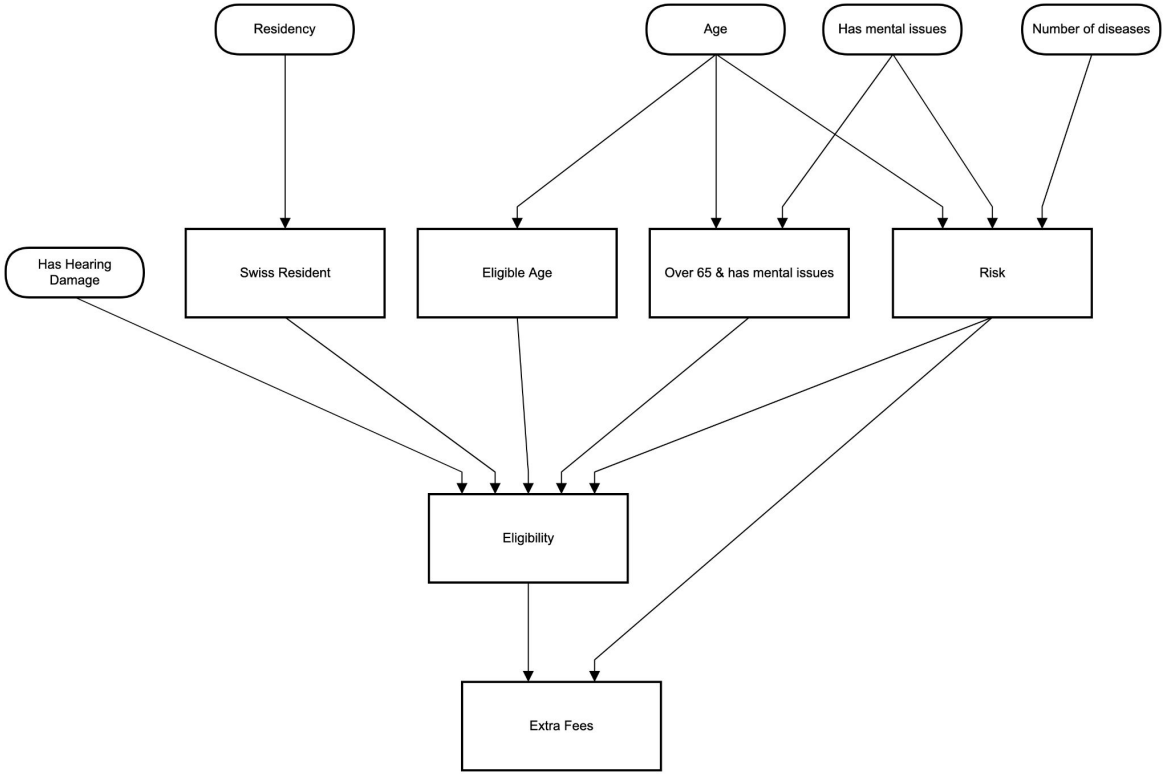
AI Technologies

Adam Threlfall & Ulrich Pogson

Decision Tree



Swiss Resident

Swiss Resident | **Hit Policy:** Unique

	When Country string	Then Eligible boolean
1	Switzerland	true
2	Germany	false
+	-	

Age

Eligible Age | **Hit Policy:** Unique

	When Age integer	Then Eligible boolean
1	[21 .. 70]	true
2	[1 .. 21)	false
3	(70 .. 99]	false

Over 65 & Mental Issues

Over 65 & has mental issues | **Hit Policy:** Unique

	When Age integer	And Mental Health Issues boolean	Then Over 65 & has menta… boolean
1	> 65	true	true
2	<= 65	false	false
3	<= 65	true	false
4	> 65	false	false

Risk

Risk | **Hit Policy:** Unique

	When Age integer	And Has Mental Health Is… boolean	And Number of disease integer	Then Risk "low","medium","high"
1	>55	true	1	medium
2	[21 .. 70]	-	2	medium
3	[21 .. 70]	-	3	high
4	[21 .. 70]	false	0	low

Extra fees

Extra Fees | **Hit Policy:** Unique

	When Eligible boolean	And Risk "low","medium","high"	Then Extra Fees boolean
1	true	low	false
2	false	low	false
3	true	medium	true
4	false	meidum	false
5	true	high	false
6	false	high	false

Eligibility

Eligibility | **Hit Policy:** Unique

	And Swiss Resident boolean	And Has Hearing Damage boolean	And Over 65 & has mental… true	And Risk boolean	Then Eliagable boolean
1	Switzerland	fale	-	low	true
2	Switzerland	false	-	low	true
3	Switzerland	true	-	low	false
4	Switzerland	false	-	low	false
5	Switzerland	false	-	high	false
+	-	-	-	-	

