

# Введение в Python

Гейне М.А.

12.09.2025

# Что такое Python?

Python - это высокоуровневый язык программирования общего назначения, известный своей простотой, читабельностью и универсальностью. Он широко используется в различных областях, включая веб-разработку, науку о данных, машинное обучение, автоматизацию и научные вычисления.

# История Python

## Pre-Python

- Конец 80-ых. Голландский программист **Гвидо ван Россум** работает в институте Centrum Wiskunde & Informatica, CWI
- Разрабатывает распределённую ОС Amoeba
- Под рукой только sh и C
- В этом же институте разрабатывается учебный язык ABC
- Идея - сделать язык с читаемостью и простотой ABC, но с применимостью C

# Python 0.9.0

- Февраль 1991, Python 0.9.0 в общем доступе.  
Разрабатывался Гвидо ван Россумом в качестве нового учебного языка на замену ABC.
- Классы с наследованием, исключения, функции, `list`, `dict`, `str` и прочие базовые типы
- Система модулей
- В 1994 появился основной форум Python -> цель на открытость системы

## Fun Fact

Python - это не в честь змеи, а в честь шоу "Летающий цирк Монти Пайтона" (Monty Python's Flying Circus)

## Январь 1994, Python 1.0

- `lambda` , `map` , `filter` , `reduce` . Позаимствованы из Lisp сообществом.
- 1.4: `keyword arguments` , комплексные числа, "сокрытие данных путём искажения имён"
- Computer Programming for Everybody (CP4E)
- BeOpen и GPL совместимая лицензия

## Октябрь 2000, Python 2.0

- List comprehensions (списочное выражение), garbage collector - из ФП
- Python Software Foundation License - собственная открытая лицензия, которая разрешает в том числе приобретение прав на производные работы
- Декабрь 2001, Python 2.2, унификация типов и классов
- Сентябрь 2006, Python 2.5, захват ресурсов выражением `with`. RAII.

# Подготовка к Python 3

- Python 2.6 & 2.7 выходили вместе с 3.0 & 3.1, содержали часть фич из старших релизов, а также предупреждения об удалении фич в новых версиях. В ноябре 2014 объявлено о прекращении поддержки версии 2.7 в 2020 году. Последний релиз, 2.7.18, состоялся 20 апреля 2020 года



## 3 декабря 2008, Python 3.0

- Проектировался, чтобы исправить фундаментальные недостатки языка -> обратно несовместимые изменения -> цифра 3 (см. семантическое версионирование)
- "reduce feature duplication by removing old ways of doing things" - сокращаем повторяющиеся фичи за счёт устранения старых способов работы
- "There should be one — and preferably only one — obvious way to do it"

# Проблемы совместимости

- Смена версий не прошла безболезненно. Были средства автоматизации (например `2to3`), но они справлялись с задачей лишь частично
- Изначальная схема совместимости: держите всю кодовую базу на Python 2, а для Python 3 делайте дополнительную версию с помощью `2to3`, и изменения туда лучше не вносить. Получалось две кодовые базы.
- С 2012 года было решено держать только одну кодовую версию для 2.7 версии языка и для 3.x одновременно, а совместимости добиваться подключением модулей совместимости, например `six`.
- К примеру, `six.string_types` в 2 версии был `(str, unicode)`, а в 3 - `str`. `isinstance(my_var, six.string_types)` за счёт этого работал одинаково в обеих версиях языка

# Основные фишки Python 3

- `print` стал функцией, а не выражением -> легко заменить его поведение
- Замена функции `input`, которая раньше не просто возвращала строку, а вычисляла входное выражение
- Убран `reduce` из-за меньшей читабельности, чем у циклов
- Добавлены аннотации типов

## Fun Fact

Долгое время Гвидо ван Россум носил титул **BDFL** — "Великодушный пожизненный диктатор" (Benevolent Dictator For Life)

В 2018 году он ушел в отставку с этого поста. Теперь язык управляется **Руководящим советом** (Steering Council) из 5 человек, который ежегодно переизбирается

