Digital Object Identifier

2023 UCD Summer Research Practicum: Individual Report

MOHAMMED CHOUDHARY.
Group 15, Data Lead

I. INTRODUCTION

Our project focuses on the development of an application that utilizes a model predicting the busyness of various tourist attractions in Manhattan. This tool aims to serve both tourists and the general public of Manhattan by providing real-time data on the crowd levels at different venues. The objective is to facilitate better planning for tourists and locals alike, allowing them to visit attractions during less busy times.

However, there are certain limitations to the application. The busyness model primarily relies on taxi density to estimate crowd levels. Furthermore, the hardware constraints of our current deployment restrict us from utilizing our most advanced and accurate models. Beyond its immediate utility, the project also serves as a valuable learning experience for the group. It offers insight into the process of building a real-world application from start to finish, the nuances of team coordination, and the refinement of fundamental computer science skills like programming, pseudocode formulation, and working with various external applications, libraries, and tools.

II. GROUP/TEAM STRUCTURE

The project group is categorized into three main teams: the Data team, the Front-end team, and the Back-end team.

The Data team is spearheaded by two roles: the Data Lead and the Coordinator Lead. While the Data Lead is chiefly accountable for data analysis and machine learning model delivery, the Coordinator Lead synchronizes the efforts of all three teams, schedules team meetings, and arranges interactions with supervisors and support staff.

The Front-end team is led by the Front-end Lead, who is responsible for the website's design across both desktop and mobile platforms, including writing the associated CSS, HTML, and other relevant code. Concurrently, the Customer Lead handles branding tasks like logo design, naming the application, creating promotional videos, and refining the user interface.

Lastly, the Back-end team comprises two roles: the Maintenance Lead and the Back-End Code Lead. While the former ensures code compatibility and quality across teams, the latter focuses on the web app framework, back-end code, and database design.

A. ABBREVIATIONS AND ACRONYMS

This report employs two abbreviations for clarity and conciseness. The first, ".pkl", represents a Python file format used for both serializing and saving objects with the pickle module[1]. The second, "PMML", is an acronym for Predictive Model Markup Language, an XML-based language designed to define and share statistical and data mining models[2].

III. ROLE AND CONTRIBUTIONS

I served as the Data Lead for Group 15. My contribution encompassed close to 100% of the data team's total code[3] including data cleaning, data analysis, and machine learning model production. This included all 12 training runs, all preparation and processing steps for the Yellow Taxi Trip data[4], and the merging of Yellow Taxi Trip data with the OpenWeather historical data[5] for select training runs.

The data team made an informed choice to utilize the 2018 and 2019 versions of the Yellow Taxi Trip data. Our rationale was that the more recent datasets from 2020 and 2021 might have faced distortions due to the pandemic[6]. I started with two years of data to explore if adding an additional year would enhance the model's accuracy. However, the model's accuracy remained unchanged with the addition of more data. Subsequently, I produced a data quality report to evaluate the Yellow Taxi Trip data's quality. This helped in guiding decisions about feature selection and cleaning. Table 1 below showcases a segment of the results from the data quality report, highlighting the count of negative values, missing values, and outliers for the respective features.

Table 2 below lists some of the Pandas methods I utilized during the process.

For a comprehensive data quality assessment of the Yellow Taxi Trip data, please refer to the GitHub repository[3]. Subsequently, I developed the code to clean both the 2018 and 2019 Yellow Taxi Trip data[4]. The data was cleansed using the Pandas library[7].

