Problem Set 3

D. Jason Koskinen

koskinen@nbi.ku.dk

Advanced Methods in Applied Statistics

Feb - Apr 2024

Photo by Howard Jackman

Info

- The submission is:
  - A write-up as a PDF document, which includes any plots, diagrams, tables, pictures, and explanations
  - In a separate "file", submit all code used to derive the results
    - Tarball, zipped directory, lots of individual files w/ self-explanatory titles, etc.
    - Do NOT include lines of code in your write-up. If results are dependent on coding choices then include those comments in the write-up.
- Include any original data files or how the data was accessed
  - If you use a internet scraping tool, note the date when you retrieved the data
  - If you can save the data to a file, do so and submit the data file. There is no need to change the format, e.g. HTML, XML, txt, JSON…

Problem 1 (4 pts.)

- Census data collected in the 1990s of working adults in many countries can be used as a data set to establish earning potential
- Create a classifier which separates lower income earners (≤50k) from higher income earners (>50k)
    - See further criteria for training requirements on the next slides (Problem 1a has 2 slides which should be read through entirely before starting)
- The data set has been divided:
    - Training/Testing data set is at http://www.nbi.dk/~koskinen/Teaching/data/earning_potential_train_test.txt
    - The analysis data set is at http://www.nbi.dk/~koskinen/Teaching/data/earning_potential_real.txt
        - Only used in problem *part c*
    - Include **both input files** when submitting your solution

Problem 1a

- Using your classifier, what is the **precision** of selecting earners with >50k if the selection must contain less than 15% of earners with ≤50k.

Positive = classified = earn more than 50k
TP/(TP+FP) > 0.85
FP/(TP+FP) < 0.15

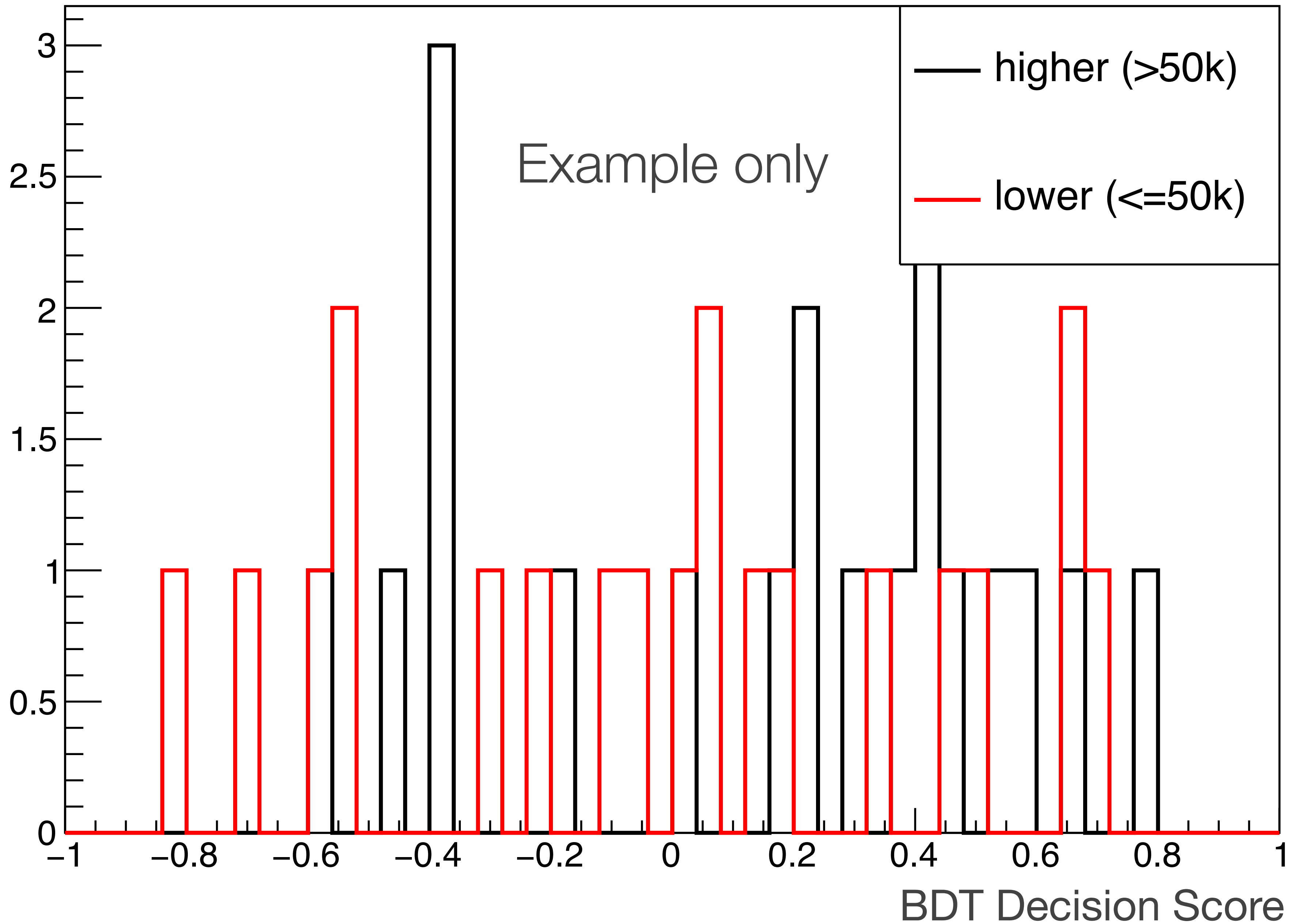
  - Classified sample is at least 85% high-earners and at most 15% low-earners for individuals above some cut on the classification score
    - Ensure there are at least 500 low earners and 500 high earners in a validation/test data set that are **NOT** included in the training data set
  - The precision is the fraction of 'true positives' properly classified out of the total number of 'positives'
  - https://en.wikipedia.org/wiki/Confusion_matrix
- Use the training set to establish the precision
  - Note that you'll have to use **this** trained algorithm on an unknown data set in *part c*. So be mindful about overtraining to get excellent purity and efficiency for *part a* only to get penalized in *part c* when you have use the algorithm on unknown data.

