CS188 Final Project

Due March 11, 2021 at 10am PST via Gradescope and Kaggle

Introduction:

As one of California's fastest growing FreightTech companies, NEXT is on a mission to make freight painless. Through complete first-to-last mile solutions, we're defining how FreightTech can transform the $800B trucking and shipping industry. From our digital freight marketplace to smart load-matching, our technology provides shippers with access to limitless capacity and full transparency, while empowering drivers to work the way they want, when they want.

As we build the most trusted brand for our customers, we are creating the best place to work for our growing team of top talent from the tech and logistics industries. We have been recognized as one of Built in LA's Best Small Companies to Work For and 50 Startups to Watch.

Founded in 2015, NEXT Trucking is backed by Brookfield Ventures, one of the world's largest infrastructure investors, and Sequoia Capital, one of the most recognizable venture capital firms in the world.

Challenge:

NEXT Trucking is interested in better understanding the drivers in its network. The ultimate goal is to both build a predictive model capable of determining overall network capacity to handle incoming shipping requests as well as more effectively recruit high-performing drivers to enhance their network. As a first step, they are interested in developing a classification model to better determine whether or not a driver will be a high-performing one.

You and your team (yes you will be allowed to work in groups!), will serve as consultants to NEXT Trucking. You will be asked to develop a predictive model for this task and report out your findings to them.

This project will include both a structured component, where much like Projects 1 and 2, you will be given a specific set of instructions to complete. There will also be an unstructured contest for you to complete as well where you will be competing against your classmates to achieve the best model.

Project Overview:

This project will be comprised of several discrete components. Full credit for Project 3 will require your completion of all 3 components. Specifically you will be asked to produce or submit to the following:

- **Report:** A report documenting your work on the project and your findings
- **Coding Project:** Follow the steps detailed below on a Jupyter Notebook
- **Kaggle Contest** Submission

Final Deliverables:

- PDF output of Jupyter Notebook (submitted via Gradescope)
- PDF of Final Report (Submitted via Gradescope)
- Kaggle Competition entry submission

Timeline:

- Project will be released on Feb 18th
- Project will be due before the last class of the quarter (Thursday, Week 10) on **March 11th before 10am PST**

Collaboration Policy:

- Project work can be completed individually or as a group effort
- Groups of **up to four** members will be allowed
    - There will be no grading scale for group work (i.e., no difficulty adjustment for groups) so group work is encouraged
- **All Group Members must submit their work individually** to gradescope to ensure credit for their work.

Contest:

Once you have effectively trained your model, as a next step you will be asked to participate in a contest among your peers, hosted on **Kaggle**!

Unbeknownst to you, we have withheld 1000 instances from the dataset, the labels (and the two features needed to develop them) have been removed. You will use your model to generate a series of predicted labels. Your outputs will then be compared against the real labels and an accuracy score generated.

This score will be compared against your peers.

Bonus points on the project will be awarded to high-performing teams. The top 3 group submissions will also be offered the opportunity to present their findings to the NEXT Trucking leadership team.

Project Requirements:

Specific Coding Requirements:

1. **Generate labels** - Where drivers in the 75th percentile of 'loads' and the 75th percentile of 'most_recent_load_date' are assigned a label of 1 (indicating a high performing driver) with all others being assigned a 0 - (NOTE: your labels will likely be unbalanced. You will need to determine an approach to balancing your labels).
2. **Drop 'load' and 'most_recent_load_date' from your data frame -** Since those fields are being directly used to label your data please remove them from your training and testing cohorts.
3. **Run some basic statistics on your variables including correlations with labels and report findings** - Particularly once you employ PCA and Neural Nets and other 'black box' methods, the descriptive power of any of your features will effectively disappear. Still you want to report out meaningful correlations to NEXT Trucking to help them flag key indicators they can employ (this step will also be helpful for you in flagging potential co-linearities).
4. **Create a data feature extraction plan and implement a pipeline to execute it -** Determine and execute a plan to process your data for modeling and then implement a pipeline to execute it. Specifically:
    - **a.** Determine which fields to retain and which to drop.
    - **b.** For those you retain, determine a categorization strategy
    - c. Determine an imputation strategy (you should choose more than one imputation method depending on the specifics of your data
    - d. Augment at least one feature, ideally a feature cross, or non-linear transition
    - e. Determine a strategy for scaling features
    - **f. Implement a single pipeline to execute this transformation**
    - **g. Document your data strategy in your report.** Provide an explanation or justification for why you chose the data you did, and also detail any experiments you ran and the results
5. **Implement a basic Linear Regression** - With your newly pipelined data find and interpret important features (e.g. using regression and associated p-values). If there are any collinearities be careful when incorporating them into the regression.

