DSCI 430 - Fairness, Accountability, Transparency and Ethics (FATE) in Data Science

Module 1 - Introduction and Ethical foundations

Assignment overview

This assignment is composed of three parts:

- **Part 1 - The Black Mirror Writers' Room.** In this portion of the exercise, you will brainstorm near future technology and its possible drawbacks, and illustrate them in a futuristic cautionary tale. You will also be asked questions about ethical theories and how they apply to the scenario you have described. Credits: Casey Fiesler - The Black Mirror Writers Room: The Case (and Caution) for Ethical Speculation in CS Education
- **Part 2 - Python review.** As this course uses Python as the programming language for our exercises, a basic understanding of the fundamentals and the use of some libraries is necessary. This portion of the exercise will help you review useful Python syntax and/or fill the gap in your knowledge before tackling larger exercises. We recommend discussing with an instructor if you find this portion of the assignment too difficult to complete with a reasonable amount of effort.
- **Part 3 - Final thoughts.** Complete this section so that we can better understand how you completed the assignment and any issues you may have encountered.

For this assignment, it is possible to work in **groups of up to 2 students**. Read the instructions carefully, as they may assign tasks to specific students.

Group members

Leave Student 2 blank if group has less than 2 members:

- Student 1: Ayuho Negishi
- Student 2: Muhan Yang

Learning Goals:

After completing this week's lecture and tutorial work, you will be able to:

1. Define ethics and describe what constitutes an ethical issue

2. Explain the need for ethics in data science
3. Identify common ethical issues in data science
4. Describe common ethical frameworks and how they can be applied to data science applications
5. Imagine scenarios in which current technology could be used in unethical ways
6. Evaluate and make arguments around data science scenarios using ethical theories (e.g., Kantianism, utilitarianism, virtue ethics etc.)
7. Compare and contrast different ethical theories and explain the case for and against each one as they apply to data science

The Black Mirror Writers' Room

Black Mirror is a Netflix series centered around the use of advanced technology and its possible unexpected (sometimes catastrophic) consequences. In this exercise, you will come up with your very own Black Mirror episode (or at least a synopsis)!

Warm up

Before jumping into the creative writing part, we should review the various elements of FATE in Data Science and make sure that they are clear:

FATE element	Definition
Fairness	The idea that every group or population that is affected by a technological application is being treated equally and not receiving a different outcome *solely because they belong to their group*.
Accountability	Clear definition of who should be held responsible of the outcome of the technological application and under what circumstances.
Transparency	The technical definition of transparency in Data Science refers to being able to understand why a technological application produced a specific outcome. This is also called *explainability*. But transparency can also refer to the demand of making the use of algorithms more transparent to the public, including informing the users about when they are used, where the data used was sourced from, and making algorithms available for auditing.
Ethics	Evaluation of whether or not a technology should be used based on the moral values of a group or society. Society may reject a technology because it is does not follow the principles of Fairness, Accountability or Transparency, but also for other reasons.

Question 1

Consider the following scenario:

In the country of Dataland, the police department uses an algorithm to assess the risk level of people reporting cases of domestic abuses and violence. Thanks to this algorithm, they can identify the most serious threats and intervene accordingly. The algorithm has had a positive impact, assessing cases with more accuracy than other prior strategies and allowing the police force to make an efficient use of their resources. However, it occasionally fails to correctly identify people at high risk of violence (*false negatives*), leaving them without the protection they need. It is also affected by other issues. For each issue outlined in this table, check whether it is a Fairness, Accountability or Transparency problem.

