To estimate the validity of the predictions we propose to use two different measures: the coefficient of determination (R^2), which shows the skill of the mean prediction; and the reliability, which measures the accuracy of the spread in the prediction.

These two measures can be mathematically expressed as:

$$R^2 = \frac{\sum_{t,p}\left(\overline{m_i(t,p)}^{i} - o_(t,p)\right)^2}{\sum_{t,p} o(t,p)^2}$$

$$Reliability = \sqrt{\overline{\left[\frac{\left(\overline{m_i(t,p)}^{i} - o(t,p)\right)^2}{\langle m_i(t,p)\rangle}\right]}^{t,p}}$$

$$score = -\left[\log\left(R^2\right) + \left|\log\left(Reliability\right)\right|\right]$$

where t is time, p is the pixel, i is the possibility or state index, $o(t, p)$ are the observation, and m_i are the predicted possibilities. The bar denotes an average over time or sum over possibilities depending on the superscript. The prediction provides:

$\overline{m_i(t,p)}^{i}$ that is the mean predicted value,

$\langle m_i(t,p)\rangle$ that is the variance of the predicted value.

When a reliable prediction has large skill ($-\log(R^2) >> 0$) we expect the prediction uncertainty to be small (Reliability$\sim$1). On the other hand, when a reliable prediction system has low skill we expect the prediction uncertainty to be as big as the observed variance. In this context, and regardless of its skill, a reliable prediction system always needs to have a log reliability close to 0. A wrong reliability estimation will lead to a large absolute value of its logarithm.

The final score is the negative sum of the two logarithms. Bigger score means the best skill and an accurate reliability.

It is possible to have a look on the data using the attached python script. For instance the next command line shows the 10 years of the model 6 and the corresponding prediction:

```
>python show_model.py train_X.csv 6 0
```

Here is the `show_model.py` python script:

```
"""
Show the model data
"""

import numpy as np
import sys
import healpy as hp
import matplotlib.pyplot as plt

def show_model(dataframe_x, id_model,id_data):
    """

    Args
        dataframe_: Pandas Dataframe
            Dataframe containing the learning database values.
            This dataframe was obtained by reading a csv file with following instruction:
            dataframe_base_x = pd.read_csv(CSV_1_FILE_PATH, index_col=False, sep=',')
            5 columns :
            - DATASET: Define the dataset id. database stores several consistent dataset. Each dataset are independant
                       a dataset is composed by:
                    * 3072 temperature anomalies within the all world from 22 models (model id from 1 to 22) during 10 years
```

```
                    * 3072 temperature anomalies within the all world from the observation (model id = 0)during 10 years.
                    * the predicted 192 temparature anomalies within the all world for the 22 models
            - MODEL: model id (1-22) for models and (0) or the observation
            - TIME: id of the time (0-9) for the 10 year history and 10 for the predicted date.
            - POSITION: earth coordinate in healpix (nside=4 for prediction, nside=16 for history) the ordering is in nested
            - VALUE : the corresponding temperature anomalies

    Returns
        test: -1 if something goes wrong
    """

    # RETURN -1 if the dimension of the dataframe are not the proper one
    nline,ncol=dataframe_x.shape
    print(ncol)
    print(nline,ncol,10*3072*23+192*22)
    if ncol!=5:
        return -1E30,-1E30
    if nline%(10*3072*23+192*22)!=0:
        return -1E30,-1E30

    ndata=nline//(10*3072*23+192*22)
    print('%d dataset in the dabase'%(ndata))

    iddata  = np.array(dataframe_x['DATASET'])
    idmodel = np.array(dataframe_x['MODEL'])
    idtime  = np.array(dataframe_x['TIME'])
    idpos   = np.array(dataframe_x['POSITION'])
    value   = np.array(dataframe_x['VALUE'])

    #look for the 10 first years of the 22 models
    idx = np.where((iddata==id_data)*(idmodel==id_model)*(idtime<10))[0]
    if len(idx)<1000:
        print('id data(%d) or id model (%d) unknown',id_data,id_model)
        return -1
    model=np.zeros([10,3072])
    model[idtime[idx],idpos[idx]]=value[idx]

    #look for the  years of the 22 models
    idx = np.where((iddata==id_data)*(idmodel==id_model)*(idtime==10))[0]
    pred_model=np.zeros([192])
    pred_model[idpos[idx]]=value[idx]

    plt.figure(figsize=(10,3))
    for  i in range(10):
        hp.cartview(model[i,:],cmap='jet',hold=False,sub=(2,5,1+i),title='Time=%2d'%(i),
                    nest=True,margins=(0,0.05,0,0.05))

    plt.figure(figsize=(6,3))
    hp.cartview(pred_model,cmap='jet',hold=False,sub=(1,1,1),title='Prediction',
                    nest=True,unit=r'$\Delta Kelvin$',margins=(0,0.05,0,0.05))
    plt.show()

    return 0


if __name__ == '__main__':
    import pandas as pd
    CSV_FILE_X = sys.argv[1]
    model = int(sys.argv[2])
    data = int(sys.argv[3])
    df_x = pd.read_csv(CSV_FILE_X, index_col=0, sep=',')
    print(show_model(df_x,model,data))
```