Prolog

V1

```
ageRange(A) :- A >= 21, A =< 70.
swissResident(R) :- R == ch.
hasHearingDamage(H) :- H == true.
overSixyfiveAndhasMentalIssue(A, M) :- A > 65, M == true.
riskLow(A, M, D) :- \+ riskMedium(A, M, D); \+ riskhigh(D).
riskMedium(A, M, D) :- A >= 55, M == true; D >= 2.
riskhigh(D) :- D >= 3.

eligable(A,R,H,M,D) :-
    ageRange(A),
    swissResident(R),
    \+ hasHearingDamage(H),
    \+ overSixyfiveAndhasMentalIssue(A, M),
    \+ riskhigh(D).

extraFees(A,R,H,M,D) :-
    eligable(A,R,H,M,D),
    riskMedium(A, M, D).
```

Prolog

V2

```
withinAgeRange(A) :-
    A >= 21, A =< 70.
isSwissResident(R) :-
    R == ch.
hasHearingDamage(H) :-
    H == true.
overSixyfiveAndhasMentalIssue(A, M) :-
    A > 65, M == true.
risk(_A,_R,_H,_M,D, high) :-
    D >= 3.
risk(A,R,H,M,D, low) :-
    \+ risk(A,R,H,M,D, medium);
    \+ risk(A,R,H,M,D, high).
risk(A,_R,_H,M,D, medium) :-
    A >= 55, M == true; D >= 2.

eligible(A,R,H,M,D) :-
    withinAgeRange(A),
    isSwissResident(R),
    \+ hasHearingDamage(H),
    \+ overSixyfiveAndhasMentalIssue(A, M),
    \+ risk(A,R,H,M,D, high).

extraFees(A,R,H,M,D) :-
    eligible(A,R,H,M,D),
    risk(A,R,H,M,D, medium).
```

Prolog

V3

```
% Information regarding customer with ID 123.
age(123,65).
residency(123,ch).
disease(124, 'Mental Issue' ).
disease(124,'Hearing Damange').

withinAgeRange(ID) :-
    age(ID,Age), Age >= 21, Age =< 70.
isSwissResident(ID) :-
    residency(ID,ch).
hasHearingDamage(ID) :-
    disease(ID,'Hearing Damange').
overSixyfiveAndhasMentalIssue(ID) :-
    age(ID,Age), Age > 65, disease(ID, 'Mental Issue' ).
risk(_ID,D, high) :-
    D >= 3.
risk(ID,D, low) :-
    \+ risk(ID,D, medium);
    \+ risk(ID,D, high).
risk(ID,D, medium) :-
    age(ID,Age), Age >= 55, disease(ID, 'Mental Issue' ); D >= 2.

eligible(ID,D) :-
    withinAgeRange(ID),
    isSwissResident(ID),
    \+ hasHearingDamage(ID),
    \+ overSixyfiveAndhasMentalIssue(ID),
    \+ risk(ID,D, high).

extraFees(ID,D) :-
    eligible(ID,D),
    risk(ID,D, medium).
```

Machine Learning

[ ]:
```python
import pandas as pd
data = pd.read_csv('project_data.csv')
```

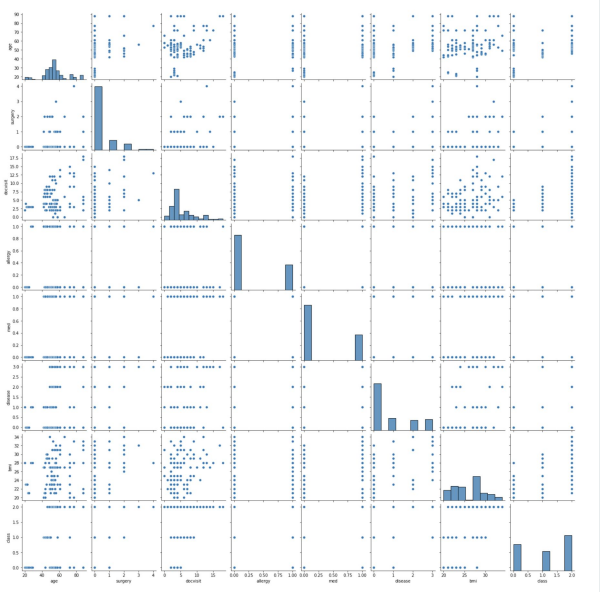
[ ]:
```python
data
```

[ ]:

	age	surgery	docvisit	allergy	med	disease	bmi	class
0	20	0	2	no	no	cholesterol	28	low
1	21	0	4	no	no	no	23	low
2	22	0	3	no	no	no	23	low
3	23	0	3	no	no	no	23	low
4	24	0	3	no	no	no	21	low
..	...	...	...	...	...	...	...	...
112	88	0	2	yes	yes	no	21	low
113	88	0	2	no	no	no	22	low
114	88	2	18	yes	yes	no	28	high
115	88	1	5	no	yes	diabetes	33	high
116	88	2	17	no	yes	diabetes	32	high

[117 rows x 8 columns]

Preprocessing



[ ]:
```
data.dtypes
```

[ ]:
```
age          int64
surgery      int64
docvisit     int64
allergy      int64
med          int64
disease      int64
bmi          int64
class        int64
dtype: object
```

[ ]:
```
target = data.pop('class')
```

[ ]:
```
data
```

[ ]:

	age	surgery	docvisit	allergy	med	disease	bmi
0	20	0	2	0	0	1	28
1	21	0	4	0	0	0	23
2	22	0	3	0	0	0	23
3	23	0	3	0	0	0	23
4	24	0	3	0	0	0	21

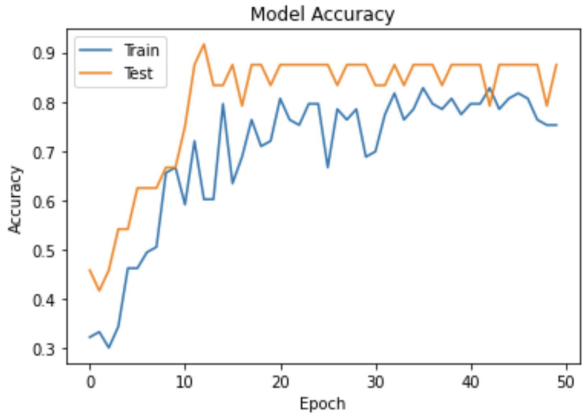
[ ]:
```
target
```

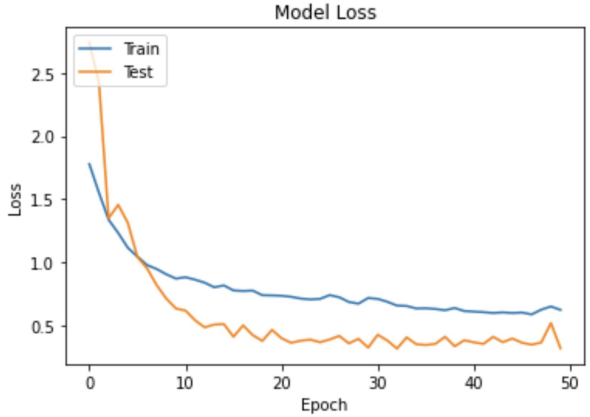
[ ]:
```
0    0
1    0
2    0
3    0
4    0
     ..
```

Training

[ ]:
```
from keras.models import Sequential
from keras.layers import Dense,Flatten,Activation,Dropout

model = Sequential()
model.add(Flatten())
model.add(Dense(100, input_dim=7, activation='relu'))
model.add(Dense(3, activation='softmax'))

model.
 ↪compile(optimizer='adam',loss='sparse_categorical_crossentropy',metrics=['accuracy'])

history = model.fit(data, target, validation_split=0.2, epochs=50,␣
 ↪batch_size=16, verbose = 2)
```

Evaluation





[ ]:
```
model.evaluate(data,target, batch_size=16, verbose = 3)
```

[ ]:
```
[0.5567188262939453, 0.8034188151359558]
```

Prediction

```
[ ]: import numpy as np
     predict = model.predict(data)
     classes = np.argmax(predict,axis=1)

     classes = np.where(classes == 0,'low', classes)
     classes = np.where(classes == '1','medium', classes)
     classes = np.where(classes == '2','high', classes)

     print(classes)
```

```
['low' 'low' 'low' 'low' 'low' 'low' 'low' 'low' 'medium' 'medium' 'high'
 'medium' 'high' 'high' 'medium' 'low' 'medium' 'high' 'high' 'medium'
 'high' 'medium' 'medium' 'high' 'medium' 'medium' 'medium' 'high' 'high'
 'high' 'high' 'high' 'medium' 'low' 'high' 'high' 'high' 'high' 'low'
 'high' 'low' 'high' 'high' 'high' 'low' 'high' 'high' 'high' 'low' 'high'
 'medium' 'low' 'high' 'low' 'high' 'high' 'low' 'high' 'high' 'high'
 'low' 'low' 'low' 'low' 'high' 'low' 'high' 'high' 'high' 'low' 'high'
 'high' 'low' 'low' 'high' 'high' 'low' 'high' 'high' 'low' 'high' 'high'
 'high' 'high' 'high' 'low' 'high' 'low' 'high' 'low' 'high' 'low' 'low'
 'high' 'high' 'high' 'low' 'high' 'low' 'low' 'low' 'low' 'low' 'high'
 'low' 'high' 'low' 'high' 'high' 'high' 'high' 'low' 'low' 'low' 'high'
 'high' 'high']
```

Combination of knowledge from experts and past applications