Issue	Fairness	Accountability	Transparency
When the algorithm fails to identify a high-risk case and violence occurs, it is unclear if the police department should shoulder any responsibility.		○	
An analysis of the algorithm's results suggests that false negatives occur more frequently among victims with physical disabilities.	○		
The majority of people reporting domestic abuse are not aware that their cases are being evaluated by an algorithm, or do not know the score they received.			○
The police department receives a recommendation for each case, but does not know which characteristic(s) of the case have resulted in the final evaluation.			○
The algorithm was trained using past cases filed by the police department, but the people involved where not informed that their information was being used for this purpose.			○

Question 2

Considere the issues outlined in the previous question, as well as the fact that the algorithm is the best system of appraisal available to the police forces so far for cases of domestic violence. Do you think that the use of this algorithm is *ethical*? Clearly state your thesis (opposed/favourable) and use one of the ethical perspectives listed in this reading to support it.

I am opposed to the use of this algorithm and its ethics, because it goes against the Utilitarian perspective, which emphasizes on the overall happiness, the balance of interests, and the consequences. This perspective holds that an action is ethical as long as it increases the happiness and welfare of all those people involved.

Consequences: However, the algorithm "occasionally fails to correctly identify people at high risk of violence (false negatives)", and those categorized as false negatives are unable to receive the necessary protection. As a consequence, since the violence is the most serious threat and people who become the false negatives involved in the *algorithm lack protection and their safety or even their lives are facing fatal challenges. Moreover, the algorithm is facing ethical concerns in that people who were abused are not aware that their cases are used in the algorithm. As a consequence, if they know this issue one day, it might cause them emotionally uncomfortable and even distressed. In addition, given the police department has no idea about how the algorithm evaluate the data, as the consequence, if the algorithm reports more and more wrong cases, the police department cannot locate the person or group being responsible for the mistake, cannot fix and improve the algorithm, and might also lead to a decent number of false cases before realizing the problem.*

Interests: The positive impacts that the algorithm bring is that the police could efficiently deal with most cases and assess cases with higher accuracy.

Even though it has the positive side and people could potentially benefit from this algorithm, we would argue that the interests and consequences are not balanced. The consequences apparently outweigh the interests that the algorithm brings, as certain people's life are in danger because of the false negatives in the algorithm, certain people might potentially develop psychological issue because of the algorithm used their private information/cases without consent, and the police might fail to identify responsibility in each step of the algorithm. Therefore, the consequences of the algorithm outweigh its advantages, and thus it did not maximize the happiness and interests for people involved, which refutes the viewpoint proposed by the Utilitarian perspective.

Note: this case is fictional but inspired by a real algorithm, called VioGén, used in Spain to determine the risk level of victims of gender-based violence and assign protection measures. The algorithm has been recently going under severe scrutiny (Read more).

Question 3: Write your own Black Mirror episode

Now that you have acquired the necessary familiarity with some required knowledge and terminology, it is time to use your creativity!

Step 1: Brainstorm *one* near future technology based on a topic of your choice. It should be close enough that it seems like a plausible future. Describe it in the next cell.

Microchips are implanted directly into the human body to store personal and financial information, such as IDs, medical records, and credit card details. This technology allows easy access to services, quick payments, and enhanced convenience in everyday life, replacing wallets and physical identification.

Step 2: What are the potential social implications and/or ethical issues and/or regulatory challenges with this technology? Explain if and how they are connected to FATE (e.g. is it a Fairness issue? Or maybe a Transparency issue? It could be more than one option).

The microchips technology might have multiple issues relate to the violation of FATE, and two of them could be:

***Fairness issue:** Not everyone can afford the advanced security versions of the chip. Vulnerable populations (e.g. lower socio-economic status group) may end up with less secure versions, making them easier targets for exploitation. Criminals could target these individuals, knowing their chips are less protected.*

***Transparent issue:** Once microchips are put into practice in people's every-day life, a majority of people as users might not be aware of or be informed by how much private personal information of themselves are collected by the algorithm, given its popularity and wide usage because of the easy access.*

Step 3: Time for storytelling! Write the summary of a new Black Mirror episode based on the technology of your choice. Try covering all of the following:

- What do you think might be a cautionary tale related to this technology?
- What fictional person in the future would best illustrate this caution? Provide a detailed description and explain what makes them the best character to carry your message.
- What is their story? Explain their background, their motivations, and their journey through your episode.