Negative values, which were not consistent across any features, were rectified by using their absolute values. It was assumed that such negative entries were errors. Rows with missing values were either discarded or imputed[8], while outliers were either imputed or restricted to upper and lower

Feature	Outliers	Missing Values	Negative Values
VendorID	n/a	163435	0
tpep_pickup_datetime	n/a	0	n/a
tpep_dropoff_datetime	n/a	0	n/a
passenger_count	6822193	163435	0
trip_distance	8616584	0	6379
RatecodeID	n/a	163435	0
store_and_fwd_flag	n/a	163435	n/a
PULocationID	n/a	0	0
DOLocationID	n/a	0	0
payment_type	n/a	163435	0
fare_amount	7245537	0	138930
extra	115547	0	69385
mta_tax	526965	0	136433
tip_amount	3377959	0	763
tolls_amount	4674154	0	3101
improvement_surcharge	166209	0	138742
total_amount	7254872	0	138908
congestion_surcharge	599687	4585780	118992

TABLE 1. Data Analysis Summary[3]

Function	Description
len()	Counts items
isnull().sum()	Returns counts for null values
dropna()	Removes null values
unique()	Returns unique values
fillna()	Fills null values
df[feature] < 0).sum()	Counts negative values
df[feature].apply()	Converts negatives
tolist()	Converts to list
dtypes	Gets data types
to_datetime()	Converts to datetime
sort_values()	Sorts values
value_counts()	Counts unique values
drop()	Drops labels
var()	Computes variance
corr()	Computes correlation
del	Deletes objects
dt	Accesses datetime properties
head()	Gets first n rows
tail()	Gets last n rows
isfinite	Tests if finite
nunique()	Counts unique elements
isin()	Filters by list
map()	Maps values

TABLE 2. Pandas Methods Used[3]

quartiles[8]. The identification of outliers was accomplished through the Interquartile Range (IQR) method[9]. Table 2 delineates some of the functions employed during this cleaning phase[10].

To ensure the integrity and correctness of the cleansing process, I validated the decisions made by re-executing portions of the code from the data quality report. This was essential to ascertain that all data quality issues identified in the report had been addressed correctly, and that no inadvertent issues were introduced during the cleaning.

In the subsequent stages, I engineered new features primarily centered on the date-time data since the model would have faced challenges processing date-time features directly[11]. Two features, in particular, were introduced: ride duration and average speed. These were added to gauge their significance for the model and potentially enhance its accuracy. As it turned out, both features were of high importance, a fact corroborated by the feature importance documentation available in the GitHub repository[3].

- `day_of_month`: Day of the month of drop-off
- `day_of_week`: Day of the week of drop-off
- `hour`: Hour of the day of drop-off
- `month`: Month of drop-off
- `ride_duration`: Duration of the ride in minutes
- `average_speed`: Average speed of the trip in mph

During the design of the target feature, I encountered a significant challenge. Identical taxi volumes in different geographical areas might be indicative of varied levels of activity or busyness. To address this, I incorporated the Shape Area feature from the taxi zones dataset[12] and amalgamated it with the Yellow Taxi Trip data[4]. This strategic move enabled the design of a more nuanced target feature, one that normalized it based on the taxi zone size. The resultant target feature, *busyness*, is expressed as:

$$\text{busyness} = \frac{\text{number of taxis}}{\text{time} \times \text{area of zone}} \tag{1}$$

Predictions were formulated on an hourly basis, motivated by hardware constraints and the observation that significant variations were not prevalent from one hour to the next. Consequently, a finer granularity might not have introduced substantial value.

A particular challenge surfaced due to the disparity between the dataset's row count and the necessary rows for hourly predictions. My devised solution involved row aggregation, ensuring one row for each prediction. This configuration facilitated the machine learning models to establish a one-to-one relationship between a row and a target value. For categorical features, the mode of the row was employed, while continuous features used the row's mean. Subsequently, the *busyness* feature was categorized into five bins, ranging from 1 (least busy) to 5 (most busy). This categorization stemmed from a collective decision to depict busyness through a 1 to 5-star rating system, streamlining the user's interpretation. Categorical features underwent one-hot encoding[13], and when required, continuous features were scaled before consolidating them into the final Comma Separated Values (CSV) file designated for model training. Model evaluation harnessed standard metrics for classification issues such as accuracy, precision, recall, and the f1 score[14]. The model's test set encompassed the last 20 percent of the rows. It was pivotal to utilize a contiguous chunk of rows over a randomized selection to ensure the model's capability for sequential predictions, rather than isolated random predictions.

My exploration covered diverse algorithms suited for multi-class classification tasks with balanced classes. Their performance was benchmarked across separate training sessions to facilitate algorithm selection. The initial six training runs (specifically 1, 7, 8, 9, 10, and 11) informed the algorithmic decision-making. Several algorithms, encompassing Random Forest, Logistic Regression, K-Nearest Neighbour, XGB classifier, and Neural Networks[14][15], were trained. An attempt to employ the Support Vector Machine[14] was initiated but subsequently halted owing to its computational intensity. The constraints of limited hardware and memory became recurrent obstacles during the training phase. To counteract this, I leveraged cloud infrastructure through Google Colab. This not only equipped the data team with advanced computational resources and augmented RAM but also facilitated efficient file storage with a premium Google Drive account. There was a marked improvement in training speeds on the cloud platform, and file transfers within Google's ecosystem were expedited compared to my personal setup. Post-training, I determined metrics like accuracy, recall, f1 score, and precision[14] to evaluate, compare, and discern the nuances across various algorithms, datasets, and hyperparameters. The subsequent Tables 3 and 4 encapsulate the outcomes of these training runs.

Run	Purpose	Accuracy	Precision
1	Random Forest	0.8523	0.8522
2	Random Forest + Weather	0.8319	0.8313
3	Random Forest (2018+2019)	0.8491	0.8491
4	Random Forest (Feature changes)	0.8547	0.8546
5	Random Forest (Hyperparameters)	0.8537	0.8536
6	Random Forest (small)	0.7391	0.7380
7	Logistic Regression	0.7188	0.7178
8	K Nearest neighbour	0.8283	0.8302
9	Decision tree classifier	0.7879	0.7869
10	XGB classifier	0.8322	0.8314
11	Neural network	0.8850	0.8875
12	Random Forest (small 2)	0.7074	0.7097

TABLE 3. Summary of Training Runs with Accuracy and Precision Metrics[3]

Run	Purpose	F1	Recall
1	Random Forest	0.8519	0.8521
2	Random Forest + Weather	0.8308	0.8317
3	Random Forest (2018+2019)	0.8499	0.8524
4	Random Forest (Feature changes)	0.8544	0.8545
5	Random Forest (Hyperparameters)	0.8534	0.8536
6	Random Forest (small)	0.7337	0.7378
7	Logistic Regression	0.7080	0.7179
8	K Nearest neighbour	0.8282	0.8287
9	Decision tree classifier	0.7871	0.7877
10	XGB classifier	0.8312	0.8318
11	Neural network	0.8855	0.8851
12	Random Forest (small 2)	0.6984	0.7058

TABLE 4. Summary of Training Runs with F1 and Recall Metrics[3]

DEFINITIONS OF METRICS

1) **Accuracy**:
   - Definition: Proportion of all predictions that are correct[14].
   - Formula[14]:

$$\text{Accuracy} = \frac{\text{Number of Correct Predictions}}{\text{Total Number of Predictions}} \tag{2}$$

2) **Precision**:
   - Definition: Proportion of positive identifications that were correctly classified[14].
   - Formula[14]:

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}} \tag{3}$$

3) **Recall**:
   - Definition: Proportion of actual positives that were identified correctly[14].
   - Formula[14]:

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}} \tag{4}$$

4) **F1 Score**:
   - **Definition:** The F1 score is a measure of accuracy that considers a harmonized balance of recall and precision[14].
   - **Formula[14]:**

$$\text{F1 Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{5}$$

These metrics provide an in-depth understanding of the model's ability to predict "busyness" accurately. A high accuracy offers an overarching sense of the model's predictive capabilities[14]. Precision indicates the ratio of predictions for specific busyness levels that genuinely matched those levels of activity[14]. Recall illustrates the percentage of actual instances of a particular busyness level that the model correctly predicted[14]. The F1 score represents a harmonized measure of precision and recall, ensuring neither metric is disproportionately prioritized[14]. The neural network and the random forest models emerged as the top-performing ones. However, due to the back-end team's preference for a PMML file over a .pkl file and the greater compatibility of the random forest model with the PMML format, it was chosen for deployment. I incorporated the Yellow Taxi Trip data with historical weather information from Openweather[5] for New York City in training runs 2 and 12, aiming to gauge the impact of the weather data[3]. A challenge surfaced because the datasets operated on different time zones. I utilized the Pandas tz_convert method to adjust the weather data to New York's timezone. The model's performance was slightly diminished with the weather data, leading to its exclusion from the deployment model. Training run 3 assessed the influence of an enlarged Yellow Taxi Trip dataset[3]. The advantage of the additional data was negligible and didn't justify the extra hardware resources the enlarged model demanded. Training run 5 mirrored the initial Random