Problem 1a (cont.)

- Make a histogram plot using **at least** 500 low earners and **at least** 500 high earners from the train_test.txt file as a function of their decision score.
  - Use 500 low earners and 500 high earners that were **not** part of the training sample
  - Separate the two populations and plot the high earners in *black* and low earners in *red*
  - If your method does not have a ‘decision score’ then plot each earner as a function of the test statistic which is used for the classification

Problem 1a (example)

- Example here is shown for only 20 entries



Problem 1b

- Rank the variables starting with most important to least important
  - Discuss any variables that have similar discrimination power
  - Provide the ranked list
- Discuss how to identify and avoid overtraining supervised machine learning algorithms

Problem 1c

- Using the same classifier developed in Problem 1a, run the classifier over all the entries on the blind/real sample (earning_potential_real.txt)
    - The new data file has an additional first column which is the ID number of the earner
    - Produce a text file which contains **only** the IDs which your classifier classifies as **low earners** (your_name.low_ID.txt)
    - Produce a text file which contains **only** the IDs which your classifier classifies as **high earners** (your_name.high_ID.txt)
    - Basic text files. No Microsoft Word documents, Adobe PDF, or any other extraneous text editor formats. Only a single ID number per line in the text file that can be easily read by numpy.loadtxt().
        - One entry per line and no commas, brackets, parenthesis, etc.

Problem 2 (4 pts.)

- There is data regarding crashes in the town of Cary, North Carolina, U.S.A. over the span of years
  - Presumably they are car crashes
  - We will be using specific variables in the data
    - Longitude="lon", Latitude="lat", and the time/data field is "crash_date"
    - Negative longitude implies 'West', whereas positive implies 'East' from a spot in Greenwich Park in London, England

- https://www.nbi.dk/~koskinen/Teaching/data/cpd-crash-incidents.csv
  - Use all the data in the file

Problem 2a (2 pts.)

- Create a scatter plot of ALL the crash entries as a function of the latitude and longitude (in decimal degrees)
  - https://en.wikipedia.org/wiki/Decimal_degrees
  - Longitude on x-axis and latitude on y-axis
- Make a histogram of the time of day of each crash where the x-axis goes from 0-24 hours.
  - Bin width is 1 hour
  - Lowest bin edge starts at 00:00 (for the HH:MM time format)
- Describe how would you create a kernel density estimation using a gaussian kernel with a bandwidth of 0.25 hours to produce a probability density function of the time of day for crashes
  - This is a qualitative description. Feel free to use written text, diagrams, hand-drawn plots, or any other illustrative tool to describe the KDE PDF generation.
  - N.B. For a 24-hour clock, time 'wraps' at 00:00 & 24:00. For example, 00:01 and 23:59 are very close to each other in actual time, but not numerically.