As part of your submission, please update the episode thumbnail slide (from Module 1 slide deck) using information from your episode. Don't forget to add a picture! Then, share it with the rest of the class on Canvas.

- *Cautionary Tale: In the near future, people have microchips implanted in their bodies to store personal information like IDs, medical records, and credit card details. This makes daily life more convenient, as people no longer need to carry wallets or identification. However, a dark side of this technology begins to emerge. Criminals start targeting individuals with less secure chips, knowing they are easier to hack. They violently steal body parts where the chips are implanted to extract sensitive information, causing fear and panic throughout society.*
- *Fictional Charactor: John, a 35-year-old delivery driver from a working-class background, uses a basic, less-secured chip for payments and ID. As crimes targeting chip users rise, he becomes anxious, knowing he cannot afford better protection. He represents those vulnerable due to financial limitations.*
- *Story: John, the 35-year-old delivery driver, lives in fear as news spreads about criminals attacking people with microchips implanted in their wrists. The criminals steal personal information by cutting off the part of the body where the chip is placed. Most of the victims are people with low incomes who cannot afford a more secure version of the chips, like John. However, since the microchips ease people's every-day life, almost every place and store refuses to take transactions using*

> *traditional methods, and people on the relatively low socio-economics status like John have no choice but to be forced to use it to maintain a normal life, due to the development of technology. John's fear grows worse when his uncle, who is also from a working-class background, is attacked. The criminals cut his wrist to take the chip, and he dies from losing too much blood. This is the first reported death from chip-related crimes. The tragedy triggers public outrage, sparking protests and riots fueled by anger over the unequal security of the technology and the government's failure to protect vulnerable groups. As the situation gets more dangerous, John realizes he is a potential target. He cannot afford to upgrade his chip, so he thinks about joining the underground network to illegally remove his chip altogether, despite the physical life risks and not being to live normally in a futuristic modern city. John must decide whether to live in constant fear or take a dangerous step to protect himself.*

Step 4: Let's take a step back, and imagine that you are (one of) an activist/legistlator/technology practitioner at the time the technology described in your episode is being developed and its use being discussed. Select *one* of the ethical perspectives listed in this reading, and use this perspective to argue against its deployment. Pay particular attention to the counter-arguments! People with interests in this technology will certainly argue against you, and you must anticipate and rebut their claims. Include at least 2 conter-arguments and how you would respond to them.

Ethical perspective chosen: the Justice/Fairness Perspective

Argument: The use of microchips is unfair because the benefits and risks are not shared equally. People from lower socioeconomic groups are more at risk since they can't afford the secure versions of the chips. While wealthier individuals enjoy the benefits of security and convenience, poorer people face a higher chance of being targeted by criminals, making the technology unjust and the positive outcome unevenly more inclined to rich people rather than everyone regardless of their socio-economic status.

First counter-argument (with rebuttal):

- Counter-argument: Some might say that everyone, even lower-income people, benefits from the convenience of microchips. The technology makes life easier for everyone, regardless of their financial situation.

- Rebuttal: Although convenience improves for everyone, the risks are much greater for lower-income people. Wealthier individuals can protect themselves with better security, but poorer people face higher risks without that option. This will end up having rich people benefit more from the technology, whereas all that the poorer people obtain from it is higher risks instead of benefits, which lead to the inequality in the outcome from microchips and hence resulting in the justice issue.

Second counter-argument (with rebuttal):

- Counter-argument: Even people in high positions can be targeted because their information is valuable, so they aren't completely safe either.

- Rebuttal: While high-status individuals may be targeted, they can afford better protection, which lowers their risks. Poorer individuals have no way to defend themselves, making them easier and more frequent targets. The imbalance between the outcome of the rich and poor people, in which rich people benefits more but it might harm poor people more, makes this technology having fairness issue.

