Devin Yerasi
305 167 818

1. **Data Collection and Analysis**
    a. There are a few main problems with this method of data analysis and collection. The choice of using only tweets containing keywords/hashtags related to LA public health will not be representative of the population of patients who visit public clinics in the city. Not only will this dataset include those who have never been to an LA public clinic, but it will also not include the whole population of public clinic visitors. LA public clinic attendees may not be on twitter for personal or financial reasons, or they simply may not publicly tweet about it. There will also be room for error in choosing keyword/hashtags to filter tweets by. There also is a volunteer bias, as those with strong opinions about LA public health can be more likely to tweet more about it, leading to skewed data. Also, only examining the semantics of the tweet may twist the original message intended for the tweet. For example, a tweet ridiculing LA traffic with a small blurb about how LA public clinics are a positive of LA could be falsely analyzed by the semantic algorithm to be critical of LA public clinics. Only examining positive/neutral/negative sentiment in the overall tweet is likely to obscure meaning regarding different objects in the tweet.

2. **Experimental Design**(aiming to predict a student's final grade in the class)
    a. The more features I gather, the better.
        i. Student's school year
        ii. Class past grade distribution
        iii. Past classes taken with what professor with what ending grade
        iv. The other classes the student is taking along with the course
        v. How interested the student is in the course material
        vi. Student attendance in lectures
        vii. Student attendance in classes
        viii. Student attendance in office hours
        ix. Whether the student turns on their webcam or not
        x. How often the student checks the course page/discussion per week
        xi. Key concepts of the course
        xii. Key concepts of other courses the student has taken
        xiii. How many other students in the course the student knows
        xiv. Grading scheme of this course and past courses
        xv. Has the student had the professor before
        xvi. And many other features involving the student's wellbeing and activity such as sleep data, stress levels, financial stability, physical health, organizations the student is involved in, etc
        xvii. Basically, I would have as many features as possible initially and remove some as I start to analyze the data
    b. How I would formulate labels:
        i. School year as an int 1-5, with 5 being a bucket for anything > 4 years
        ii. A list of tuples (grade, percent of class getting grade)
        iii. List of tuples containing (course id, professor name, grade, date taken)
        iv. Int of number of units being taken in the quarter
        v. Scale 1-10 indicating student's relative interest in the course material
        vi. Float of number of classes attended / total classes
        vii. Float of number of discussions attended / total discussions
        viii. Int of number of office hours attended in past courses

ix. Float of number of times camera is on / total number of times student attended
x. Int of average past weekly logins to ccle/piazza
xi. List of key words in description of course on syllabus
xii. List of tuples of (key words, course id) of past course
xiii. Int of number of friends in the course
xiv. 1-5 buckets of grading types with 5 leaning towards greater weight on tests and 1 meaning pure participation
xv. Boolean value T if student has had the course professor before

c. How I would source/formulate/gather the data:
    i. Obtain student transcript including current courses being taken
        1. Generate past courses, professors, and grades from this
        2. Use ccle to get syllabuses of past courses
            a. Hash through description of course in syllabus containing only words that are relatively unique(do not include and, the, etc..)
            b. Parse grading types into weight that tests have on final grade and bucket it 1-5 with 5 meaning grade is 100% determined by tests and 1 meaning grade is 0% based on tests
    ii. I would use bruinwalk or ask the professor for a grade distribution for past offerings of the course
    iii. Google form with questions about interest in course, friends in course and other personal questions
        1. Ask student to fill out after signing up for course
    iv. zoom/ccle/piazza data on attendance and past weekly usage from previous quarters
        1. Collab with administration for this one
    v. Much of this information is personal and could dissuade a student from taking part in the study, if so, we may want to exclude data generated from a survey and focus on school generated data such as those from the transcript.