6. **Implement Principle Component Analysis (PCA)** - Since your resulting dataframe is likely to be high-dimensionality, employ PCA to reduce the complexity of your dataframe
7. **Employ an ensemble method to your classification exercise -** Leveraging bagging or equivalent ensemble learning method to generate an optimized classification model
8. **Develop a Neural Net classifier** - Modify parameters to optimize outcomes. Report your customized parameter settings in the report.
9. **Cross-Validate your training results -** Employ K-Fold Cross-validation to your training regimen for both ensemble and NN classifers. (Optional: employ a stratifiedshufflesplit as well to ensure equitable distribution along a key parameter)
10. **Experiment with your own custom models and report out your highest performing model. Submit the model to the class-wide contest. -** For this part of the project you have free range to employ any of the tools you've learned in class, along with any additional tools or techniques you research independently.

Report Requirements:

Each team will be expected to submit a report accompanying their project. There is no specific length or formatting requirement, however this report will be shared with NEXT, and therefore is expected to be professionally produced. Points will be deducted for incomplete or unprofessional reports.

The report will be expected to contain the following sections:

1. **Executive Summary:** Single-page highlevel summation of the work done and key findings
2. **Background/Introduction:** Use the accompanying information provided by NEXT, along with your own industry research, to better explain the domain challenges
3. **Methodology:** Incorporate requirements 1, 5G and 10 from the coding requirements into a general description of the work that you have done on this project
4. **Results:** Report out your results from coding requirements 3, 8, 9.
5. **Discussion:** Provide context to the results you've obtained. Additionally, provide a set of recommendations to NEXT for how to leverage your findings along with next steps for analytic work
6. **Conclusion:** concisely summarize the work done on the project

Contest Submission Requirements:

Once you have trained your own model, as a next step you will be asked to participate in a contest among your peers, hosted on Kaggle.

Go to: https://www.kaggle.com/c/uclacs188/overview , register for a kaggle account and join this contest.

Under the data tab access **score.csv,** a file with 1000 additional instances from the dataset, the labels (and the two features needed to develop them) have been removed from.

Upload and pipeline it and plug it into your model in order to generate the predicted labels.

Output these predictions into a CSV file, using the format described on the contest page, and submit them to the contest. Your outputs will be compared against the real labels and an accuracy score generated.

This score will be compared against your peers.

Bonus points on the project will be awarded to high-performing teams. The top 3 group submissions will also be offered the opportunity to present their findings to the NEXT Trucking leadership team.

Metadata:

TABLE: driver_metrics_all.csv

- Row count: 8,414
- Total number of drivers signed up: 5313
- Total number of drivers approved: 363
- Date Range: 1/26/2015-2/17/2021

fields	description
dt	short for date
weekday	day of the week (Monday for example)
year	year value parsed from field dt
id_driver	driver ID
id_carrier_number	carrier number. For carrier type equals to fleet, one carrier can have multiple drivers; while for carrier type equals to Owner Operator, one carrier has one and only one id_driver.
dim_carrier_type	two types of carriers, one is Fleet and the other is Owner Operator. For Fleet, it can be a small company with multiple truckers but for Owner Operator, it usually only has one trucker.
dim_carrier_compa	carrier company name

ny_name	
home_base_city	the home base city the driver claimed
home_base_state	the home base state the driver claimed
carrier_trucks	type of the trucks. Please refer to the "Company Intro.pdf" if you'd like to learn more about truck types
num_trucks	# of trucks associated with this carrier
interested_in_drayage	Drayage - is a logistics term that involves shipping goods a short distance via ground freight, for example from port terminal to customer warehouse. This field becomes true if a carrier reported himself or her interested in providing drayage services.
port_qualified	This field becomes true if a carrier reported him or herself as port qualified so that we can assign them jobs to enter a port terminal.
signup_source	can be mobile or other as the value
ts_signup	Signup timestamp
ts_first_approved	The timestamp if a carrier was first approved. If null that means Next Trucking has not approved this driver yet.
days_signup_to_approval	If a carrier was approved, this field will be a non-null value which calculates the date from first signup till approval date
driver_with_twic	if a driver has TWIC insurance then this field will be true
dim_preferred_lanes	the driver can specify which lanes (preferred routes) he'd like to take
first_load_date	this field captures the date the driver serviced his first load
most_recent_load_date	this field captures the date the driver serviced his most recent load
load_day	the date of the loads serviced
loads	# of the loads a driver serviced
marketplace_loads_otr	An OTR (over-the-road) type of the loads covered by driver from marketplace (our app)
marketplace_loads_atlas	An drayage (ATLAS is our in-house solution that hosts drayage jobs) type of the loads covered by driver from marketplace (our app)

marketplace_loads	the sum of marketplace_otr and marketplace_atlas
brokerage_loads_otr	An OTR (over-the-road) type of the loads covered by a driver assigned by brokers which is a more traditional way and we have less control on.
brokerage_loads_atlas	An drayage (ATLAS is our in-house solution that hosts drayage jobs) type of the loads covered by a driver assigned by brokers which is a more traditional way and we have less control on.
brokerage_loads	the sum of brokerage_loads_otr and brokerage_loads_atlas
total_loads	the sum of brokerage_loads and marketplace_loads