Step 5: Finally, let's end on a more positive note and imagine a "Light Mirror" scenario, where the negative consequences of the technology you have described are averted and positive results are achieved in their stead. Try answering the following questions:

1. What kinds of solutions can be deployed in the immediate for addressing the harms of the technology you have described? What could we do to ensure that we don't get to the negative consequences you imagined later in the future?
2. Could you imagine a scenario where the technology you have described is used with positive consequences, given the appropriate safe guards?

1.

There might be a couple immediate solutions to address the harms of microchips. One would be letting the government provide funds specifically to support the lower income people to upgrade their microchips based on their annual tax filings, just so every citizen regardless of their socioeconomic background could obtain the same level of privacy protection in the application of this technology, and therefore, the fairness issue would be solved since everyone involved in this technology has been treated equally. At the same time, the government should also have a part of the funding to encourage the public service sectors (such as stores, hospitals, police offices, or businesses) to continue using traditional methods for transactions at the same time, just so people are able to choose between opt-in microchips or stay the same as before.

The other would be create user-centered informed consent or terms and conditions for users to easily understand the benefits and risks, to make the technology more transparent to public. At the same time, a smooth opt-out system should also be provided by the microchip manufacturer to ensure the users could withdraw safely and in a timely manner whenever they would like to discontinue. To ensure we do not get to the negative consequences we imagined, having specific laws and guidelines to ensure harsh punishments applied towards chip-related crimes would also be helpful to decrease the incidences of such crimes.

2.

With the help of the government-supported funding, every citizen could afford to upgrade their microchips to the high-level privacy-protected ones now. People like John, who developed PTSD symptoms by watching their families get attacked and become a victim of chip crimes, can also live a carefree life without worrying the traditional method of transaction being completely substituted. If users are unsatisfied with the microchip service, raise safety concerns, or have any emergency situation while using the service, with the newly developed fast opt-out system by the manufacturer, they could also get an immediate opt-out in less than a minute, which prevents their information being hacked/violently attacked and robbed by criminals. One time, John's brother Jason is walking on the street and he notices someone is approaching him with an intent to attack him violently and rob his microchips. He quickly decided to opt-out and clean his information out of the chip system. He also reported the criminal characteristics to the police shortly after. Thank to the systematic law guidelines, the criminal gets punishments accordingly and the news spread among the publics.

With the speedy and highly safe methods of accessing personal information, youth people learned to apply microchips in their every-day life without the need of carrying all their IDs and files included sensitive privacy information. It also decreases the report of identification or physical credit card lost. For senior people, once they get used to the microchip services, such technology eliminates the barrier of them not being able to learn to use mobile payment like Apple Pay or PayPal. Instead, all they need to use is their own wrist with chips embedded.

Sources

The Black Mirror Writers' Room exercises was designed by Dr. Casey Fiesler. Links to her work and publications:

- The Black Mirror Writers Room: The Case (and Caution) for Ethical Speculation in CS Education
- "Run Wild a Little With Your Imagination": Ethical Speculation in Computing Education with Black Mirror

Python Review

In this section of the assignment, we will review useful Python functions and libraries that will allow you to read and analyze data, as well as training simple Machine Learning models.

Section 1: Exploring datasets with Pandas

First, we will need a dataset to work on. Let's use a weather type dataset, a good starting dataset (we will save more interesting cases for later!). Download this dataset from the link to use it for this exercise.

We also need to import the necessary library to read and manipulate our dataset, which is Pandas. The imports are given to you. Next, use the `read_csv()` function to import the data in your workspace. The documentation for this function can be found here.