3. **Imputation**:
    a. Replace nan values with mean:
        i. Replacing nan values with the mean of the numerical feature will always preserve the mean of the feature. It can, however, add much more bias to the dataset. When the numerical feature data follows a normal distribution that is not skewed, replacing nan values with that feature mean is a decent way to go about imputation. As the data does not need to be sorted, this imputation can be done in O(n), making it advantageous for large data sets. If there is a large proportion of nan values, or the data is skewed and not normally distributed, using the mean is not a good option as it will not lead to a dataset that is representative. The mean is more prone to error if the dataset includes large outliers in either direction. This will underestimate the variance.
    b. Replace nan values with median
        i. This is a more computationally expensive imputation method. If the data is not already sorted, it can be done in O(nlogn) as we must sort the data first. As such, this is a better imputation method for already sorted data. Also, the data does not need to be normally distributed, and the median is more resilient than the mean in dealing with large outliers. Once again, this is meant for numerical features, and is inadvisable for categorical features. This will underestimate the variance.
    c. Replace nan values with mode
        i. When a feature is categorical or has a small clustered number of data points, the mode is a way to go about imputation. It is best if there are a small number of missing entries in the feature when using this, or the previous imputation methods, as this will underestimate the variance. It is not computationally expensive, O(n), to compute the mode and replace nan values with it. It is best in datasets where the mode frequency is significantly higher than the next few most frequent data values.
    d. Hot Deck Imputation

i. Hot Deck imputation involves replacing each missing value with a value from a similar data entry. How “similar” is defined and implemented is not standardized, and the most basic implementation(from what I could see online) of it involves just choosing the value from other entries at random. In this simple way, imputation can still be done in O(n), and this method helps maintain standard deviation by choosing the values at random. One can also make this method of imputation very computationally expensive by generating heuristic functions to find the most similar entry to replace the nan value in one feature by examining relationships between entries across all the other features. Using this method also has the advantage of remaining inside the existing set of values for a feature. However, it is highly likely that one could destroy/contradict existing relationships between features by substituting nan values with other existing values in the dataset.

4. **Utility of One-Hot-Encoding**:
    a. Heart Rate(beats/minute): I would not one-hot encode this feature, as it is already numerical, spans a large range, and has an implicit ordering in its definition that would be destroyed by OHE.
    b. A health category 1-5: I would not OHE this feature either, as there is an implicit ordering in its definition that should be preserved as it seems useful for many health problems.
    c. A list of fashion brands: Provided the list is not too long, I would definitely OHE this feature, as each fashion brand is a separate entity and has no definite implicit ordering to the brands. If it was a large list of brands, I would consider a single int feature based on some metric of the brands such as yearly revenue if it is relevant to the question I am analyzing, as one-hot-encoding a large list will blow up the number of features and make my model much more computationally expensive and weight the total brand feature much more than it should, possibly leading to a worse model.
    d. State(part of address): As there are only 50 states, I would opt to one-hot-encode them as there is, most likely, no ordering among them that I should preserve. So one-hot-encoding will generate a good numerical/boolean representation of which state the row entry is in without creating some random ordering of them that will feed in false information into my model.
    e. Interaction term of day and direction facing when trying to predict sun-exposure to a plot of land: Assuming time of day is at most grouped by hour and not minute, I would one-hot-encode this interaction term. There will be a relatively small, discrete set of possible values which will each correspond to a unique sun/direction setting. As the location of each plot of land is variable, there is no inherent ordering relating sun exposure to the interaction term values. Each of the values of the interaction term should have their own coefficient when predicting sun-exposure with linear regression for a more accurate model.

5. **True or False, Simple Explanations:**
    a. FALSE: a p value is the estimated probability of finding the observed/given results given that the null hypothesis is true. As such, a small p-value($p<0.005$) is grounds to reject the null hypothesis.
    b. FALSE: One of the main issues that can arise from augmenting your data with additional features is overfitting. By adding additional features based on the same sample data, you put more weight on the sample data which may not be a good representation of the population data, leading to a model that overfits.
    c. FALSE: Sometimes there is no good available dataset to address a data science problem, so one must go about generating the data themselves.
    d. FALSE: One-hot-encoding should not be used on categorical features with a large number of distinct values, as it will blow up the feature dimensions. Each distinct value will end up being its own feature.
    e. TRUE: the use of historical datasets comes with the understanding that there are historical biases in the data. Using this data to make decisions about the future allows the historical biases to influence future decisions. For example, if you used data from the 1950s on best hiring practices for law firms to create a model screening modern workers applying to a law firm, you might see the model favor white males coming from the east coast, most likely, due to the social trends in the 1950s and not due to the applicants skills.

6. **Basic Probability: let x1 = first draw, x2 = second**
    a. P(X=1) = P(x1 = 1, x2 = 0) + P(x1 = 0, x2 = 1) = 2/6*4/5 + 4/6*2/5 = 8/15
    b. P(X=0, Y=1) = P(x1 = w, x2 = !(w || r)) + P(x1 = !(w || r), x2 = w) = 2*P(x1 = w, x2 = b) = 2 *(3/6)*(1/5) = 1/5