifier was fitted with feature
names
  warnings.warn(
/Users/nirendy/school-repo/biology-hw1/.venv/lib/python3.12/site-
packages/sklearn/base.py:493: UserWarning: X does not have valid
feature names, but RandomForestClassifier was fitted with feature
names
  warnings.warn(
/Users/nirendy/school-repo/biology-hw1/.venv/lib/python3.12/site-
packages/sklearn/base.py:493: UserWarning: X does not have valid
feature names, but RandomForestClassifier was fitted with feature
names
  warnings.warn(
/Users/nirendy/school-repo/biology-hw1/.venv/lib/python3.12/site-
packages/sklearn/base.py:493: UserWarning: X does not have valid
feature names, but RandomForestClassifier was fitted with feature
names
  warnings.warn(
/Users/nirendy/school-repo/biology-hw1/.venv/lib/python3.12/site-
packages/sklearn/base.py:493: UserWarning: X does not have valid
feature names, but RandomForestClassifier was fitted with feature
names
  warnings.warn(
/Users/nirendy/school-repo/biology-hw1/.venv/lib/python3.12/site-
```

```
packages/sklearn/base.py:493: UserWarning: X does not have valid
feature names, but RandomForestClassifier was fitted with feature
names
  warnings.warn(
/Users/nirendy/school-repo/biology-hw1/.venv/lib/python3.12/site-
packages/sklearn/base.py:493: UserWarning: X does not have valid
feature names, but RandomForestClassifier was fitted with feature
names
  warnings.warn(
/Users/nirendy/school-repo/biology-hw1/.venv/lib/python3.12/site-
packages/sklearn/base.py:493: UserWarning: X does not have valid
feature names, but RandomForestClassifier was fitted with feature
names
  warnings.warn(
/Users/nirendy/school-repo/biology-hw1/.venv/lib/python3.12/site-
packages/sklearn/base.py:493: UserWarning: X does not have valid
feature names, but RandomForestClassifier was fitted with feature
names
  warnings.warn(
/Users/nirendy/school-repo/biology-hw1/.venv/lib/python3.12/site-
packages/sklearn/base.py:493: UserWarning: X does not have valid
feature names, but RandomForestClassifier was fitted with feature
names
  warnings.warn(
/Users/nirendy/school-repo/biology-hw1/.venv/lib/python3.12/site-
packages/sklearn/base.py:493: UserWarning: X does not have valid
feature names, but RandomForestClassifier was fitted with feature
names
  warnings.warn(
/Users/nirendy/school-repo/biology-hw1/.venv/lib/python3.12/site-
packages/sklearn/base.py:493: UserWarning: X does not have valid
feature names, but RandomForestClassifier was fitted with feature
names
  warnings.warn(
/Users/nirendy/school-repo/biology-hw1/.venv/lib/python3.12/site-
packages/sklearn/base.py:493: UserWarning: X does not have valid
feature names, but RandomForestClassifier was fitted with feature
names
  warnings.warn(
/Users/nirendy/school-repo/biology-hw1/.venv/lib/python3.12/site-
packages/sklearn/base.py:493: UserWarning: X does not have valid
feature names, but RandomForestClassifier was fitted with feature
names
  warnings.warn(
/Users/nirendy/school-repo/biology-hw1/.venv/lib/python3.12/site-
packages/sklearn/base.py:493: UserWarning: X does not have valid
feature names, but RandomForestClassifier was fitted with feature
names
  warnings.warn(
/Users/nirendy/school-repo/biology-hw1/.venv/lib/python3.12/site-
packages/sklearn/base.py:493: UserWarning: X does not have valid
feature names, but RandomForestClassifier was fitted with feature
names
  warnings.warn(
/Users/nirendy/school-repo/biology-hw1/.venv/lib/python3.12/site-
packages/sklearn/base.py:493: UserWarning: X does not have valid
feature names, but RandomForestClassifier was fitted with feature
names
  warnings.warn(
/Users/nirendy/school-repo/biology-hw1/.venv/lib/python3.12/site-
packages/sklearn/base.py:493: UserWarning: X does not have valid
feature names, but RandomForestClassifier was fitted with feature
names
  warnings.warn(
/Users/nirendy/school-repo/biology-hw1/.venv/lib/python3.12/site-
packages/sklearn/base.py:493: UserWarning: X does not have valid
feature names, but RandomForestClassifier was fitted with feature
names
  warnings.warn(
/Users/nirendy/school-repo/biology-hw1/.venv/lib/python3.12/site-
packages/sklearn/base.py:493: UserWarning: X does not have valid
feature names, but RandomForestClassifier was fitted with feature
names
  warnings.warn(
/Users/nirendy/school-repo/biology-hw1/.venv/lib/python3.12/site-
packages/sklearn/base.py:493: UserWarning: X does not have valid
feature names, but RandomForestClassifier was fitted with feature
names
  warnings.warn(
/Users/nirendy/school-repo/biology-hw1/.venv/lib/python3.12/site-
packages/sklearn/base.py:493: UserWarning: X does not have valid
feature names, but RandomForestClassifier was fitted with feature
names
  warnings.warn(
/Users/nirendy/school-repo/biology-hw1/.venv/lib/python3.12/site-
packages/sklearn/base.py:493: UserWarning: X does not have valid
feature names, but RandomForestClassifier was fitted with feature
names
  warnings.warn(
/Users/nirendy/school-repo/biology-hw1/.venv/lib/python3.12/site-
packages/sklearn/base.py:493: UserWarning: X does not have valid
feature names, but RandomForestClassifier was fitted with feature
names
  warnings.warn(
```

```
array([[<Axes: xlabel='Age', ylabel='ALE'>,
        <Axes: xlabel='Number of sexual partners', ylabel='ALE'>,
        <Axes: xlabel='First sexual intercourse', ylabel='ALE'>],
       [<Axes: xlabel='Num of pregnancies', ylabel='ALE'>,
        <Axes: xlabel='Smokes (years)', ylabel='ALE'>,
        <Axes: xlabel='Hormonal Contraceptives (years)',
ylabel='ALE'>],
       [<Axes: xlabel='IUD (years)', ylabel='ALE'>,
        <Axes: xlabel='STDs: Number of diagnosis', ylabel='ALE'>,
        <Axes: xlabel='STDs: Time since first diagnosis',
ylabel='ALE'>],
       [<Axes: xlabel='STDs: Time since last diagnosis',
ylabel='ALE'>,
        None, None]], dtype=object)
```



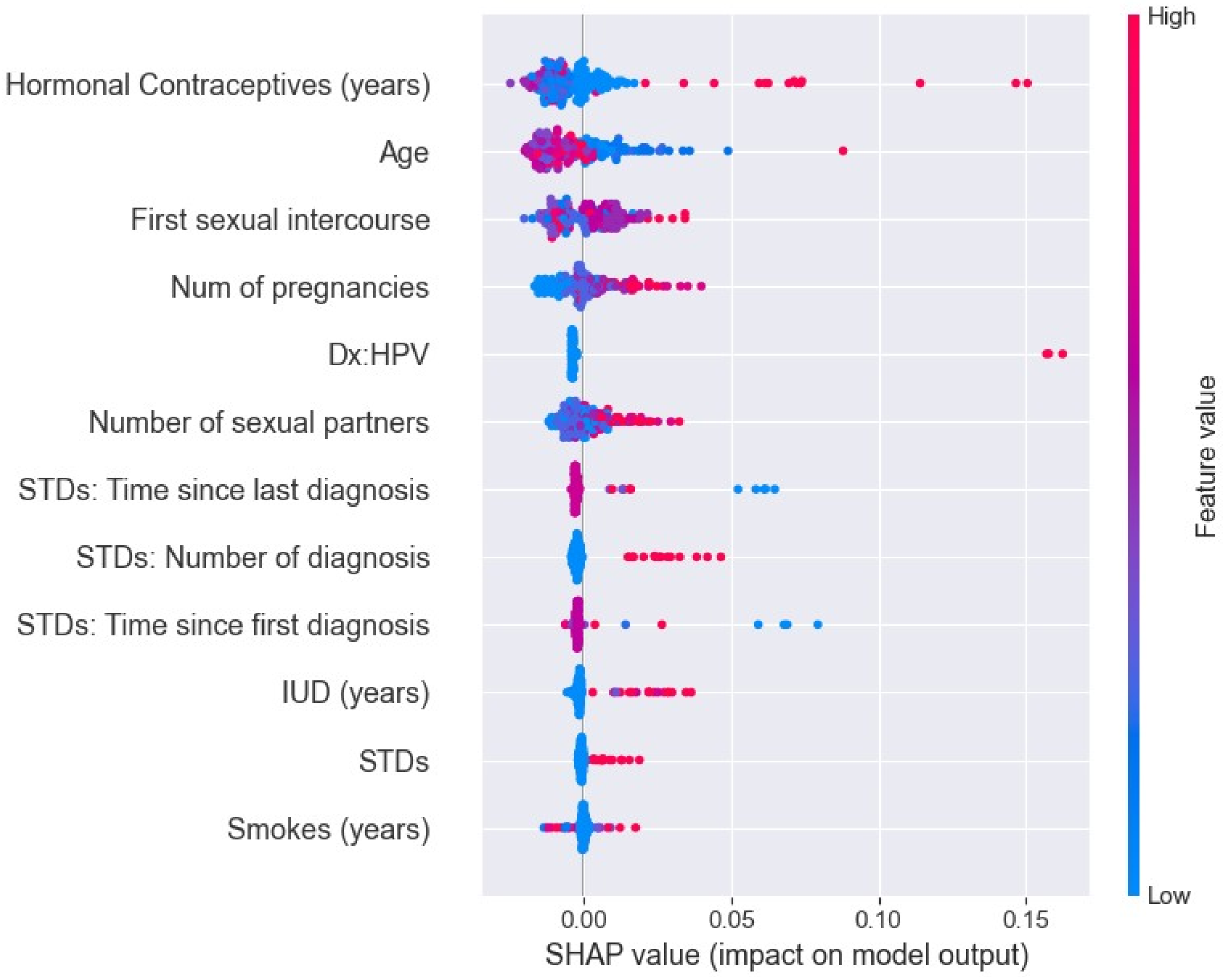
E. Shapley Values

Let us now try to compare how each model relies on each feature, and explain individual predictions. Using the TreeExplainer class of the shap package, calculate the Shapley values of the Random Forest model over the test set.

1. Visualize the Shapley values using a summary plot. How do they compare to ALE plots?

```
from shap import TreeExplainer, summary_plot
explainer = TreeExplainer(forest)
```

```
shap_values = explainer.shap_values(X_test)[:,:,1]
summary_plot(shap_values, X_test)
```



2. Calculate the Shapley feature importance, as the average of the absolute value of the Shapley values. How do they compare to the feature importances determined in D.2?

```
abs_shap_values = np.abs(shap_values).mean(0)
pd.DataFrame(abs_shap_values, index=X_test.columns,
columns=['Importance']).sort_values('Importance', ascending=False)
```

```
                                  Importance
Hormonal Contraceptives (years)     0.012181
Age                                 0.010647
First sexual intercourse            0.008509
Num of pregnancies                  0.007285
Dx:HPV                              0.005950
Number of sexual partners           0.005686
STDs: Time since last diagnosis     0.004279
```

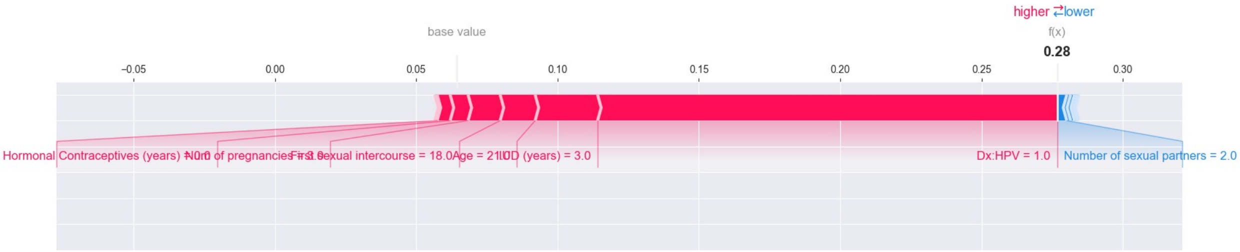
```
STDs: Number of diagnosis        0.004030
STDs: Time since first diagnosis    0.003697
IUD (years)                      0.003478
STDs                             0.001398
Smokes (years)                   0.001315
```

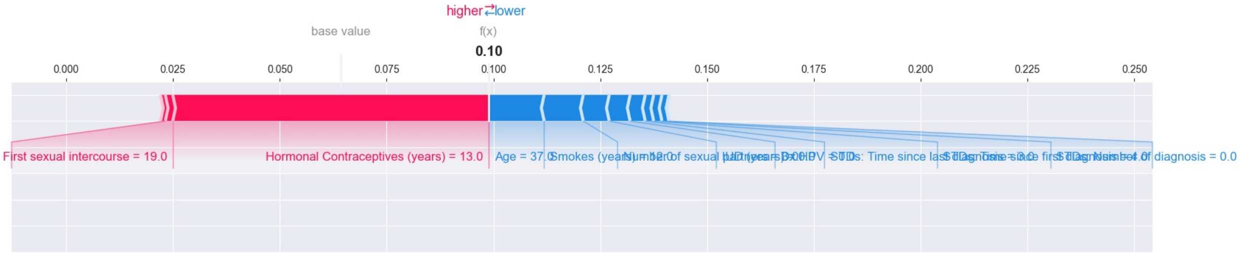
The absolute of average Shapley values somewhat agree with the feature importance in D.2. Namely, in both `Hormonal Contraceptives (years)` is considered the most important feature, and in both `age` is another important feature.

They also agree on features that are not important, such as `Smokes (years)`

3. Pick two test set instances for which the Random Forest model makes an incorrect prediction. Explain their corresponding prediction using a Shapley values force plot.

```
from shap import force_plot
pred = forest.predict(X_test)
incorrect_indices = np.where(pred != y_test)[0]
samples = incorrect_indices[:2]
print(pred[samples])
for sample in samples:
    force_plot(explainer.expected_value[1], shap_values[sample], 
X_test.iloc[sample], matplotlib=True)

[0 0]
```





F. Bonus

Based on the various model interpretations provided, can you come up with a better model, based on a different set of features?