In [1]:
```
import pandas as pd

df = pd.read_csv("weather_classification_data.csv")
df.head()
```

Out[1]:

	Temperature	Humidity	Wind Speed	Precipitation (%)	Cloud Cover	Atmospheric Pressure	UV Index	Seasor
0	14.0	73	9.5	82.0	partly cloudy	1010.82	2	Winte
1	39.0	96	8.5	71.0	partly cloudy	1011.43	7	Spring
2	30.0	64	7.0	16.0	clear	1018.72	5	Spring
3	38.0	83	1.5	82.0	clear	1026.25	7	Spring
4	27.0	74	17.0	66.0	overcast	990.67	1	Winte

Now, let's use the `describe()` function of the Pandas library to get an overview of the dataset, and answer the following questions (you can write your answers in this box):

- What is the maximum temperature recorded in the dataset? *109.000000*
- What is the average wind speed? *9.832197*

Note: some of the values you will see may appear unrealistic (such as incredibly high temperatures). The dataset is artificially generated and purposefully includes outliers to practice detection and handling, but it is not something we will worry about it this exercise - we are just interested in getting some practice with useful commands.

In [2]:
```
df.describe()
```

Out[2]:

	Temperature	Humidity	Wind Speed	Precipitation (%)	Atmospheric Pressure	
count	13200.000000	13200.000000	13200.000000	13200.000000	13200.000000	1320
mean	19.127576	68.710833	9.832197	53.644394	1005.827896	
std	17.386327	20.194248	6.908704	31.946541	37.199589	
min	-25.000000	20.000000	0.000000	0.000000	800.120000	
25%	4.000000	57.000000	5.000000	19.000000	994.800000	
50%	21.000000	70.000000	9.000000	58.000000	1007.650000	
75%	31.000000	84.000000	13.500000	82.000000	1016.772500	
max	109.000000	109.000000	48.500000	109.000000	1199.210000	1

The `describe()` function is helpful, but it does not answer all the questions we may have. For example, we did not get any idea about the class distribution in our dataset, that is, how many samples we have for each of the four classes (Rainy, Cloudy, Sunny, Snowy). Can you write a line of code to answer this question?

In [3]:

```python
print(df["Weather Type"].value_counts().to_frame())
```

```
              count
Weather Type       
Rainy          3300
Cloudy         3300
Sunny          3300
Snowy          3300
```

Thanks to `describe()`, we know that the minimum temperature recorded is -25 C, but we have no idea which sample it belongs to. Can you write a line of code to find the sample number and also the Weather Type associated to it?

In [4]:

```python
print(df[df["Temperature"]==df["Temperature"].min()]["Weather Type"])
```

```
4609    Snowy
Name: Weather Type, dtype: object
```

Again thanks to `describe()`, we know that 25% of the samples in the dataset have a recorded Precipitation higher that 82 (you can verify this in the output table), but how many of these are Snowy? Answer this question in 1 line of code.

In [5]:

```python
print(len(df[(df['Precipitation (%)'] > 82)&(df['Weather Type']=='Snowy')]))
```

```
1243
```

Finally, sometimes we may be interested in sorting the dataframe by the values in a column. In this cell, sort the dataframe by humidity in descending order, and check the results by printing the first 5 rows.

In [6]:

```
df.sort_values(ascending=False, by="Humidity").head()
```

Out[6]:

	Temperature	Humidity	Wind Speed	Precipitation (%)	Cloud Cover	Atmospheric Pressure	UV Index	S
1303	29.0	109	21.0	93.0	partly cloudy	1018.98	9	\
8716	16.0	109	27.0	102.0	overcast	1007.30	1	\
9707	51.0	109	17.0	98.0	overcast	994.03	8	S
2812	16.0	109	39.0	87.0	partly cloudy	1011.38	11	S
12566	4.0	109	16.0	93.0	overcast	988.15	12	\

As last step of this section, save the sorted dataframe in a new csv file called "weather_data_by_humidity.csv"

In [7]:

```
df.sort_values(ascending=False, by="Humidity").to_csv("weather_data_by_humic
```

Section 2: Training ML models with Scikit-learn

We are now interested in creating a model to predict the weather type based on the features available. Let's see how to do that using the python library Scikit-learn, while reviewing some important concepts about training and evaluating models. Simply run the cells below to see the output and answer the related questions.