Problem 2b (1 pt.)

- Create a kernel density estimation using an Epanechnikov kernel with a bandwidth of 0.8 hours to produce a probability density function of the time of day for crashes
- Make a plot of the PDF constructed by the KDE
- In a table in the write-up, provide the evaluation of the PDF at the following times [00:23, 01:49, 08:12, 15:55, 18:02, 21:12, 23:44]
- Using only the time KDE PDF, if additional police officers patrolling the roads reduced the relative crash rates by 10% for a duration of 2 continuous hours, what 2-hour window would be the best to patrol? How much would the 24-hour percentage of crashes be reduced for your choice of 2-hour window of additional patrols?
    - Written slightly differently: the KDE PDF describes the relative crash likelihood during a 24-hour window. If additional police patrolled the entire 24-hour time, then there would be a 10% decrease in the likelihood of accidents. So what is the percentage of the 24-hour daily crashes that will be reduced by your 2-hour window?

Problem 2c (1 pt.)

- Create a 2-dimensional KDE PDF from the crash data using the latitude and longitude with an Epanechnikov kernel and a bandwidth of 0.01 in both dimensions
  - This is time independent, so include all times
  - Make a contour plot of the KDE PDF with the longitude on the x-axis and latitude on the y-axis. Include an appropriate color bar for the plot which shows numerical value of the color **or** include labels on plotted contour lines.

- What is the total percentage of crashes estimated by the KDE PDF to be within the 'box' of longitude range of [-78.76, -78.72] and latitude range of [35.74, 35.78]

Problem 3 (2 pts.)

- Consider an experiment set up to measure the lifetime of an unstable nucleus, N, using the reaction: $A \rightarrow N e \bar{\nu}$, $N \rightarrow Xp$
- The creation and subsequent decay of N has a signature of an electron and proton. The lifetime of each N, which follows the PDF $f = \frac{1}{b} e^{-t/b}$, is measured from the time, observing the electron and proton with a gaussian resolution of σ_t
  - Normally the lifetime would be represented by 'τ' instead of 'b', but this becomes a disaster when dealing with t, t', and τ
- The expected PDF is then the convolution of the exponential decay and the gaussian resolution:

$$f(t; b, \sigma_t) = \int_0^\infty \frac{e^{-\frac{(t-t')^2}{2\sigma_t^2}}}{\sqrt{2\pi}\sigma_t} \frac{e^{-t'/b}}{b} dt'$$

Problem 3(cont.)

- Neither b nor σ_t are explicitly known, and we want to test whether b=1 *second* can be rejected. We can do so via a hypothesis test, where the two hypotheses H_0 and H_1 are given as:

$$
\begin{aligned}
b_0 &= 1.0\ s \\
H_0 &: b = b_0 \\
H_1 &: b \neq b_0
\end{aligned}
$$

- Use the likelihood ratio test:

$$
\lambda = \frac{\mathcal{L}(\hat{\omega})}{\mathcal{L}(\hat{\Omega})} \qquad \begin{array}{l} \omega \text{ given by } b = b_0,\ 0 < \sigma_t < \infty \\ \Omega \text{ given by } 0 < b < \infty,\ 0 < \sigma_t < \infty \end{array}
$$

- Where $\mathscr{L}(\hat{\omega})$ is the value of the null hypothesis likelihood calculated using the maximum likelihood estimator(s) $\hat{\omega}$

Problem 3a (1 pt.)

- There are 20000 events in the online file below, which corresponds to 100 simulated pseudo-experiments where each pseudo-experiment has 200 events
- For each of the 100 pseudo-experiments find the values of the $\ln$-likelihoods that are maximized for the two hypotheses
- As a histogram, plot the values of $-2\ln(\lambda)$
- The data is at http://www.nbi.dk/~koskinen/Teaching/data/NucData.txt

Problem 3b (1 pt.)

- Is the distribution of $-2\ln(\lambda)$ chi-squared distributed?
  - Be sure to use the correct number of degrees of freedom
  - Justify and explain your answer
- How many pseudo-experiments have $-2\ln(\lambda) > 2.706$?
- Is the number of pseudo-experiments with $-2\ln(\lambda) > 2.706$ consistent with the expectation from a chi-squared distributed test-statistic and 100 data 'points'?