NYC Taxi Trip Duration

Kaggle Competition Case Study

Allan Dieguez

Introduction

Case Objective

Predict the total duration, in seconds, of a taxi trip in NYC.

Available Measurements:

- ❏ Vendor ID
- ❏ Passenger Count
- ❏ Store & Forward Flag (system)
- ❏ Pickup Date/Time
- ❏ Pickup Point (latitude / longitude)
- ❏ Dropoff Point (latitude / longitude)

Development Tools

Data Exploration

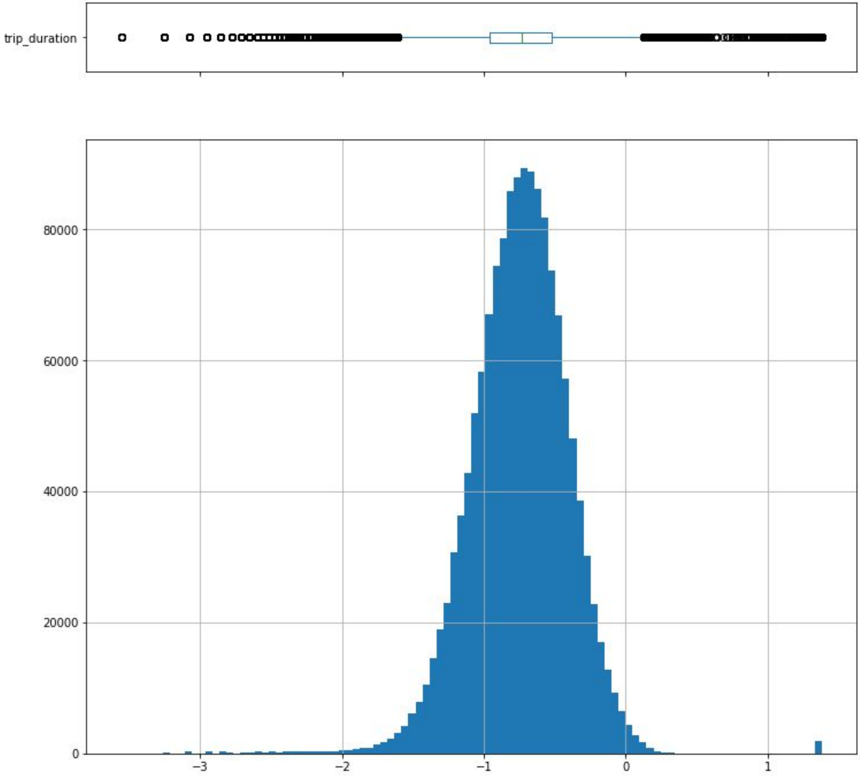
Trip Duration (Dependent Feature)

Observations:

- 4 trips with duration > 1 day (22 to 40)
- ~99% trips under 1 hour
- ~0.8% trips between 1 and 10 hours
- ~0.1% trips between 10 and 24 hours

Cuts:

- Trip Duration < 3 minutes
- Trip Duration > 15 hours
- Keep: 97.96 % of the training data



Trip Duration histogram: X-axis in ***log10 scale***

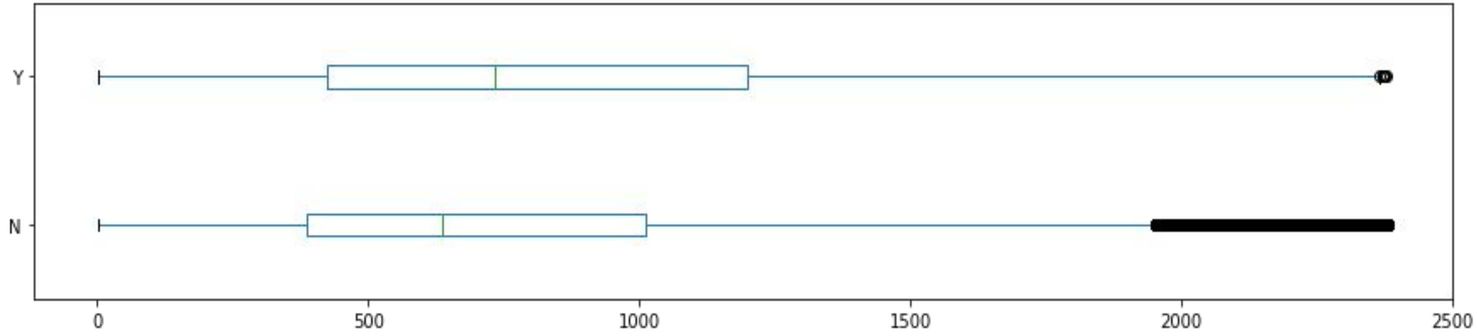
Vendor ID & “Store and Forward” Flag

Vendor IDs:

- 2 Vendors / Taxi Companies
- Shares:
  - Vendor 1: 46.5 %
  - Vendor 2: 53.5 %
- **No cuts performed**

“Store & Fwd” Flag:

- Only available for Vendor 1
- Share: 1.18 % of all Vendor 1 trips
- Trip Duration distribution slightly higher for “Y”
- **No cuts performed**



Trip Duration (in seconds) distribution by**Store & Fwd** flag values **Y** and **N** (Vendor 1).

Passenger Count

Observations:

- 60 trips with zero passengers
  - 42 lasted less than 1 minute
  - 17 lasted less than 1 hour and more than 1 minute
  - 1 lasted more than 23 hours
  - Vendor IDs:
    - Vendor 1: 31 trips
    - Vendor 2: 29 trips
- 126,425 trips with more than 4 passengers
  - 4 of them had more than 6 people

Cuts:

- zero passenger trips
  - Too noisy
  - Not that representative

Feature Engineering:

- Feature is Categorical
- Classes 1 to 9 (ignore 0)

Pickup Date/Time

Observations:

- No inconsistency on Trip Duration Vs Pickup to Dropoff timestamps
- All trips happened between January and June of 2016
- Trip counts:
  - High on the middle of the week
  - Low from Saturday to Monday

Cuts:

- **No cuts performed**

Features Engineering:

- Hour
- Week Day
- Month

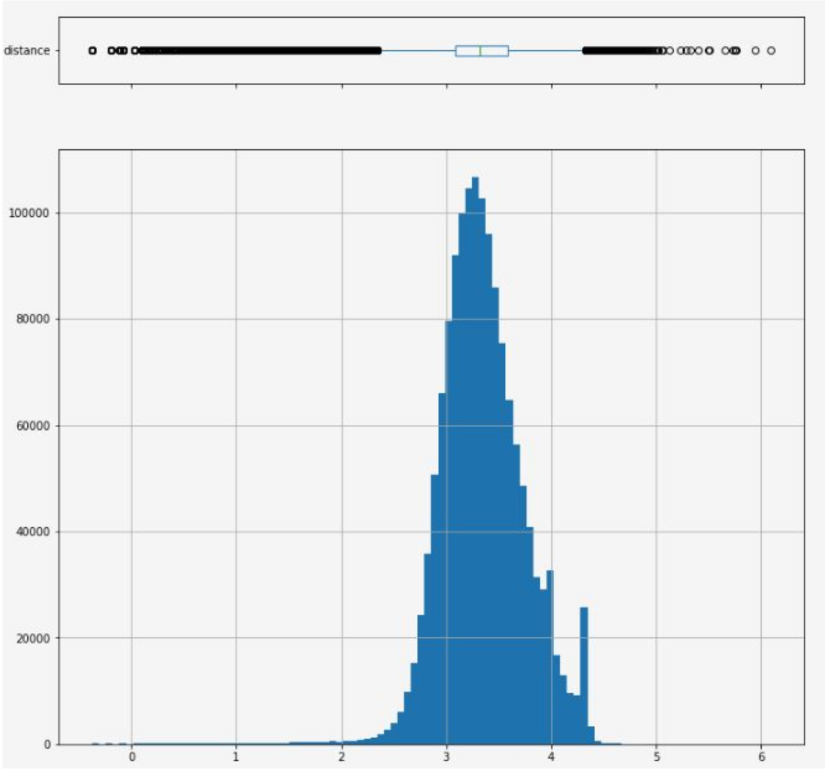
Pickup & Dropoff Distance

Observations:

- Method: Vincenty Distance
- Strange Trips outside NYC:
  - Canada
  - San Jose
  - Atlantic Ocean

Cuts:

- Trip Distance < 100 m
- Trip Distance > 100 km
- Latitude between 40 and 42
- Longitude between -74.5 and -73.5
- Keep: 99 % of the training data



`Trip Distance histogram: X-axis in` ***`log10 scale`***

Feature Engineering

Pickup & Dropoff Points

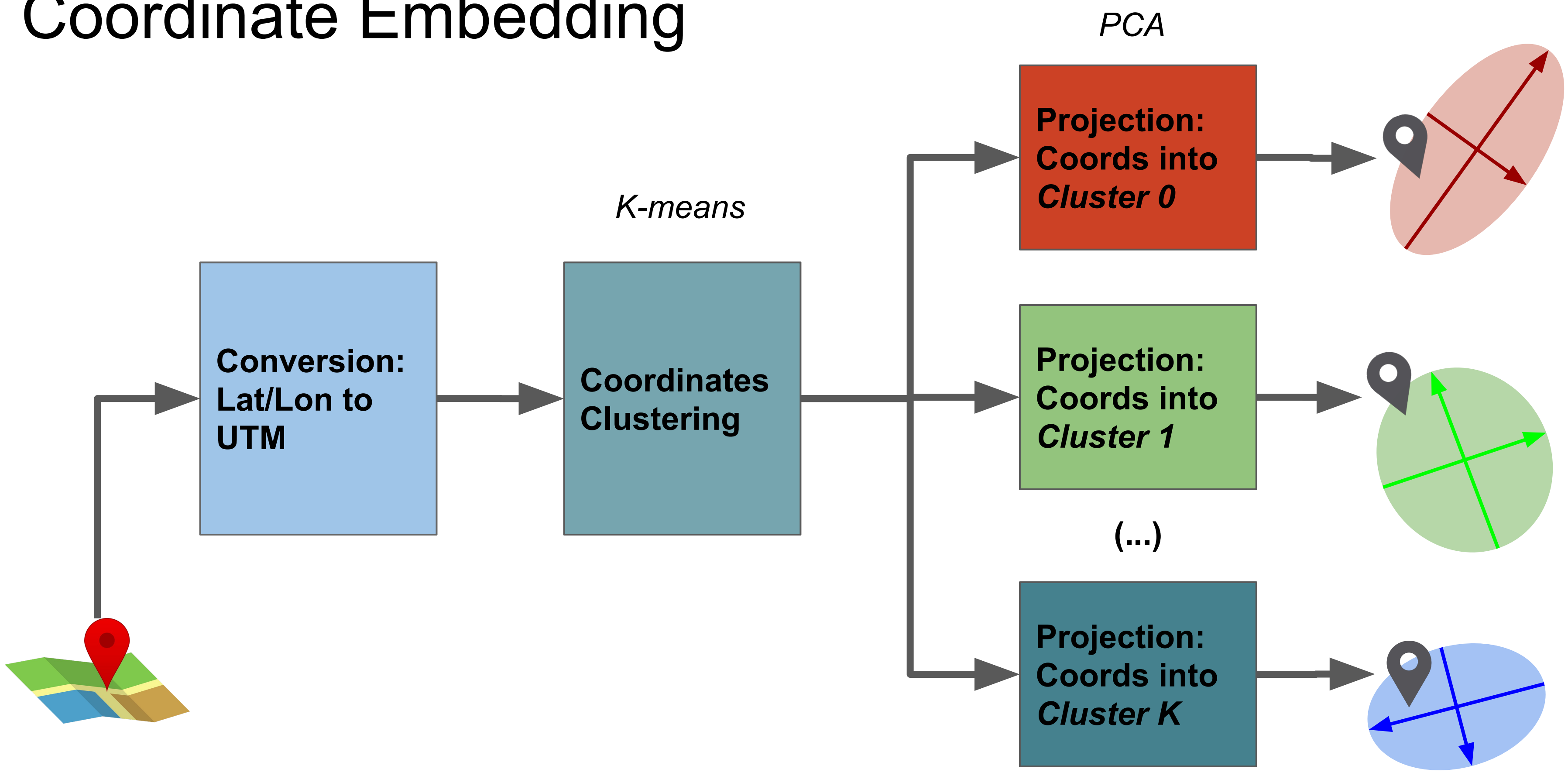
Observations:

- Latitude and Longitude are **lousy features**: they are unique globally
- Feature engineering is required to leverage the potential of the feature

Feature Engineering:

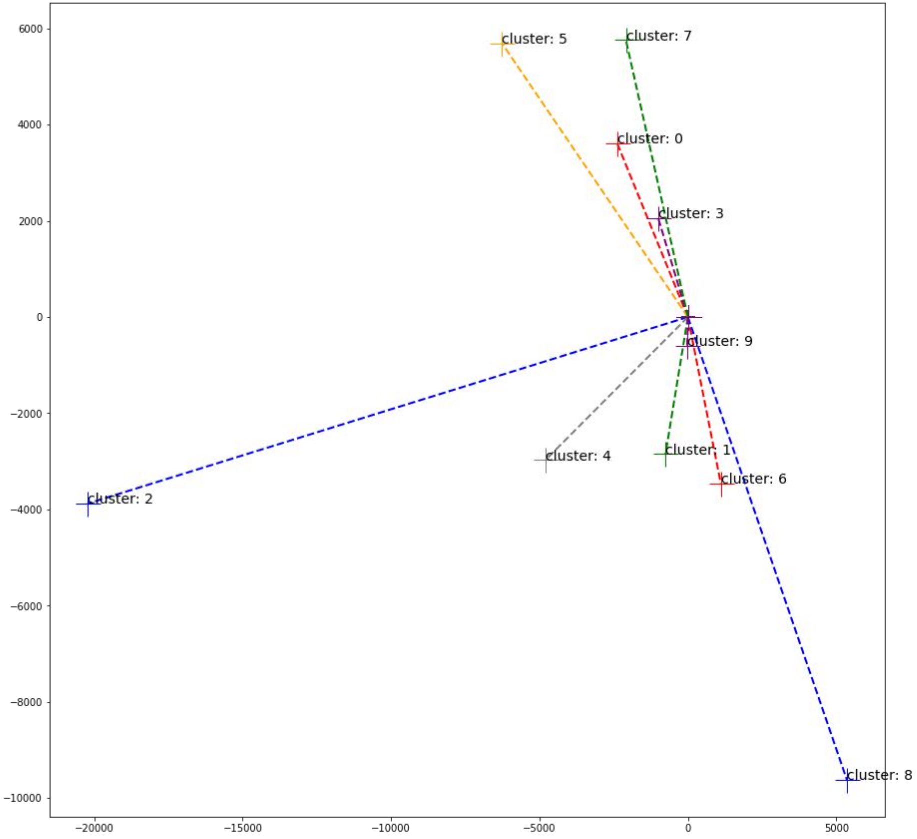
- Direct distance between Pickup and Dropoff points
- Local Projection, e.g. UTM (easting, northing)
- Distance from known Control Points in the city:
  - Domain knowledge of the city
    - Airports, Ferry, Railroads, Bus Stations
    - Landmarks, Tourism
    - Economic Centers
  - Automatic discovery: Clusters!!!

Coordinate Embedding

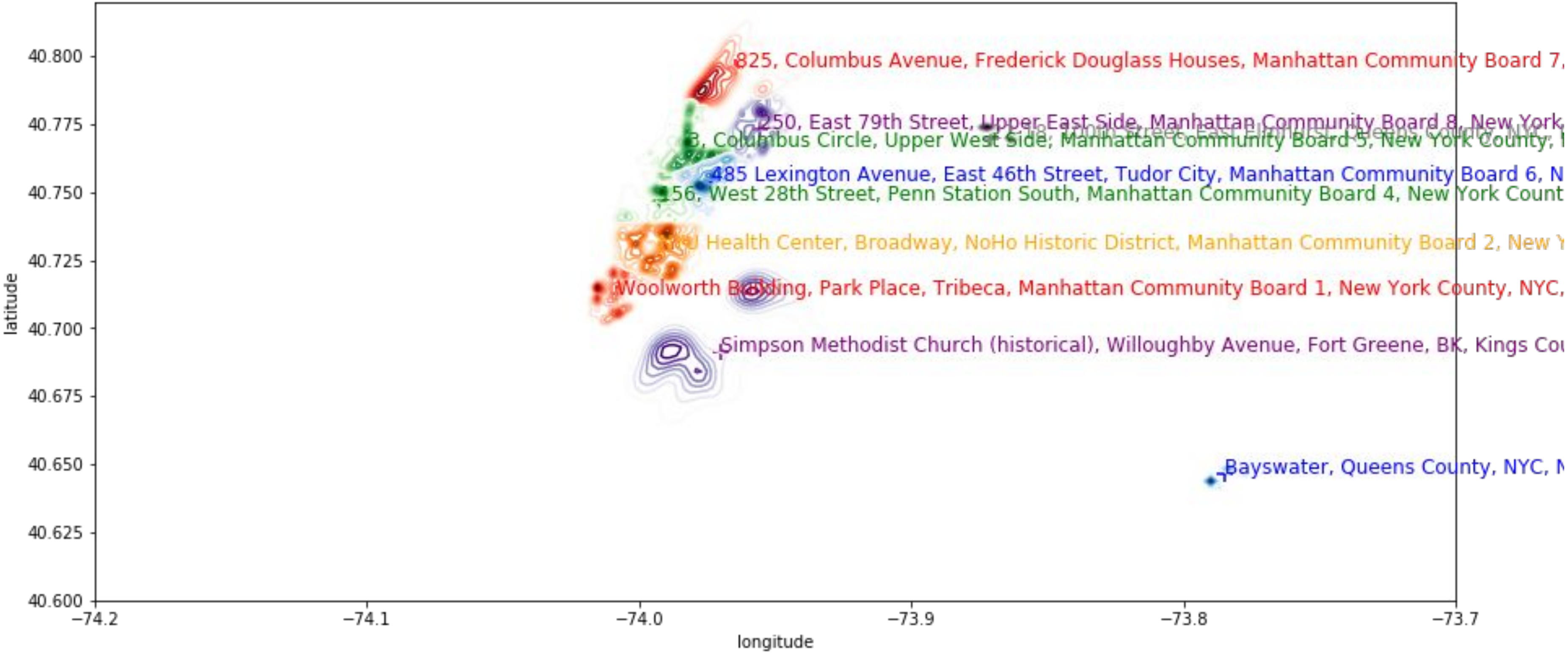


Coordinate Embedding Example

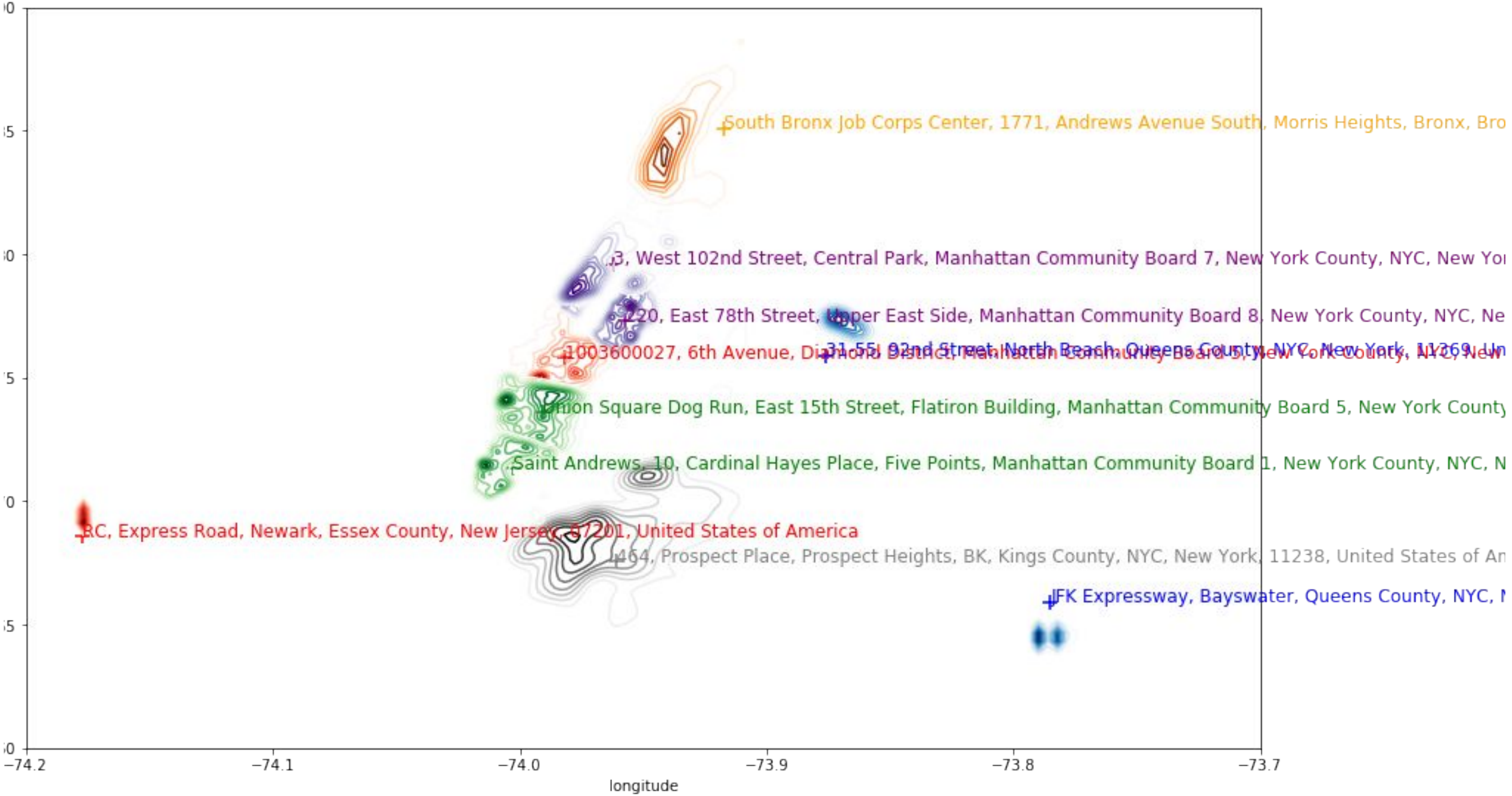
cluster	p_x	p_y	distance
0	-2377.885565	3607.939127	4321.060577
1	-755.140228	-2850.952545	2949.265532
2	-20221.077076	-3878.751157	20589.722403
3	-999.514160	2057.553115	2287.477514
4	-4797.905809	-2974.375511	5645.069515
5	-6264.539381	5694.434074	8465.874620
6	1109.488131	-3459.550444	3633.105172
7	-2086.670278	5757.588498	6124.052430
8	5352.696850	-9618.242088	11007.358649
9	-21.320613	-593.153911	593.536967



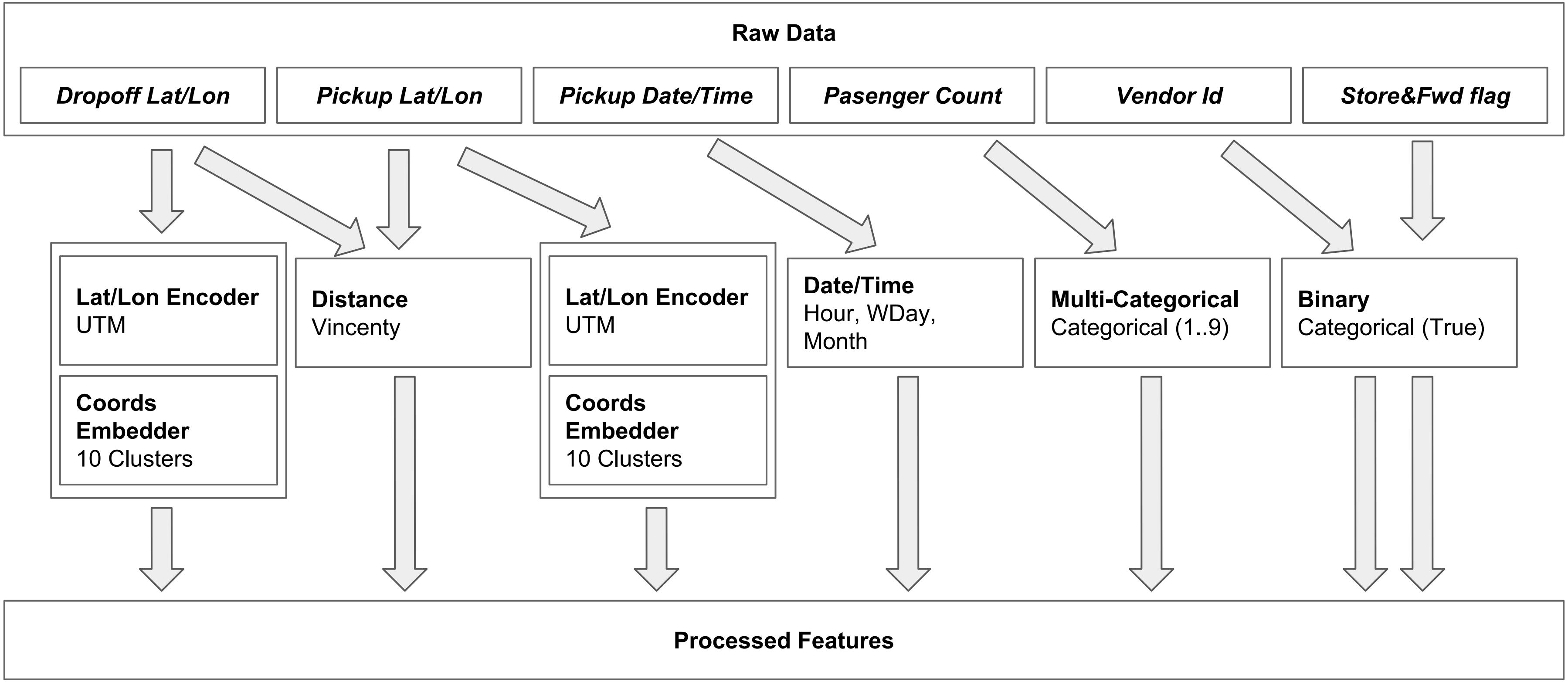
Pickup Clusters



Dropoff Clusters



Feature Engineering



Model Training & Evaluation

Strategy

Gradient Boosted Trees

- Motivation:
  - Mainstream in Competitions
  - Pretty Fast Training
- Implementation:
  - Sci-Kit Learn
- Validation Strategy:
  - Train Data: 90% (~1.2M)
  - Test Data: 10% (~120K)
- Hyper Params Selection:
  - Cross-Valdation w/ 3-Fold
  - RandomSearchCV

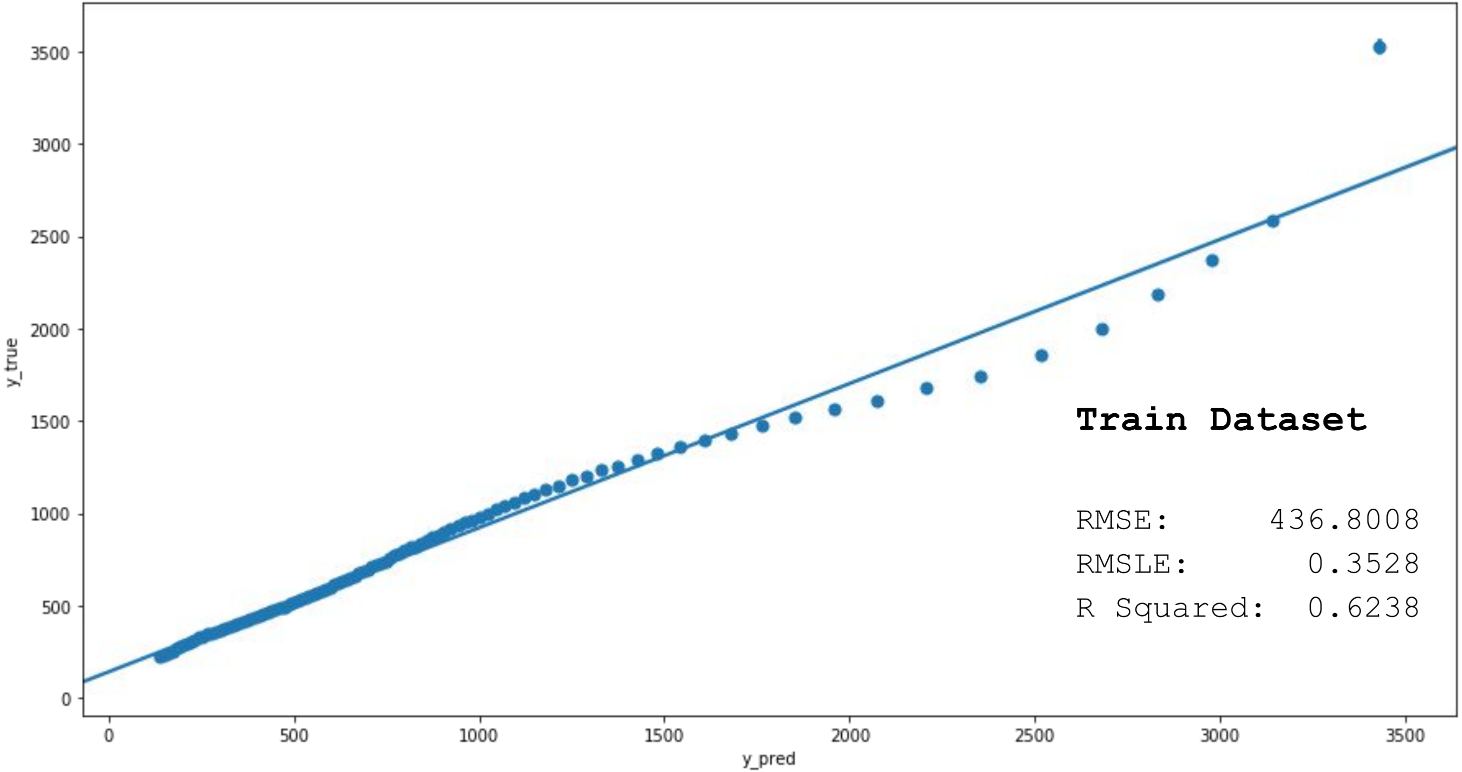
Artificial Neural Network

- Motivation:
  - Recent Developments
  - Personal Favorite
- Implementation:
  - Keras + TensorFlow
- Validation Strategy:
  - Train Data: 80% (~1M)
  - Validation Data: 10% (~120K)
  - Test Data: 10% (~120K)
- Hyper Params Selection:
  - Manual Settings based on convergence

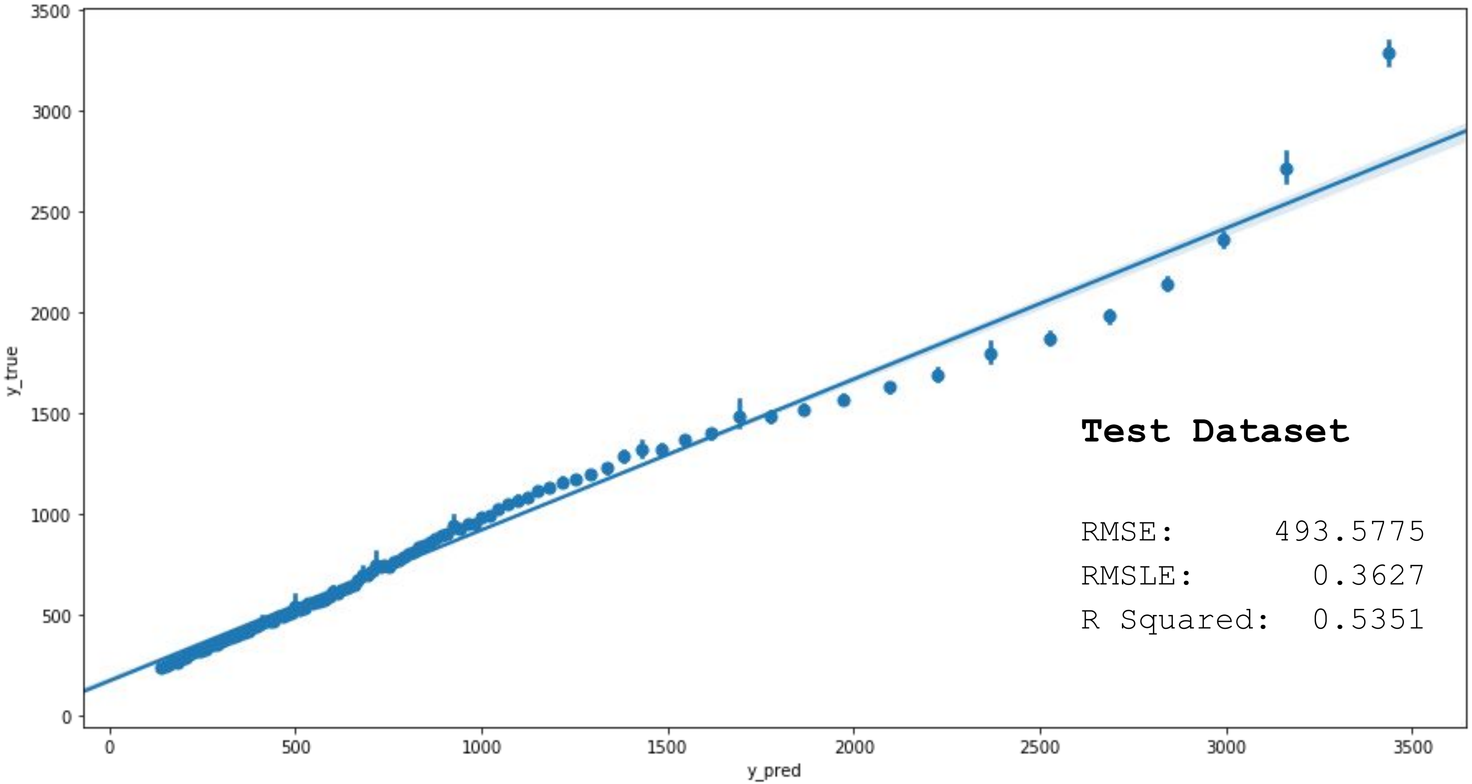
Gradient Boosted Trees

```
model_trainer = RandomizedSearchCV(
    n_iter=3,
    estimator=GradientBoostingRegressor(),
    param_distributions={
        "criterion": ["mse"],
        "loss" : ["ls", "lad", "huber", "quantile"],
        "learning_rate": [.3],
        "n_estimators": [100, 400, 1000],
        "max_depth": [3, 5],
        "max_features": ["auto", "log2", "sqrt"],
        "min_samples_split": [10, 50, 100],
        "min_samples_leaf": [10, 50, 100],
        "verbose": [1]
    },
    verbose=True,
    refit=True,
    cv=3,
    n_jobs=-1
)
```

Train Data Evaluation (GBT)

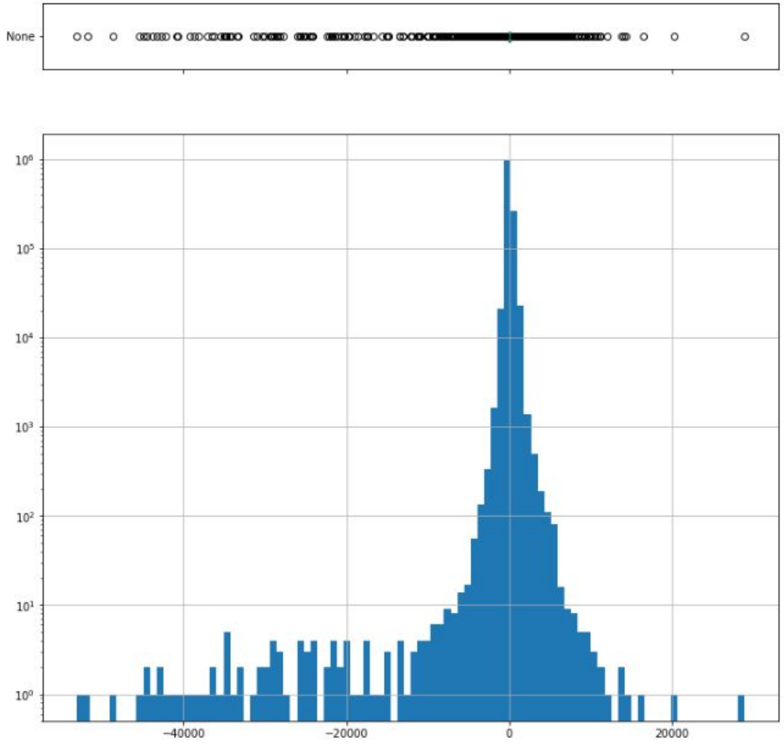


Test Data Evaluation (GBT)



Error Analysis on Test Data (GBT)

count	142233.000000
mean	**58.161655**
std	**490.140437**
min	**-50718.016371**
0.1%	-1947.865384
2.3%	-618.262730
15.9%	-169.218073
50%	26.963351
84.1%	281.570376
97.7%	990.189996
99.2%	1303.457544
99.9%	2055.188628
max	**29848.400339**