First, we need to split our data set into training and testing set. The next cell shows how to do that. We will also separate the Weather Type column (target) from the other columns (features)

In [8]:

```
from sklearn.model_selection import train_test_split

train_df, test_df = train_test_split(df, test_size=0.2, random_state=123)  #

X_train, y_train = train_df.drop(columns=["Weather Type"]), train_df["Weathe
X_test, y_test = test_df.drop(columns=["Weather Type"]), test_df["Weather Ty

X_train.head()  # quick visual check on X_train, the features dataframe
```

Out[8]:

	Temperature	Humidity	Wind Speed	Precipitation (%)	Cloud Cover	Atmospheric Pressure	UV Index	Sea
12987	26.0	45	3.5	10.0	clear	1011.01	7	Aut
905	29.0	71	21.0	86.0	partly cloudy	1013.77	12	Wi
5590	38.0	63	5.5	11.0	clear	1013.87	11	Sp
7269	17.0	66	18.0	63.0	partly cloudy	992.22	1	Wi
1417	32.0	39	7.5	3.0	clear	1021.43	9	Aut

Question for you: creating a testing set is very important when training a model. **Why? How is it used? What would happen if we did not do this very important step?**

Creating a test set is important because it allows us to ensure that the model is generalizable to unseen data. The test set acts as a new dataset and helps determine whether the model performs well on information it hasn't seen before. The test set can only be used after the model finishes its training, hyperparameter tuning, and validation, and it can only be used for once, which is the golden rule of machine learning. Otherwise, if the test set is breached and exposed to the model for repeated times, the measure will no longer be accurate and might lead to an artificially good result but it will actually perform poorly in real-world practice. Without a test set, if we only used the training data, including the validation data, to evaluate the model, we would not be able to compare different models and have a dataset to objectively evaluate their performance. We also wouldn't be able to compare the difference between a model's performance on training data and that on the unseen data. Moreover, if we use the validation set for both tuning hyperparameters and evaluating the final model, the evaluation will be biased and overly optimistic as the model is sort of trained on the validation.

As you can see, the dataset includes categorical features. Most classifiers require categorical features to be transformed before they can be used for training and prediction. The code below uses One Hot Encoding to convert the categorical features Cloud Cover, Season and Location, while leaving the numberical features unchanged.

This is a simple example of data preprocessing. Preprocessing can be more extensive (for example, including scaling of numerical features), but we are only interested in an overview of the fundamentals, so we will just apply One Hot Encoding to make the data usable.

In [9]:

```
from sklearn.preprocessing import OneHotEncoder
from sklearn.compose import make_column_transformer

passthrough = ['Temperature', 'Humidity', 'Wind Speed', 'Precipitation (%)',
                'Atmospheric Pressure', 'UV Index', 'Visibility (km)']
categorical = ['Cloud Cover', 'Season', 'Location']


ct = make_column_transformer(
    (OneHotEncoder(), categorical),  # OHE on categorical features
    ("passthrough", passthrough),  # no transformations on the numberical fe
)

# Fit the encoder on the training data and transform
train_encoded = ct.fit_transform(X_train)

# Transform the test data
test_encoded = ct.transform(X_test)

# Convert the encoded data back to DataFrame for better readability

column_names = (
    ct.named_transformers_["onehotencoder"].get_feature_names_out().tolist()
)

X_train_encoded = pd.DataFrame(train_encoded, columns=column_names)
X_test_encoded = pd.DataFrame(test_encoded, columns=column_names)
```

In [10]:

```
# Run this cell to see what the encoded data set looks like

X_train_encoded
```

Out[10]:

