Set threshold

Dep: above 100 or below 100

4models: knn, scn, bagged trees, decision tree, naïve tree (fast) , boosting model (moderate)

False Positive is better than false Negative

My project business value

- predicting if airbnb nightly prices will fall into two categories: above or below the listed nightly price- feature examples: room type, cleaning fee, cancellation policy, review score rating, minimum nights, host is super host, host response rate, host response rate, host since, number of reviews... etc- 50600 observations

guests\_included

bed type

beds

accomendates

bathrooms

property\_type

1. you'll see at the bottom of the sheet there's another spreadsheet (yay) that contains a project planning doc. you don't have to, but it would make your project a lot more clear when you're presenting it to us to see if it's viable!

2. remember that this is a CLASSIFICATION project (vs a REGRESSION project) so what you're trying to predict should NOT be a continuous variable. (multiple categories is ok!!!)

3. models we can use (you should test AT LEAST 4):

- logistic regression

- k nearest neighbors

- trees (decision tree, bagged trees, random forest)

- boosting models (adaboost, gradient boost, xgboost)

- support vector machines

- naive bayes

**Module 5 Final Project**

**Introduction**

In this lesson, we'll review all the guidelines and specifications for the final project for Module 5.

**Objectives**

* Understand all required aspects of the Final Project for Module 5
* Understand all required deliverables
* Understand what constitutes a successful project

**Final Project Summary**

Congratulations! You've made it through another *intense* module, and now you're ready to show off your newfound Machine Learning skills!

[](https://raw.githubusercontent.com/learn-co-curriculum/dsc-mod-5-project/master/smart.gif)

All that remains for Module 5 is to complete the final project!

**The Project**

For this project, you're going to select a dataset of your choosing and create a classification model. You'll start by identifying a problem you can solve with classification, and then identify a dataset. You'll then use everything you've learned about Data Science and Machine Learning thus far to source a dataset, preprocess and explore it, and then build and interpret a classification model that answers your chosen question.

**Selecting a Data Set**

We encourage you to be very thoughtful when identifying your problem and selecting your data set--an overscoped project goal or a poor data set can quickly bring an otherwise promising project to a grinding halt.

To help you select an appropriate data set for this project, we've set some guidelines:

1. Your dataset should work for classification. The classification task can be either binary or multiclass, as long as it's a classification model.
2. Your dataset needs to be of sufficient complexity. Try to avoid picking an overly simple dataset. Try to avoid extremely small datasets, as well as the most common datasets like titanic, iris, MNIST, etc. We want to see all the steps of the Data Science Process in this project--it's okay if the dataset is mostly clean, but we expect to see some preprocessing and exploration. See the following section, ***Data Set Constraints***, for more information on this.
3. On the other end of the spectrum, don't pick a problem that's too complex, either. Stick to problems that you have a clear idea of how you can use machine learning to solve it. For now, we recommend you stay away from overly complex problems in the domains of Natural Language Processing or Computer Vision--although those domains make use of Supervised Learning, they come with a lot of other special requirements and techniques that you don't know yet (but you'll learn soon!). If you're chosen problem feels like you've overscoped, then it probably is. If you aren't sure if your problem scope is appropriate, double check with your instructor!
4. ***Serious Bonus Points*** if some or all of the data is data you have to source yourself through web scraping or interacting with a 3rd party API! Having projects that show off your ability to source data effectively make you look that much more impressive when showing your work off to potential employers!

**Data Set Constraints**

When selecting a data set, be sure to take into consideration the following constraints:

1. Your data set can't be one we've already worked with in any labs.
2. Your data set should contain a minimum of 1000 rows.
3. Your data set should contain a minimum of 10 predictor columns, before any one-hot encoding is performed.
4. Your instructor must provide final approval on your data set.

**Problem First, or Data First?**

There are two ways that you can about getting started: ***Problem-First*** or ***Data-First***.

***Problem-First***: Start with a problem that you want to solve with classification, and then try to find the data you need to solve it. If you can't find any data to solve your problem, then you should pick another problem.

***Data-First***: Take a look at some of the most popular internet repositories of cool data sets we've listed below. If you find a data set that's particularly interesting for you, then it's totally okay to build your problem around that data set.

There are plenty of amazing places that you can get your data from. We recommend you start looking at data sets in some of these resources first:

* [UCI Machine Learning Datasets Repository](https://archive.ics.uci.edu/ml/datasets.html)
* [Kaggle Datasets](https://www.kaggle.com/datasets)
* [Awesome Datasets Repo on Github](https://github.com/awesomedata/awesome-public-datasets)
* [New York City Open Data Portal](https://opendata.cityofnewyork.us/)
* [Inside AirBNB](http://insideairbnb.com/)

**The Deliverables**

Your completed should contain the following deliverables:

1. A ***Jupyter Notebook*** containing any code you've written for this project.
2. An ***"Executive Summary" PowerPoint Presentation*** that gives a brief overview of your problem, dataset, process for answering your questions and your results.

**Jupyter Notebook Must-Haves**

For this project, your jupyter notebook should meet the following specifications:

***Organization/Code Cleanliness***

* The notebook should be well organized, easy to follow, and code is commented where appropriate.
  + Level Up: The notebook contains well-formatted, professional looking markdown cells explaining any substantial code. All functions have docstrings that act as professional-quality documentation.
* The notebook is written to technical audiences with a way to both understand your approach and reproduce your results. The target audience for this deliverable is other data scientists looking to validate your findings.

***Process, Methodology, and Findings***

* Your notebook should contain a clear record of your process and methodology for exploring and preprocessing your data, building and tuning a model, and interpreting your results.