# Основные фичи Python 3

- Типы `str` и `Unicode` были унифицированы
- Добавлены типы для последовательностей байтов
- Убраны особенности обратной совместимости
- Изменено поведение целочисленного деления.  $9 / 2 = 4$   
vs  $9 / 2 = 4.5$
- Полная поддержка `Unicode` в исходном коде

## Другие крупные релизы после 3.0

- 3.6: f-строки, подсказки типов, улучшения asyncio
- 3.7: Data classes, `async with`
- 3.9: `setdefault()` и `popitem()` для словарей, улучшения в работе с таймзонами, изменение конвенций об именовании встроенных модулей
- 3.10: Pattern matching, union operator `|`

## Другие крупные релизы после 3.0

- 3.11: Performance boost CPython, группы исключений
- 3.12: Per-interpreter GIL, более гибкие f-строки
- 3.13: Улучшенный REPL. Экспериментальная JIT-компиляция и настоящая многопоточность
- 3.14: Настоящая многопоточность в релизе *Coming soon*

# Ключевые особенности Python

- **Удобство чтения:** Синтаксис Python разработан так, чтобы его было легко читать и понимать, что делает его отличным выбором для новичков.
- **Универсальность:** Python можно использовать для решения широкого круга задач, от простых скриптов до сложных приложений.
- **Большое сообщество:** Python имеет большое и активное сообщество, предоставляющее обширную документацию, библиотеки и поддержку



# Ключевые особенности Python

- **Кроссплатформенная совместимость:** Python работает на различных операционных системах, включая Windows, macOS и Linux.
- **Динамическая типизация:** Python автоматически определяет тип данных переменных во время выполнения программы, что делает его более гибким по сравнению с языками со статической типизацией.
- **Интерпретация over компиляция:** Python исполняется интерпретатором, т.е. не имеет этапа компиляции программы.

# Виды интерпретатора

1. **CPython**. Основной интерпретатор Python, наиболее распространённый и считающийся стандартом. Написан на, внезапно, C.
2. **PyPy**. JIT-компилятор для Python. Быстрее CPython, особенно в вычислительно сложных задачах, совместим с большинством библиотек
3. **Jython**. Python to Java bytecode
4. **IronPython**. Python to .NET
5. **Stackless Python**. CPython, но с отказом от стека вызовов C в пользу собственного

# Применение Python

- Data Science and Machine Learning
  - NumPy, Pandas и Matplotlib - основные средства для интерпретации и визуализации данных
  - Sklearn, TensorFlow, Keras, PyTorch - ML фреймворки для построения и обучения моделей
- Scientific Computing
  - SciPy и SymPy - численные симуляции и математические исчисления
  - Matplotlib - визуализация научных данных

# Применение Python

- Веб-разработка
  - Django, Flask, Pyramid - фреймворки для разработки веб приложений
  - BeautifulSoup и Scrappy - инструменты извлечения данных с веб-страниц
- Образование
  - Python - изначально учебный язык, и он до сих пор используется для обучения программированию

# Применение Python

- Автоматизация
  - Простой скриптовый язык позволяет достаточно легко и быстро автоматизировать ручные рутинные задачи
- Системы управления
  - Home Assistant - система управления умным домом с открытым исходным кодом, работающая в основном на Python
  - Интеграции с умными устройствами пишутся на Python и доступны любому для редактирования

# Пример анализа данных

## 0. Подготовка датасета

ФИО	Срок постановки на учет в Ж/К	Дата постановки на учет в центре	Способ родоразрешения	Х/П с какого срока	ВН в 36 недель	Дата родов	Срок беременности	Х/п ребенку	Примечания	Год
[REDACTED]	Не состояла	24.04.2016	Самостоят роды	Однократно со срока 18 нед, назначена в МРЦКБ	неизвестная	01.01.2019	39	3 преп, отказ от ребенка	Не наблюдалась, Бродяжничала. Активный ПИН	2019
[REDACTED]	27.07.2018-8 недель	03.08.2007	к/с плановое	Арвт до беременности	Отрицательная	17.01.2019	38	1 препарат		2019
[REDACTED]	01.08.2018-12 нед	21.07.2010	к/с плановое	34 нед	Положительная	07.02.2019	39-40	3 преп	ВИЧ-диссидентка	2019
[REDACTED]	08.02.2019-33 нед	28.05.2015	к/с экстренное	АРВТ до беременности	Положительная	11.02.2019	34	3 препарат, отказ от ребенка	Не наблюдалась, Бродяжничала. Активный ПИН, коммерческая секс-работница	2019
[REDACTED]	17.08.2018 12 нед	06.03.2017	к/с плановое	Арвт до беременности	Отрицательная	12.02.2019	38-39	3 препарат	Глубокая иммуносупрессия	2019
[REDACTED]	21.08.2018-12 нед	14.05.2013	к/с плановое	С 14 нед	Отрицательная	24.02.2019	38-39	1 препарат		2019
[REDACTED]	10.10.2018-16 нед	20.10.2011	к/с экстренное	С 18 нед	Отрицательная	25.02.2019	38 нед	3 препарат		2019
Горькова	20.08.2019-11 нед	02.06.2010	к/с плановое	С 37 нед	Отрицательная	05.03.2019	38-39	3 препарат	Не наблюдалась в Центре, не направлена ж/к	2019

# 1. Загрузка датасета

```
df = pd.read_excel('Dataset2.xlsx')

date_columns = ['OBGYN', 'AIDS', 'BirthDate']

for col in date_columns:
    df[col] = pd.to_datetime(df[col], format='%d.%m.%Y', errors='coerce')
```

	Name	Week	AIDS	BirthType	Therapy	Renewed	Status	Length	Dosage	Comments	Year
0	Дав	39	-709	сам	-2	0.0	1.0	39	3.0	Не наблюдалась, Бродяжничала.Активный ПИН, отказ от ребенка	2019
1	Мак	8	-3919	план	0	0.0	0.0	38	1.0	0.0	2019
2	Тим	12	-2843	план	34	0.0	1.0	40	3.0	ВИЧ-диссидентка	2019
3	Фил	33	-1117	план	0	0.0	1.0	34	3.0	Не наблюдалась, Бродяжничала.Активный ПИН ,коммерческая секс работница, отказ от ребенка	2019
4	Сел	12	-435	план	0	0.0	0.0	39	3.0	Глубокая иммуносупрессия	2019
5	Ива	12	-1839	план	14	0.0	0.0	39	1.0	0.0	2019
6	Гал	16	-2419	экстр	18	0.0	0.0	38	3.0	0.0	2019
7	Гор	11	-2925	план	37	0.0	0.0	39	3.0	Не наблюдалась в Центре ,не направлена ж/к	2019
8	Ерм	8	-1368	план	14	0.0	0.0	39	1.0	0.0	2019
9	Кар	15	89	план	18	0.0	0.0	38	3.0	Тяжелый гепатоз беременных	2019

## 2. Трансформация данных

```
df['LengthDelta'] = pd.to_timedelta(df['Length'], unit='W')
df['PregnancyStart'] = df['BirthDate'] - df['LengthDelta']

df['OBGYN'] = df['OBGYN'].fillna(df['BirthDate'])
df['AIDS'] = df['AIDS'].fillna(df['BirthDate'])

missing_week = df[df['Week'] == '-']
missing_week['Week'] = missing_week['Length'] - (pd.to_timedelta((missing_week['BirthDate']
    * missing_week['OBGYN'])).dt.days // 7)
df.update(missing_week)
df['Week'] = df['Week'].astype(int)

# Initialize LabelEncoder
label_encoder = LabelEncoder()

# Fit and transform the 'BirthType' column
df['BirthType'] = label_encoder.fit_transform(df['BirthType'])
```

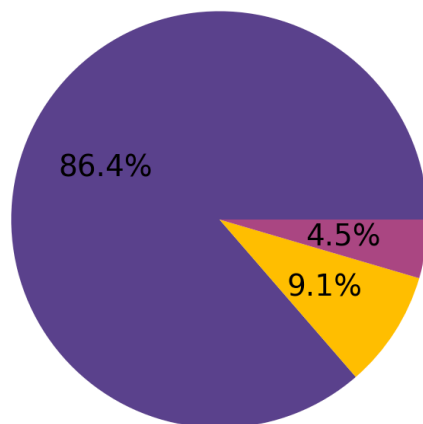


## 2. Трансформация данных

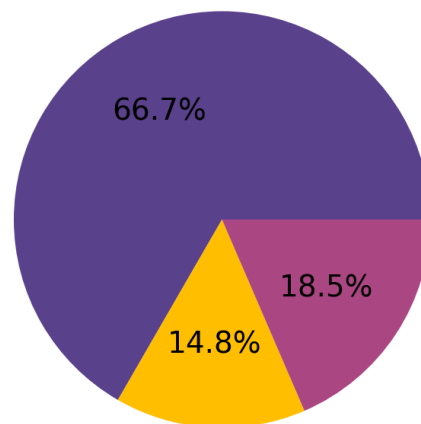
	Week	AIDS	BirthType	Therapy	Renewed	Status	Length	Dosage	PoorCommitment	SocialRisk	Migration	FSIN	Illness	Complications	Death
0	39	-709	1	-2	0.0	1.0	39	3.0	1.0	1.0	0.0	0.0	0.0	0.0	0.0
1	8	-3919	0	0	0.0	0.0	38	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2	12	-2843	0	34	0.0	1.0	40	3.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0
3	33	-1117	0	0	0.0	1.0	34	3.0	1.0	1.0	0.0	0.0	0.0	0.0	0.0
4	12	-435	0	0	0.0	0.0	39	3.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0
5	12	-1839	0	14	0.0	0.0	39	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
6	16	-2419	2	18	0.0	0.0	38	3.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
7	11	-2925	0	37	0.0	0.0	39	3.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0
8	8	-1368	0	14	0.0	0.0	39	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
9	15	89	0	18	0.0	0.0	38	3.0	0.0	0.0	0.0	0.0	1.0	1.0	0.0

### 3. Визуализация данных

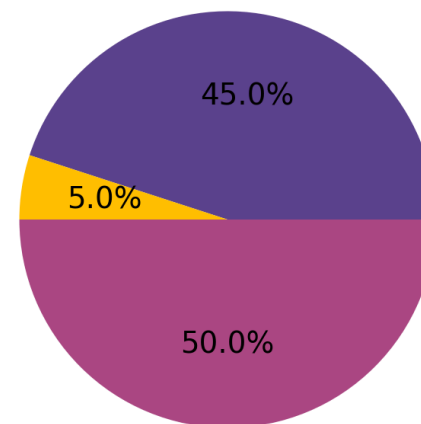
2019 год



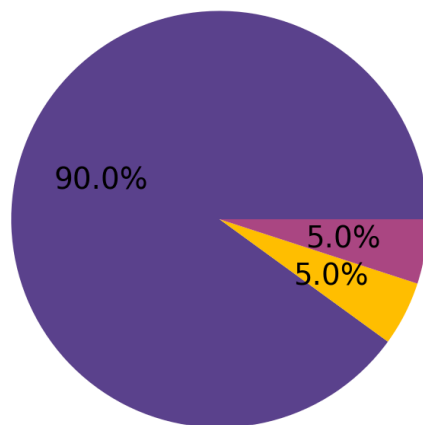
2020 год



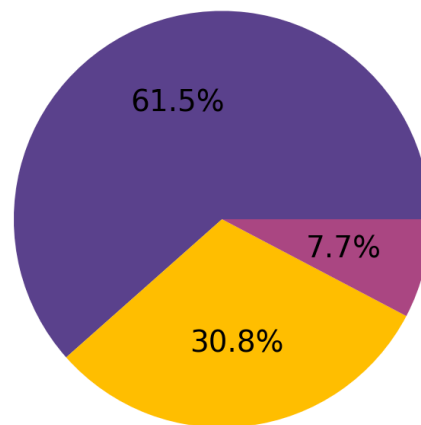
2021 год



2022 год



2023 год



### 3. Визуализация данных

```
# Get yearly data
yearly_percentages = df.groupby('Year')['BirthType'].value_counts(normalize=True).sort_index() * 100

# Create a single figure with 5 subplots
fig, axs = plt.subplots(2, 3, figsize=(15, 10))

# Plot pie charts for each year
for i, (year, percentages) in enumerate(yearly_percentages.groupby(level=0)):
    # birth_types = [label[1] for label in percentages.index]
    wedges, texts, autotexts = axs[i//3][i%3].pie(percentages, autopct='%1.1f%%', colors=generate_colors(len(percentages)))
    axs[i//3][i%3].set_title(f'{year} год')
axs[-1, -1].legend(wedges, labels, loc='center', fontsize=14)
axs[-1, -1].axis('off')
plt.savefig('fig2.png', transparent=True)
plt.show()
```

## 4. Разбиение данных на выборки

```
X = df.drop(columns=['Dosage'])
y = df['Dosage']

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42, stratify=df['Dosage'])

print("X train (features):")
print(X_train.head())
print("\nX test (features):")
print(X_test.head())
print("\ny train (target variable):")
print(y_train.head())
print("\ny test (target variable):")
print(y_test.head())
```

## 4. Разбиение данных на выборки

X train (features):

	Week	AIDS	BirthType	Therapy	Renewed	Status	Length	PoorCommitment	\
69	7	-5036	0	8	1.0	0.0	39	0.0	
101	38	291	0	39	0.0	1.0	39	0.0	
23	9	-536	0	0	0.0	0.0	38	0.0	
51	11	-2024	2	0	0.0	0.0	35	0.0	
19	30	-328	0	0	1.0	1.0	30	0.0	
	SocialRisk	Migration	FSIN	Illness	Complications	Death			
69	0.0	0.0	0.0	0.0	0.0	0.0	0.0		
101	0.0	1.0	0.0	0.0	0.0	0.0	0.0		
23	0.0	0.0	0.0	0.0	0.0	0.0	0.0		
51	0.0	0.0	0.0	0.0	0.0	0.0	0.0		
19	0.0	0.0	0.0	1.0	0.0	0.0	0.0		

X test (features):

	Week	AIDS	BirthType	Therapy	Renewed	Status	Length	PoorCommitment	\
85	12	-4744	0	0	0.0	0.0	41	0.0	
86	9	-3735	2	0	0.0	0.0	28	0.0	
67	17	124	2	17	0.0	0.0	33	0.0	
16	10	-3953	0	0	0.0	0.0	39	0.0	
37	9	-3170	0	0	0.0	0.0	39	0.0	
	SocialRisk	Migration	FSIN	Illness	Complications	Death			
85	0.0	0.0	0.0	0.0	0.0	0.0			
86	0.0	0.0	0.0	0.0	0.0	0.0			
67	0.0	0.0	1.0	0.0	0.0	0.0			
16	0.0	0.0	0.0	0.0	0.0	0.0			
37	0.0	0.0	0.0	0.0	0.0	0.0			

```
y train (target variable):  
69      3.0  
101     3.0  
23      1.0  
51      3.0  
19      3.0  
Name: Dosage, dtype: float64  
  
y test (target variable):  
85      1.0  
86      1.0  
67      3.0  
16      1.0  
37      1.0  
Name: Dosage, dtype: float64
```

## 5. Обучение

```
# Create a StandardScaler object
scaler = StandardScaler()

# Fit the scaler on the training data
scaler.fit(X_train)

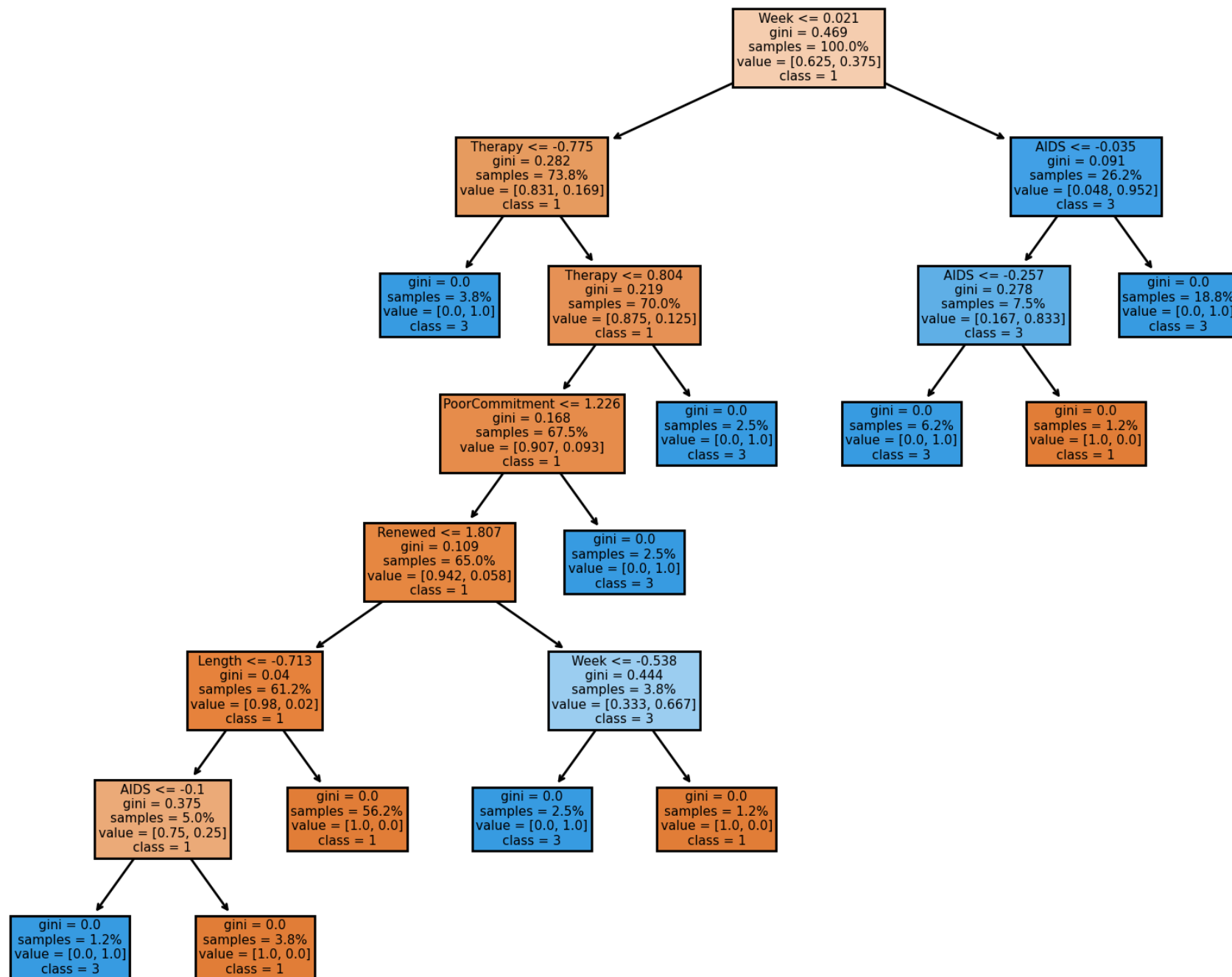
# Transform the training and testing data
X_train_scaled = scaler.transform(X_train)
X_test_scaled = scaler.transform(X_test)

# Instantiate the DecisionTreeClassifier with optional hyperparameters
tree_classifier_scaled = DecisionTreeClassifier(max_depth=50)

# Fit the model to the training data
tree_classifier_scaled.fit(X_train_scaled, y_train)

# Make predictions on the testing data
predictions_scaled = tree_classifier_scaled.predict(X_test_scaled)
```







**Спасибо за  
внимание!**