Main Github:
https://github.com/gav-ip/cse176-fall-proj

Everybody worked on Report 3 and the slides.

GAVIN:

Created *requirements.txt*

Part1

Completed logistic regression sections of part1 including hyperparameter tuning, creating confusion matrix, and *digitfilter.py* module.

Part2

Helped in completing XGBoost training over MNIST dataset. Implemented early stopping and k fold cross validation.
Created xgboost_summary.txt code block
Plotted XGBosst Error Rate: Train vs Val vs Test
Generated MNIST confusion matrix

Part3

Created code blocks to check for missing values
Generated feature visualizations
Helped implement cleaning filters for dataset as well as plotting obscure feature outliers
Helped transform target prediction dataset and implemented correctly displayed score in both RMSE seconds and raw RMSLE

Reports

Part 3 report - Helped write sections on data cleaning, feature visualization, intro, model description/ use, and reason for using scoring metric

RONAN:

Part1

Created *randforest.py* in part1 (now combined with logreg) [Includes graph + confusion matrix + code + model]

Part2

Wrote the pickle groundwork for part2 (honestly didn't do much in part 2)

Part3

Made feature importance graph/code for taxi regression
Retrained model with weather features from another dataset for reference
Created + Trained classification model
Pickled classification model (misread the TA's email; this was unnecessary, but it's in the GitHub, pretrained for future use and quick access)

Reports

Wrote the report for part2

CHRISTOPHER:

Part1

Optimised file runtime and parameters for higher accuracy
Pooled data and remapped in preparation for the report and feedback from TA

Part2

Trained the model + optimized to pickle for Part2 for both the pixel features and the LeNet feature mapping

Part3

Created a workaround for NaN's and other values for preprocessing data
Implemented using different coordinates, grid coordinates, and geographical transformations for taxi regression (Haversine, street distance, long/lat, etc)
Created a function for a sinusoidal reflection on the temporal region to scale time evenly with time periods (2400 -> 0000 is closer).
Added more features for training using algorithms for data augmentation
Optimized parameters, scoring methods, and algorithms to speed up training
Found and implemented ways to optimize and utilize all cores to speed up training and model speed

Reports

Wrote the report for Part1

Sources used:

https://www.kaggle.com/code/headsortails/nyc-taxi-eda-update-the-fast-the-curious/report#intermission---the-best-spurious-trips

- Referenced methods and cleaning in part3

https://stackoverflow.com/questions/4913349/haversine-formula-in-python-bearing-and-distance-between-two-gps-points

- Referenced for haversine formula and implmentation

https://www.kaggle.com/code/janiobachmann/credit-fraud-dealing-with-imbalanced-datasets

- Referenced methods for dealing with imbalance and stratification

https://www.kaggle.com/datasets/mlg-ulb/creditcardfraud

- Dataset used for classification on our model

https://www.kaggle.com/c/nyc-taxi-trip-duration/data

- Dataset used for regression on our model

https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.HistGradientBoostingClassifier.html

- Learning model itself w/ documentation

https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.HistGradientBoostingRegressor.html

- Regression learning model w/ docs

https://xgboost.readthedocs.io/en/stable/python/sklearn_estimator.html

- Docs in early stopping referenced in part2

3.4. Metrics and scoring: quantifying the quality of predictions — scikit-learn 1.8.0 documentation

- Reference for choosing a score function

RandomizedSearchCV — scikit-learn 1.8.0 documentation

- Referenced usage of RandomizedSearchCV