Forest Classifier training run but incorporated more decision trees[3]. The influence on accuracy and other metrics was minimal. Training run 4 explored a varied feature selection after excluding some low-importance features identified in the Github repository's feature importance document[3]. The impact on the metrics was slight. The ultimate deployment models were Training runs 6 and 12[3]. Due to the removal of certain features unavailable in real-time data, these models were more compact and showed marginally reduced accuracy. The application was designed for present and future use, not solely historical. In future endeavors, we aim to integrate more live data to enhance the model's forecasting capabilities. Training Run 12 encompassed weather data, while Run 6 omitted it[3]. Run 12's predictive power was slightly inferior to Run 6[3]. In conclusion, the models demonstrated robust performance on the test metrics and were promptly delivered to the back-end team[3].

I had various other responsibilities within the team framework. I played a pivotal role in the initial project selection. After devising a report detailing ten distinct project ideas, complete with their pros and cons, I convened a team meeting to determine the final choice. When conflicts arose in the Front-End team regarding the project's direction, I participated in meetings to foster resolution. Although the coordination lead arranged most meetings with the demonstrators, I organized additional sessions whenever the data team sought further guidance. I also contributed to the research and composition of Sections IV and V of the group paper. This task necessitated acquiring knowledge in research methodologies, report drafting, and presentation skills - tools I believe will be invaluable in my forthcoming endeavors. I am gratified with the outcomes in my key responsibility areas. The models were trained with commendable accuracy. The stipulations and hardware constraints from the back-end team were met, and the model functioned as designed, integrating flawlessly into the final application. An extensive amount of data concerning the model in diverse scenarios was amassed due to the numerous training runs conducted.

IV. LESSONS LEARNED

Personal Growth and Technical Advancements: This project has profoundly enriched my learning. Leading the data analysis, troubleshooting issues, and aligning with the needs of other teams, I've seen considerable growth in my technical expertise and leadership prowess.

Exploring Large Datasets: My position as the data lead introduced me to the intricacies of handling vast datasets, like the Yellow Taxi trip data. A pivotal lesson was recognizing the need for comprehensive analysis before venturing into constructing machine learning models.

Data Quality and Feature Engineering: Ensuring the integrity of the data was paramount. Addressing missing values, inconsistencies, outliers, and meticulously selecting and engineering features became the cornerstone for building an effective predictive model.

The Essence of Team Communication: The disagreements within the Front-End team underscored the significance of communication. In a diverse team like ours, fostering an environment conducive to open dialogue and collaborative decision-making is crucial.

Challenging Assumptions: This journey compelled me to reevaluate several presumptions about predictive models. Contrary to expectations, incorporating weather data reduced accuracy. Additionally, the minimal impact of expanding the Yellow Taxi Trip data taught me that sheer volume isn't synonymous with enhanced performance. Instead, the interplay of data relevance, quality, and quantity defines a model's success.

Flexibility in Decision Making: Real-world projects demand adaptability. This experience necessitated multiple shifts in my approach, be it in algorithm selection, feature preferences, or infrastructure considerations, ensuring the project's triumphant realization.

Understanding Scalability and Computing: The computational demands and memory constraints highlighted the imperative of scalability. The project accentuated the benefits of cloud computing, especially when maneuvering sizable datasets.

Final Thoughts: As the Data Lead for Group 15, I played an integral role in orchestrating a successful data-centric venture. This project, with its multifaceted challenges and opportunities, has fortified my foundation, paving the way for my journey as a future data scientist.

REFERENCES

[1] Python Software Foundation. "pickle — Python object serialization," in Python 3.11.4 documentation, Sep. 30, 2021. https://docs.python.org/3/library/pickle.html. Accessed on: Aug. 12, 2023.

[2] Data Mining Group. "Data Mining Group - DMG," in DMG, 2021. https://dmg.org/. Accessed on: Aug. 12, 2023.

[3] Q. Huang et al., "NewYork-Busyness-Web," GitHub repository, [Online]. Available: https://github.com/QinghanHuang/NewYork-Busyness-Web.git. [Accessed: Aug. 12, 2023].

[4] Taxi and Limousine Commission (TLC), "2019 Yellow Taxi Trip Data," City of New York, Updated: May 9, 2022, [Online]. Available: https://data.cityofnewyork.us/Transportation/2019-Yellow-Taxi-Trip-Data/2upf-qytp. [Accessed: Aug. 12, 2023].

[5] OpenWeatherMap, "OpenWeatherMap," OpenWeatherMap. Available: https://openweathermap.org/. Accessed: Aug. 13, 2023.

[6] Office of the New York State Comptroller. "New York City Tourism Industry Hit Hard by Pandemic, Visitor Spending Drops by 73 percent," in Office of the New York State Comptroller, Apr. 28, 2021. https://www.osc.state.ny.us/press/releases/2021/04/new-york-city-tourism-industry-hit-hard-pandemic-visitor-spending-drops-73. Accessed on: Aug. 12, 2023.

[7] S. Cass, “Top Programming Languages 2021,” in IEEE Spectrum, vol. 58, no. 9, pp. 28-29, Sep. 2021. https://spectrum.ieee.org/top-programming-languages-2021.

[8] R. J. A. Little and D. B. Rubin, Statistical Analysis with Missing Data, 3rd ed. Hoboken, NJ: Wiley, 2019. ISBN: 978-0-470-52679-8.

[9] J. W. Tukey, Exploratory Data Analysis, 1st ed. Reading, MA: Addison-Wesley, 1977.

[10] pandas development team. “pandas documentation,” in pandas 2.0.3 documentation, Jun. 28, 2023. https://pandas.pydata.org/docs/. Accessed on: Aug. 12, 2023.

[11] Majeed, “Improving Time Complexity and Accuracy of the Machine Learning Algorithms Through Selection of Highly Weighted Top k Features from Complex Datasets,” in Annals of Data Science, vol. 6, no. 3, pp. 599-621, Sep. 2019. https://doi.org/10.1007/s40745-019-00217-4.

[12] Taxi and Limousine Commission (TLC), "NYC Taxi Zones," City of New York, [Online]. Available: https://data.cityofnewyork.us/Transportation/NYC-Taxi-Zones/d3c5-ddgc. [Accessed: Aug. 12, 2023].

[13] Enago Academy. “One-Hot Encoding: A Simple Yet Effective Way to Handle Categorical Data,” in Enago Academy, Aug. 13, 2023. https://www.enago.com/academy/one-hot-encoding-a-simple-yet-effective-way-to-handle-categorical-data/. Accessed on: Aug. 12, 2023.

[14] M. Grandini, E. Bagli, and G. Visani, “Metrics for multi-class classification: an overview,” arXiv preprint arXiv:2008.05756, 2020. https://arxiv.org/pdf/2008.05756. Accessed on: Aug. 12, 2023.

[15] M. Belgiu and L. Drăguţ, “Random forest in remote sensing: A review of applications and future directions,” in ISPRS Journal of Photogrammetry and Remote Sensing, vol. 114, pp. 24-31, Apr. 2016. https://doi.org/10.1016/j.isprsjprs.2016.01.011.

[16] S. K. Singh, A. K. Singh, and S. K. Singh, “An efficient XGBoost–DNN-based classification model for network intrusion detection,” in Neural Computing and Applications, vol. 33, pp. 10055-10069, Jul. 2021. https://doi.org/10.1007/s00521-020-04708-x.