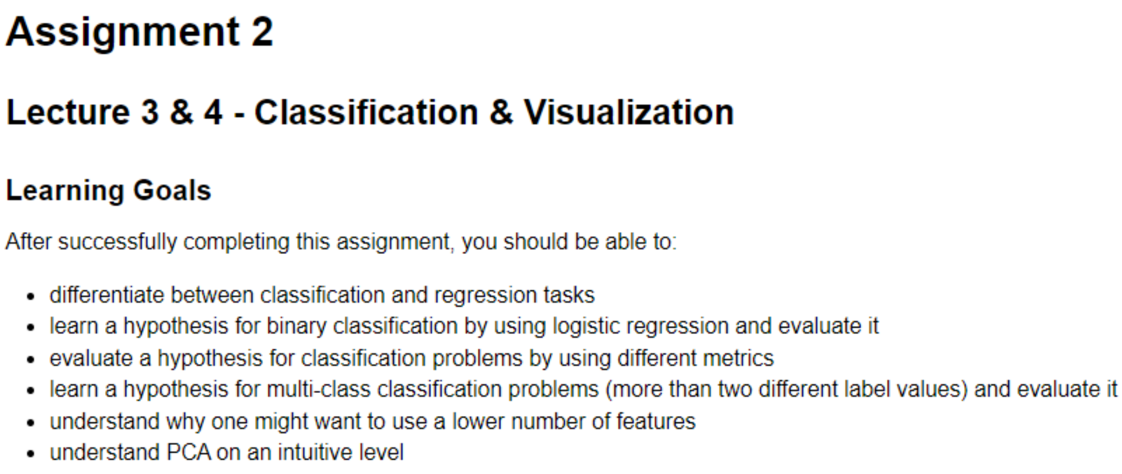
Hi, these are my feedback for assignment 2



Assignment 2

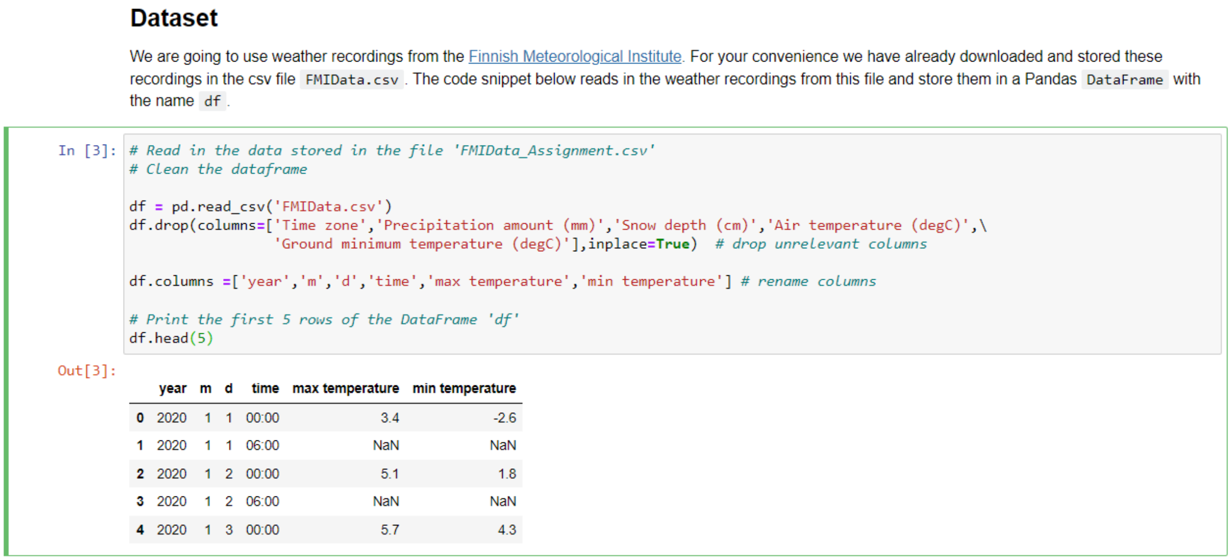
Lecture 3 & 4 - Classification & Visualization

Learning Goals

After successfully completing this assignment, you should be able to:

- differentiate between classification and regression tasks
- learn a hypothesis for binary classification by using logistic regression and evaluate it
- evaluate a hypothesis for classification problems by using different metrics
- learn a hypothesis for multi-class classification problems (more than two different label values) and evaluate it
- understand why one might want to use a lower number of features
- understand PCA on an intuitive level

- learn a hypothesis for binary classification by using logistic regression and evaluate it: can we be more specific on what is "it" here? Is it the hypothesis, the binary classification or the logistic regression?
- understand the Principle Component Analysis (PCA) on an intuitive way. I think we should write it in the full form when we mention it.



Dataset

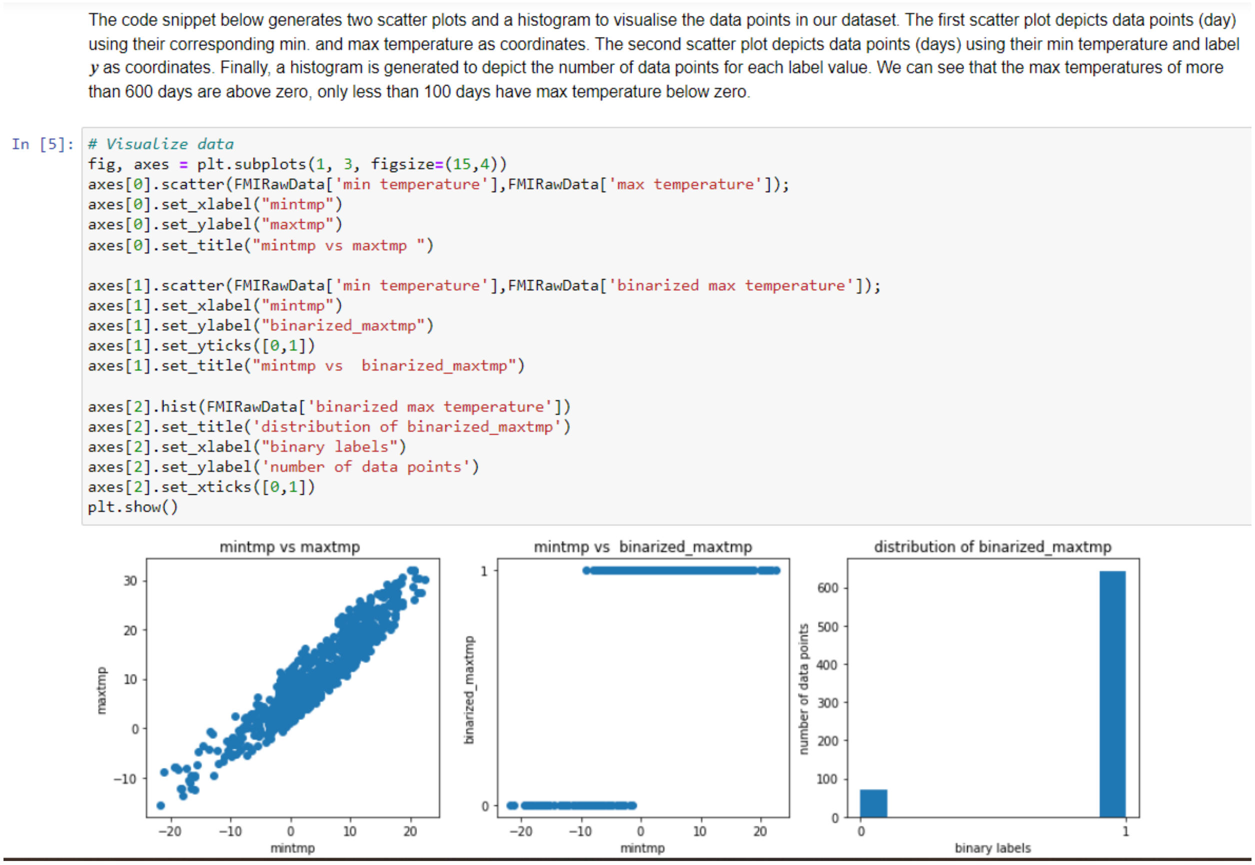
We are going to use weather recordings from the Finnish Meteorological Institute. For your convenience we have already downloaded and stored these recordings in the csv file `FMIData.csv` . The code snippet below reads in the weather recordings from this file and store them in a Pandas `DataFrame` with the name `df` .

In [3]:
```
# Read in the data stored in the file 'FMIData_Assignment.csv'
# Clean the dataframe

df = pd.read_csv('FMIData.csv')
df.drop(columns=['Time zone','Precipitation amount (mm)','Snow depth (cm)','Air temperature (degC)',\
                 'Ground minimum temperature (degC)'],inplace=True)  # drop unrelevant columns

df.columns =['year','m','d','time','max temperature','min temperature'] # rename columns

# Print the first 5 rows of the DataFrame 'df'
df.head(5)
```

Out[3]:

	year	m	d	time	max temperature	min temperature
0	2020	1	1	00:00	3.4	-2.6
1	2020	1	1	06:00	NaN	NaN
2	2020	1	2	00:00	5.1	1.8
3	2020	1	2	06:00	NaN	NaN
4	2020	1	3	00:00	5.7	4.3

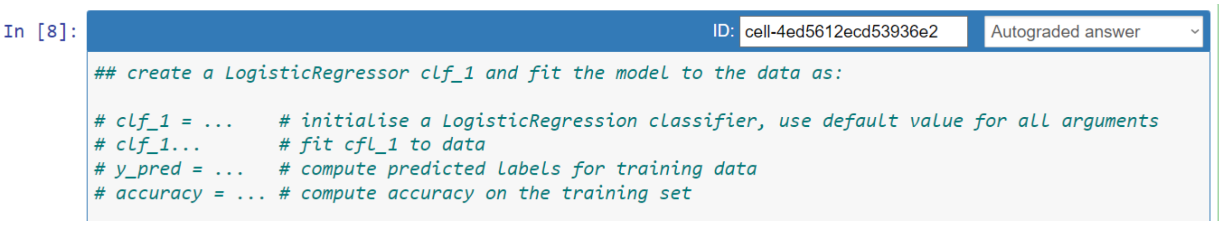
For the Dataset, I think we can write df.head(5) for the original df after df = pd.read_csv('FMIData.csv') so students have a general view of the data and how the data cleaning affects the dataframe subsequently.



The code snippet below generates two scatter plots and a histogram to visualise the data points in our dataset. The first scatter plot depicts data points (day) using their corresponding min. and max temperature as coordinates. The second scatter plot depicts data points (days) using their min temperature and label *y* as coordinates. Finally, a histogram is generated to depict the number of data points for each label value. We can see that the max temperatures of more than 600 days are above zero, only less than 100 days have max temperature below zero.

In [5]:
```
# Visualize data
fig, axes = plt.subplots(1, 3, figsize=(15,4))
axes[0].scatter(FMIRawData['min temperature'],FMIRawData['max temperature']);
axes[0].set_xlabel("mintmp")
axes[0].set_ylabel("maxtmp")
axes[0].set_title("mintmp vs maxtmp ")

axes[1].scatter(FMIRawData['min temperature'],FMIRawData['binarized max temperature']);
axes[1].set_xlabel("mintmp")
axes[1].set_ylabel("binarized_maxtmp")
axes[1].set_yticks([0,1])
axes[1].set_title("mintmp vs  binarized_maxtmp")

axes[2].hist(FMIRawData['binarized max temperature'])
axes[2].set_title('distribution of binarized_maxtmp')
axes[2].set_xlabel("binary labels")
axes[2].set_ylabel('number of data points')
axes[2].set_xticks([0,1])
plt.show()
```

Can we give the markdown description "The first scatter plot depicts… only less than 100 days have max temperature below zero." below the figures? I think this canwork as the caption for the figures and conventionally, the caption should be directly below the figures.



In [8]:

ID: cell-4ed5612ecd53936e2 | Autograded answer

```
## create a LogisticRegressor clf_1 and fit the model to the data as:

# clf_1 = ...    # initialise a LogisticRegression classifier, use default value for all arguments
# clf_1...       # fit cfl_1 to data
# y_pred = ...   # compute predicted labels for training data
# accuracy = ... # compute accuracy on the training set
```

Since we use clf_1 as the variable for testing, I believe we should write instructions here: Please use these variables name and do not change them. Because technically if the students use other variable names but they still have the correct workflow, they still get 0 points.