	Cloud Cover_clear	Cloud Cover_cloudy	Cloud Cover_overcast	Cloud Cover_partly cloudy	Season_Autumn	Se
0	1.0	0.0	0.0	0.0	1.0	
1	0.0	0.0	0.0	1.0	0.0	
2	1.0	0.0	0.0	0.0	0.0	
3	0.0	0.0	0.0	1.0	0.0	
4	1.0	0.0	0.0	0.0	1.0	
...	...	...	...	...	...	
10555	0.0	0.0	0.0	1.0	0.0	
10556	0.0	0.0	1.0	0.0	0.0	
10557	1.0	0.0	0.0	0.0	0.0	
10558	1.0	0.0	0.0	0.0	0.0	
10559	0.0	0.0	0.0	1.0	0.0	

10560 rows × 18 columns

Question for you: It appears that we applied the same One Hot Encoding transformation to both training and test set. Why did we bother doing this operation on the separate sets? Could have we just transformed the original dataframe `df`, and then split it in training and test set?

We apply OHE separately to the training and test sets to prevent data leakage, which could lead to overly optimistic evaluation results. If the model is exposed to data it should not have seen during training, it may perform better than it actually should. Therefore, we cannot simply transform the original dataframe before splitting, as it would lead to unreliable evaluation results.

There are many classifiers we can choose from. We will use Decision Trees to start. Decision trees are very simple classification algorithms, although they have typically mediocre performance on complex classification problems.

A certain level of familiarity with Decision Trees is expected in this course. You may want to review the material from your previous courses, or this introduction.

In [11]:

```python
from sklearn.tree import DecisionTreeClassifier
from sklearn import tree
from sklearn.tree import plot_tree
import matplotlib.pyplot as plt


model = DecisionTreeClassifier() # Create a decision tree
model.fit(X_train_encoded, y_train) # Fit a decision tree
```

Out[11]:

```
▾ DecisionTreeClassifier
DecisionTreeClassifier()
```

Now that we have the tree, we want to see how well it performs. Let's first check the accuracy on the training set:

In [12]:

```python
model.score(X_train_encoded, y_train) # Score the decision tree
```

Out[12]:

```
1.0
```

100% accuracy!!!

...

This sounds too good to be true... let's check the test set to see how well the trees perform on unseen samples:

In [13]:

```python
model.score(X_test_encoded, y_test) # Score the decision tree on test set
```

Out[13]:

```
0.9121212121212121
```

Accuracy dropped significantly when we moved to unseen samples!

This is because the Decision Tree, if left unsupervised, is very prone to **overfitting**.

Question for you: what does it mean for a model to overfit?

Overfitting happens when a model learns the training data too well, including unnecessary details. This makes the model perform well on training data but poorly on unseen data. It can't generalize properly.

To prevent a model from overfitting, we tune its **hyperparameters.** Hyperparameters are like knobs that we can use to regulate the way a model learn.

Some hyperparameters for the scikit-learn DecisionTreeClassifier include:

- max_depth: the maximum distance between the root node and a leaf node
- min_samples_split: the minimum number of samples required to split an internal node
- min_samples_leaf; the minimum number of samples required to be at a leaf node

You can look up other hyperparameters and their default values in the DecisionTreeClassifier documentation. By default, the maximum depth value is set to *None*, that is, the tree is free to grow until it has parfectly classified all samples. As we have seen, this results in perfect accuracy on the training set, but much lower accuracy on unseen samples.

Run the cell below to see the depth of our overfitted tree:

In [14]:

```
model.get_depth()
```

Out[14]:

```
19
```

If we could find the right depth for our tree, we could reduce the problem of overfitting.

Question for you: what would happen if we reduce the depth of the tree *too much*? What do you expect the accuracy on training and test set to look like in this case?

The test accuracy might temporarily increase, as it avoids overfitting on the training data and generalizes better on unseen data compared to deeper tree model which might be overfitting on the train data. However, the train accuracy would decrease, because the model is too simple to fit the training set. If we reduce the depth of the decision tree too much, the model will eventually underfit. This causes lower accuracy on both the training and test sets eventually.

Hyperparameter tuning is typically done on a **validation set.** A validation set is a set of samples not used for training, like the test set, but unlike the test set, we are allowed to use this multiple times as we look for the best hyperparameter values.

