BUAN / ECON 381

Dr. Levkoff

Predictive Analytics Supervised Learning Project Instructions

The purpose of this project is to demonstrate your ability to work through the analytics pipeline from beginning to end. The first part of your job is to showcase your ability to identify, extract, clean and preprocess data to be used in a predictive (supervised learning) application. The second part of your job is to apply predictive modeling to your dataset by considering techniques introduced in class (or others) to two types of supervised learning problems – a regression problem (where the output is a numerical scalar value) and a classification problem (where the output is qualitative / categorical variable).

TASK I: CHOOSING DATA

1) Groups – you will work on this project in groups of three (there will be one group of four). I will provide some class time to form groups yourselves. If you have trouble finding a group, let me know and I can assign you to one.
2) You will collect a business or economics relevant dataset -you should clear this with the instructor before proceeding to the steps below. The data can be observational data found online (recommended) or experimental data generated or collected by you and your team (the latter is not the "low hanging fruit" for the project). You can consider data from multiple sources to be able to merge datasets and transform datasets to get it in the appropriate format. It is *highly recommended* that you use cross-sectional data for this project. It is also recommended that you think about a predictive question in looking for your data that interests you and your group rather than just looking for data that satisfies the project constraints but otherwise, isn't motivated by a question of interest. You will need instructor approval of your dataset prior to moving forward to the next steps – see constraints below in the next bullet point.
3) If you have time series or panel data, there are ways you can turn these into cross-sectional datasets by extracting or aggregating data to eliminate the time series dimension. You will need at least 8 variables in your dataset. Since this is a course on "big data," you should have a minimum of 500 observations in your dataset (the more the merrier – if you can't quite meet this constraint, but otherwise have interesting data, I may be able to approve this exception).
4) One variable should be a continuous output variable that you plan to predict for the regression output task and should have a moderate to strong correlation with at least one other numeric variable (if not you need to choose a different dataset as predictions won't be very good). One variable should be a categorical variable (with at least two or three classes) that you plan to predict in the classification output task. You should *otherwise* have at least 5 numeric variables and 3 categorical/qualitative variables in your dataset (for a *minimum* of 10 variables in the dataset – you may decide not to use all of them in conducting your analysis). There is no maximum requirement. You may also use two separate datasets for each task if you would prefer, but that is your choice to make.

TASK 2: DATA ANALYSIS / PREDICTIVE MODELING

I. REGRESSION TASK(S)

1) Begin by partitioning your original dataset for the regression task into three sets: a training partition and two holdout partitions. One holdout partition will be for model selection (the validation partition). One partition will be used to report the final model out-of-sample performance estimate (the testing partition). I would recommend saving these three partitions using the write.table() or write.csv() commands (or making sure you're all using the same random seed to generate the partitions) so that all group members start with the *same* training partition and *same* holdout partitions so everyone is making "apples to apples" comparisons. Failure to do this correctly may nullify all subsequent results. I would also *highly* recommend familiarizing yourself with how to use Github to make data sharing easier amongst your group – there is a supplementary module on Canvas with an instructional video and a document with useful CLI commands for Github.
2) Decide on a set of metrics with your group that *everyone* will use when benchmarking the models for the regression task for both in-sample and out-of-sample performance.
3) Verify using a correlation matrix to verify that you have *at least one moderate to strong relationship* between the numeric output variable and a potential model input variable. If you can't meet this condition, go back to **TASK 1** and choose a more appropriate dataset.
4) (Bivariate Regression Modeling) Build a bivariate (one y-variable, one x-variable) regression model using only your *best* predictor from the previous step. All of the sub-parts of this step should consider models predicting the *same output* (y) from the *same inputs* (x) for the best "apples-to-apples" comparison.
    a) Start with a simple linear model with no other transformations of any variable. Compute an in (on the training partition) and out-of-sample (on the validation partition) error metric (be consistent per item 2) above) for benchmarking the model performance. Be sure to interpret estimated model parameters and diagnostic information regarding in-sample fit and significance.
    b) Build a bivariate model that incorporates a nonlinear transformation(s) of the x-variable and benchmark the performance against the linear model in a) (ie: polynomial basis, log transforms, etc) Be sure to interpret estimated model parameters and diagnostic information if possible.
    c) Can you regularize the model in b) (hint: you should have at least two inputs – perhaps a linear term and perhaps a nonlinear transformation)? Can you tune the regularization parameter to improve the model's out-of-sample performance?
    d) Implement a model using the *generalized additive structure*. Try Poisson regression, Quasi Poisson Regression, or SPLINE and benchmark against the models in a) and b).
    e) Create a bivariate plot (using any potting system in R you'd like) to plot each of the models you've estimated against the training and holdout partitions, all on the same diagram. Color each of the models differently so it is easy to see / label.

f) Create a table summarizing the in-sample and out-of-sample estimated performance for each of the models above from 4). Use the validation partition to choose your best model based on the estimated out-of-sample performance on the validation set. Use the uncontaminated data partition to report the *uncontaminated* out-of-sample error on the final model chosen from the validation process.

5) (Multivariate Regression Modeling) Build what you think is the *best* version of the model that you can where you are still predicting the *same* output variable from above, but now you may consider the set of *any inputs* (which may also be different across the models below). You will likely still incorporate your best predictor, but may decide to use others.
   a) Start by building a linear model without incorporating any non-linear transformation. This model may include categorical dummy variables as predictors. Compute an in (on the training partition) and out-of-sample (on the validation partition) error metric (be consistent per item 2) above) for benchmarking the model performance. Be sure to interpret estimated model parameters and diagnostic information regarding in-sample fit and significance.
   b) Implement a form of regularization for your model in a). Are you able to tune the model to improve its performance by applying regularization? Explain. Benchmark the regularized model against the others.
   c) Compare the two models above to a model that incorporates non-linear features transformations (but no regularization) of your input variables (ie: polynomial, SPLINE, log, etc.). Benchmark the in and out-of-sample performance against the models above.
   d) Estimate a support vector machine - you can consider the same or different inputs relative to the models above. Report the in-sample and out-of-sample performance to benchmark the model relative to those above. Are you able to improve predictive performance by *tuning* your model?
   e) Estimate a regression tree - you can consider the same or different inputs relative to the models above. If you are able to produce a reasonably sized tree for a person to look at, display the decision tree. Report the in-sample and out-of-sample performance to benchmark the model relative to those above. Are you able to improve predictive performance by *tuning* your model?
   f) Estimate a tree-based ensemble model - you can consider the same or different inputs relative the models above. Report the in-sample and out-of-sample performance to benchmark the model relative to those above. Are you able to improve predictive performance by *tuning* your model?
   g) Create a table summarizing the in-sample and out-of-sample estimated performance for each of the models above from 5). Use the validation partition to choose your best model based on the estimated out-of-sample performance on the validation set. Use the uncontaminated data partition to report the *uncontaminated* out-of-sample error on the final model chosen from the validation process.

II. CLASSIFICATION TASK(S)

6) Begin by partitioning your original dataset for the classification task (if different from the regression task) into three sets: a training partition and two holdout partitions. One holdout partition will be for model selection (the validation partition). One partition will be used to report the final model out-of-sample performance estimate (the testing partition). I would recommend saving these three partitions using the write.table() or write.csv() commands (or making sure you're all using the same random seed to generate the partitions) so that all group members start with the *same* training partition and *same* holdout partitions so everyone is making "apples to apples" comparisons. Failure to do this correctly may nullify all subsequent results. I would also *highly* recommend familiarizing yourself with how to use Github to make data sharing easier amongst your group – there is a supplementary module on Canvas with an instructional video and a document with useful CLI commands for Github.
7) Use the accuracy metric or the AUC metric (if appropriate and specifically asked) to benchmark model performance in and out-of-sample for the classification task.
8) (Binary Classification Tools) If your output variable has multiple classes (ie Red, White, Blue), create a new column with a "dummied up *binary version"* of the same variable (ie: Red or Not). You may want to code the new column with outcomes as 0 or 1, and may want to choose the class that occurs in a way that will make the outcome variable the *least skewed* (ie: the most balanced). You may want to consider the role of stratified sampling here. For each of the sub-parts you may use whatever inputs you'd like, but the output variable predicted should be the *same* binary dummy variable.
    a) Estimate a logistic regression model to predict your binary outcome variable. Be sure to interpret model diagnostic information.
    b) How does a "Probit" model compare to the "Logit" model above? Estimate one.
    c) Use the ROC curve and the AUC metric to benchmark the in and out-of-sample performance of the models in a) and b) and compare these to the accuracy metric for comparing those models' performance.
9) (Multi-class Prediction Tools) For this task, you can predict either a binary output class or a multi-class output variable (more than two classes). However, if you choose binary classification (same problem as in a) and b) above), you should benchmark all of the remaining models (with the exception of the SVM) against the models in a) and b) by reporting *both* accuracy and AUC metrics if appropriate. Otherwise, use model accuracy to benchmark performance in and out-of-sample if you engage in a multi-class (more than two) prediction task.
    a) Estimate a support vector machine - you can consider the same or different inputs relative to the models above. Report the in-sample and out-of-sample performance to benchmark the model relative to those above (and below). Are you able to improve predictive performance by *tuning* your model?
    b) Estimate a classification tree - you can consider the same or different inputs relative to the models above. If you are able to produce a reasonably sized tree for a person to look at, display the decision tree. Report the in-sample and out-of-sample performance to benchmark the model relative to those above. Are you able to improve predictive performance by *tuning* your model?

c) Estimate a tree-based ensemble model - you can consider the same or different inputs relative the models above. Report the in-sample and out-of-sample performance to benchmark the model relative to those above. Are you able to improve predictive performance by *tuning* your model?
d) Create a table summarizing the in-sample and out-of-sample estimated performance for each of the models above and from 8). Use the validation partition to choose your best model based on the estimated out-of-sample performance on the validation set. Use the uncontaminated data partition to report the *uncontaminated* out-of-sample error on the final model chosen from the validation process.

TASK 3: THE REPORT

1) The report should contain R code output as well as written narrative to describe and document the steps taken in your analysis from start (cleaning / preprocessing) to finish (validation / model selection) along with any useful visualizations including (but not limited to) charts, graphs, and tables.
2) Here is a *suggestion* for how to structure your report (you can, of course, deviate from this):
    a) The first section could include an executive summary walking through the data collection process, describing the variables of interest, and discussing the overall structure of the dataset(s).
    b) The second section could include a carefully documented set of instructions of steps taken in preprocess and cleaning the data, as well as any diagrams included (ie: pretty ggplot2 pictures) in the exploratory analysis of the data (this section may be quite large depending on the "rawness" of your data.
    c) The third section could include the model proposals and summary of results for the regression tasks including diagnostics and relevant benchmark statistics and validation results (ie: goodness of fit, statistical significance, confidence intervals of estimated model parameters, heteroskedasticity tests, normality tests, VIF, etc.).
    d) The fourth section should include the model proposals for the classification tasks from as well as the model diagnostics (ie: goodness of fit, statistical significance, confidence intervals of estimated model parameters, etc.) and relevant benchmark statistics and validation results
    e) The final section should wrap up your results with a conclusion and discussion of your results and lessons learned along the way. Reflect on the motivating question(s) and the extent to which you are able to answer them (if at all). In which cases did your predictive models work well? In which cases did they not? Why?

TASK 4: THE PRESENTATION

1) Your group will summarize your findings in the report with a visually appealing presentation that should not exceed 20 minutes (I may interject and ask questions during the presentation).

2) The goal of the presentation should be to highlight features of your project (cool diagrams, interesting code chunks, tables, etc.) and not regurgitate the contents of the report in its entirety.
3) Presentations will take place during the final days of class on 5/14/25, 5/16/25, and 5/21/25. *All groups should be prepared to present on the first day (5/14/25).* You should plan on attending *all days of presentations* even if you are not presenting that day.
4) You should make sure your presentation is *animated* to highlight features in tables and diagrams (you'll be using the same screen I use in class – laser pointers won't work so make sure you *animate!!!*) to make it easier for your audience to follow the salient points of your work.

A FEW GOOD DATA RESOURCES:

Some places to find datasets (there are *many many* more):

1) Kaggle (lots of luck here in the past): https://www.kaggle.com/datasets
2) Government Open Data: https://www.data.gov
3) Gapminder: https://www.gapminder.org/data/
4) Federal Reserve of St Louis: https://fred.stlouisfed.org/
5) Penn World Tables: https://cid.econ.ucdavis.edu/pwt.html
6) Yahoo Finance https://finance.yahoo.com
7) UC Irvine Machine Learning Repository: https://archive.ics.uci.edu/ml/datasets.php

In case you would like a longer, more comprehensive list of data sources, here is one: https://www.kdnuggets.com/2017/12/big-data-free-sources.html.

DELIVERABLES:

Your group will submit four items (submission details below):

1) The R code used in your analysis (can be multiple files).
2) A written report summarizing your process – including the data selection / cleaning / preprocessing steps - and findings that integrate the statistical output from R as well as any relevant diagrams, tables, and visualizations.
3) The TIDY data that you used to conduct your predictive analysis.
4) Your presentation slides (on the day of your presentation).

TASK 5: SUBMISSION

You should email your (first three) **deliverables** as an attachment to slevkoff@sandiego.edu. In the subject heading of the email, include *"PREDICTION PROJECT BUAN / ECON 381 SP25*". In the body of the email, be sure to list all group members by first and last name. The deadline to submit the project report is Sunday, 5/18/25, before midnight. You can email deliverable #4 after your presentation. You will be graded on how well your group achieves the goals outlined above and on the quality of the report / presentation.