Error histogram: y-axis in ***log scale***

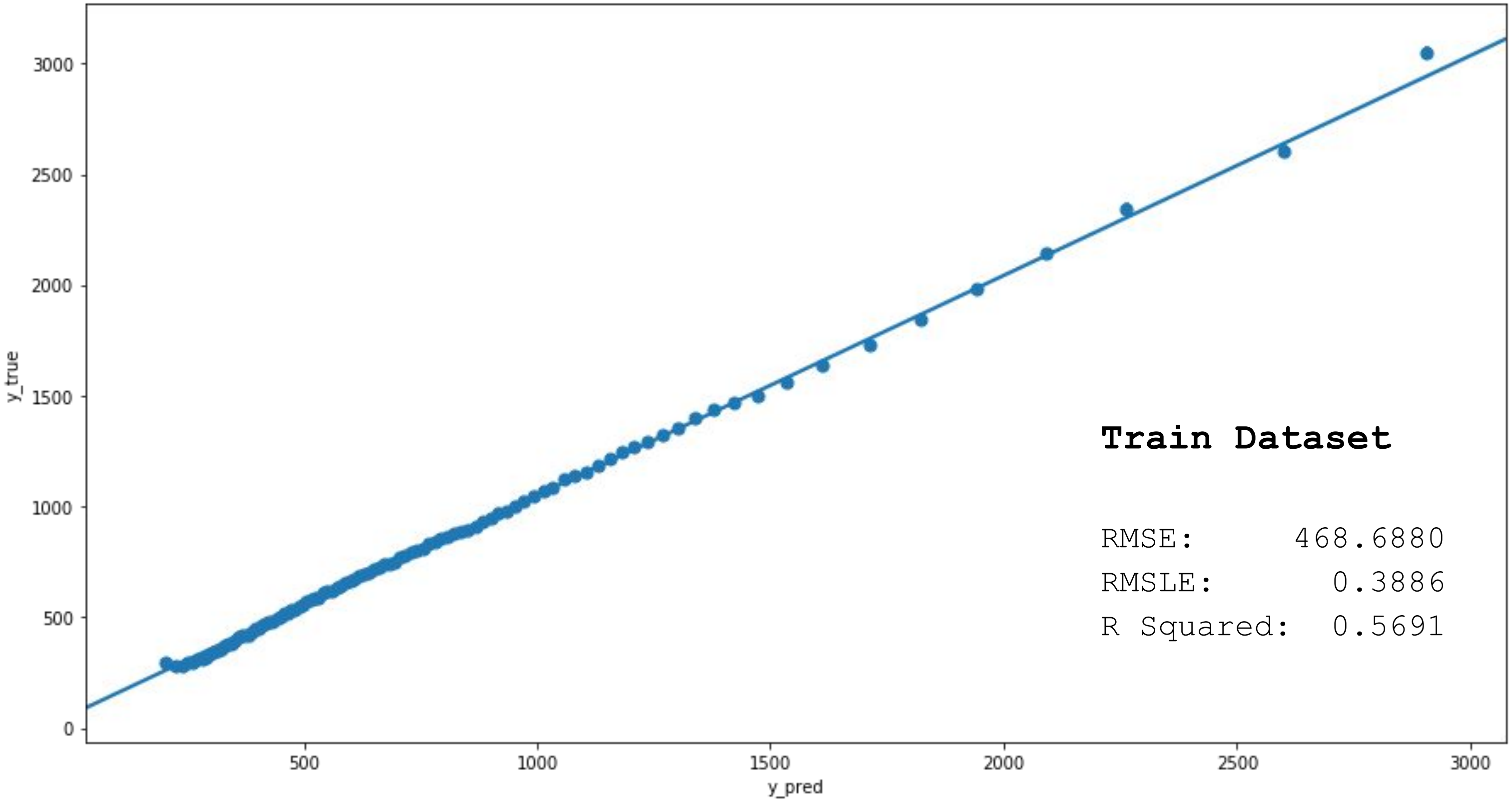
Feature Importances (GBT)

	feature_importances
pickup_coords_projection	43.33%
dropoff_coords_projection	41.74%
pickup_datetime__hour	5.43%
vincenty_pickup_dropoff	4.72%
pickup_datetime__weekday	2.72%
pickup_datetime__month	1.41%
vendor_id__1	0.30%
passenger_count__1	0.25%
store_and_fwd_flag__y	0.10%
passenger_count__others	0.00%
TOTAL	100.00%

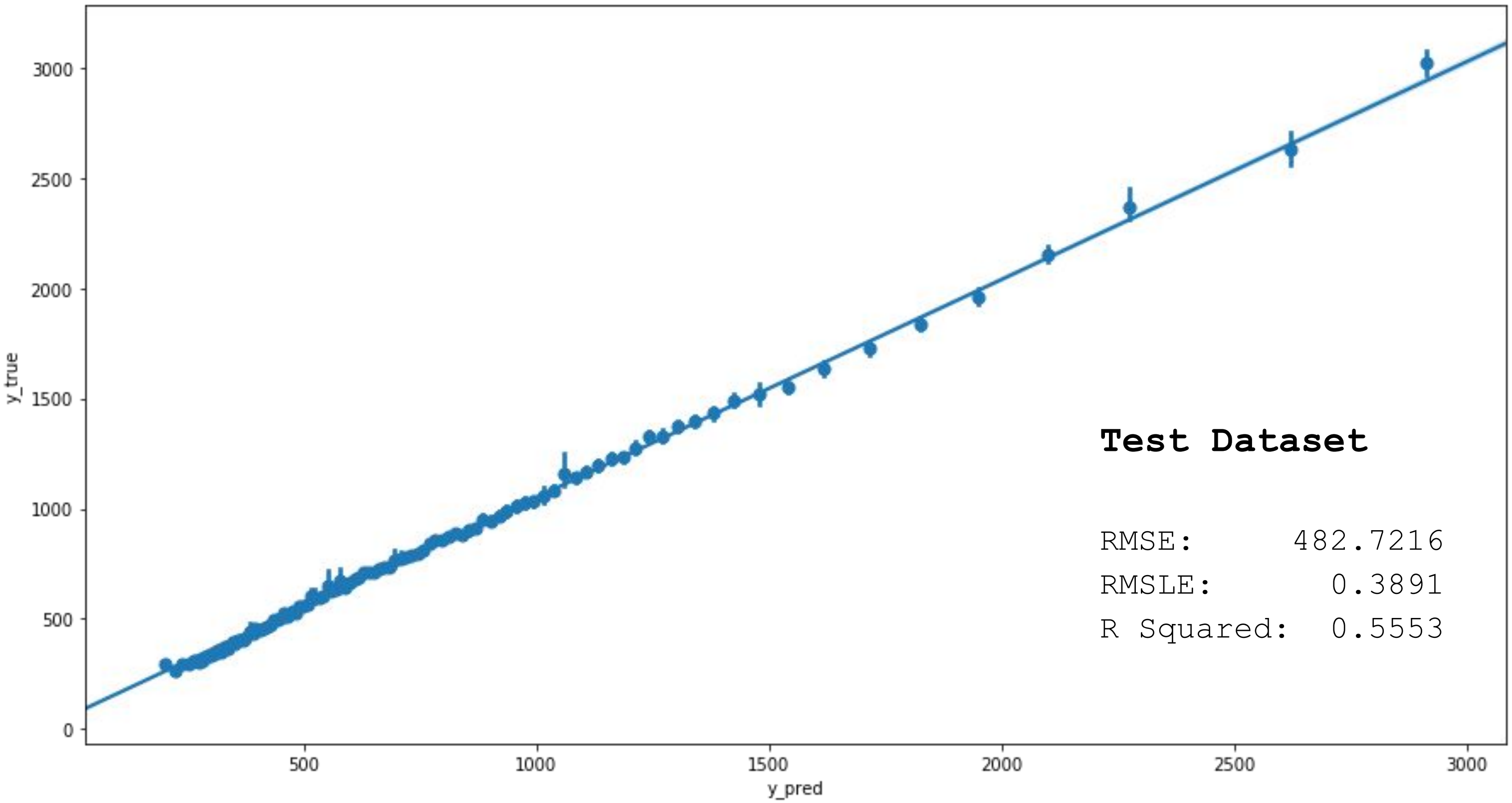
Artificial Neural Network

```
model = Sequential()
model.add(Dense(300, input_dim=X_tr.shape[1], kernel_initializer="glorot_uniform", activation='relu'))
model.add(Dropout(0.05))
model.add(BatchNormalization())
model.add(Dense(50, input_dim=300, kernel_initializer="glorot_uniform", activation='relu'))
model.add(Dropout(0.05))
model.add(Dense(1, kernel_initializer="glorot_uniform", activation='relu'))

optimizer = RMSprop()

model.compile(loss='mean_squared_error', optimizer=optimizer)
monitors = [
    EarlyStopping(monitor='val_loss', min_delta=0.001,  patience=500, mode='auto'),
    ModelCheckpoint(filepath=filename, monitor='val_loss', mode='auto',
                    save_best_only=True, save_weights_only=False)
]

model.fit(X_tr, y_tr, batch_size=50000, epochs=1000,  validation_data=(X_vl, y_vl), callbacks=monitors)
```

Train Data Evaluation (ANN)

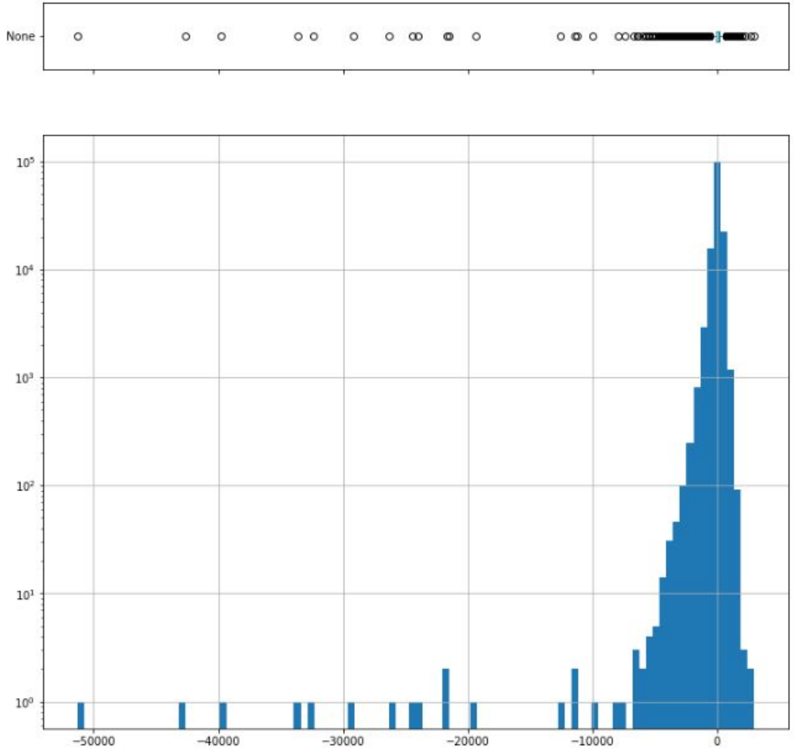


Test Data Evaluation (ANN)



Error Analysis on Test Data (ANN)

count	142233.000000
mean	**11.578110**
std	**479.724030**
min	**-51326.054443**
0.1%	-2550.644929
2.3%	-917.822066
15.9%	-237.708192
50%	70.856445
84.1%	276.725690
97.7%	604.337773
99.2%	829.732268
99.9%	1252.383570
max	**2985.758301**



Error histogram: y-axis in ***log scale***

Execution Times

Submission Dataset

- Dimensions:
  - Rows: **625,134**
  - Columns :
    - Prev Feat. Eng.: **8**
    - Post Feat. Eng.: **55**
- Feat. Eng. Execution Time:
  - CPU time: **7min 32s**
  - Wall time: **8min 38s**

Artificial Neural Network

- Predict Execution Time:
  - CPU time: **41.3 s**
  - Wall time: **48.8 s**

Gradient Boosted Trees

- Predict Execution Time:
  - CPU time: **10.5 s**
  - Wall time: **12.8 s**

Kaggle Leader Board

Submission and Description	Private Score	Public Score	Use for Final Score
submission_pred_rna.csv an hour ago by Allan Dieguez Keras: 3 RELU layers Neural Network.	0.49674	0.49473	☐
submission_pred_gbt.csv an hour ago by Allan Dieguez GBX without negative predictions (no time travels this time)	0.54462	0.54009	☐
submission_pred_gbt.csv an hour ago by Allan Dieguez Scikit-Learnn's Gradient Boosted Trees.	NULL	Error	☐

Next Steps

Feature Engineering

Coordinates Embedding

- ❏ Automate research for number of Clusters
- ❏ Test other Latitude/Longitude projections other than Universal Transverse Mercator (UTM)
    - ❏ Military Grid Reference System (MGRS)
    - ❏ United States National Grid (USNG)
    - ❏ Global Area Reference System (GARS)
    - ❏ World Geographic Reference System (GEOREF)
- ❏ Embed more than coordinates
    - ❏ e.g. date/time features
- ❏ Reimplement projection module to use vector operations
    - ❏ more RAM needed but way faster

External Data

Routes

- Best/Shorter/Fastest Routes:
  - Open Street Maps
  - Google Maps / Waze
- Transit Reports:
  - http://www.nyc.gov
  - Google Maps / Waze

News

- Google News

Weather Forecast

- http://www.weather.gov/
- Google Weather

City Events

- www.ticketnetwork.com
- http://www.nyc.gov/events
- https://www.eventbrite.com

Q&A

Thank You