Because our data set is rather small, it is not great to take more samples from the training set to create a validation set, because:

- We would have fewer samples (less information) to train our model
- The validation set would also be small, and result in a highly variable accuracy measure (meaning if we run the experiment again changing the samples in each set, we will likely get very different results)

There is a method that we can use to eliminate both problems, called ***k-_fold cross-validation.** Cross-validation iteratively separates training and validation set (_k* times), so we get multiple measures of accuracy on the validation sets, which can be averaged for a more stable result. A good understanding of how cross-validation works is important for any data scientist. I encourage you to review cross-validation from previous courses, or this introduction video (courtesy of Dr. Kolhatkar).

Scikit-learn has a great method that we can use to perform cross-validation and find the best hyperparameters for a model at the same time, called GridSearchCV. Let's use it to find the best depth for our Decision Tree:

In [15]:
```
from sklearn.model_selection import GridSearchCV
import numpy as np  # to create the array of values for depth

param_grid = {
    "max_depth": np.arange(1, 20, 1)  # testing all depths from 1 to 19
}

grid_search = GridSearchCV(
    model, param_grid, cv=10, n_jobs=-1, return_train_score=True   # 10-fold
                                                                   # depths
)
grid_search.fit(X_train_encoded, y_train)
```

Out[15]:
```
▸             GridSearchCV
▸ estimator: DecisionTreeClassifier
        ▸ DecisionTreeClassifier
```

In [16]:
```
grid_search.best_score_
```

Out[16]:
```
0.9115530303030303
```

In [17]:
```
grid_search.best_params_
```

Out[17]:
```
{'max_depth': 12}
```

Complete the sentence (replace --?--): Among all possible trees, GridSearchCV picked a tree of depth **12**, with an average validation accuracy of

0.9115530303030303.

The accuracy on the training set is no longer 100%, but we expect this tree to perform better on unseen samples. Let's try it on our test set:

In [18]:

```
best_tree = grid_search.best_estimator_

best_tree.score(X_test_encoded, y_test) # Score the decision tree on test se
```

Out[18]:

```
0.9162878787878788
```

The accuracy is similar to when the model was overfitting, but hyperparameter tuning brought us 2 advantages:

- We had a more realistic expectation of what our accuracy was going to be (closer to 91%, not 100%)
- We simplified the model and reduced its depth. This makes the model faster and easier to visualize.

Question for you: on what samples (or portion of samples) of `X_train_encoded` was the final model (`best_tree`) trained on?

The final model was trained on the entire training set (= `X_train_encoded`)

The model can now be used to get predictions for unseen samples. For example:

In [19]:

```
random_sample = X_test_encoded.sample(n=1, random_state=42)

random_sample
```

Out[19]:

	Cloud Cover_clear	Cloud Cover_cloudy	Cloud Cover_overcast	Cloud Cover_partly cloudy	Season_Autumn	Sea
2005	1.0	0.0	0.0	0.0	0.0	

In [20]:

```
best_tree.predict(random_sample)
```

Out[20]:

```
array(['Sunny'], dtype=object)
```

Final thoughts

1. If you have completed this assignment in a group, please write a detailed description of how you divided the work and how you helped each other completing it:

We did Question 1 together in class. And then Ayuho finished the rest everything but Step 5 of Question 3. Muhan finished Q3 Step 5 and did cross-check for Ayuho's answer, providing additional portion of answers for Q2-3.

2. Have you used ChatGPT or a similar Large Language Model (LLM) to complete this homework? Please describe how you used the tool. **We will never deduct points for using LLMs for completing homework assignments,** but this helps us understand how you are using the tool and advise you in case we believe you are using it incorrectly.

We used ChatGPT to help us brainstorm the general idea of potential solutions to the harms in Q3 Step 5.

3. Have you struggled with some parts (or all) of this homework? Do you have pending questions you would like to ask? Write them down here!

Not for now!

In [ ]: