Discover LinkedIn Job Seeker’s Commute Preference

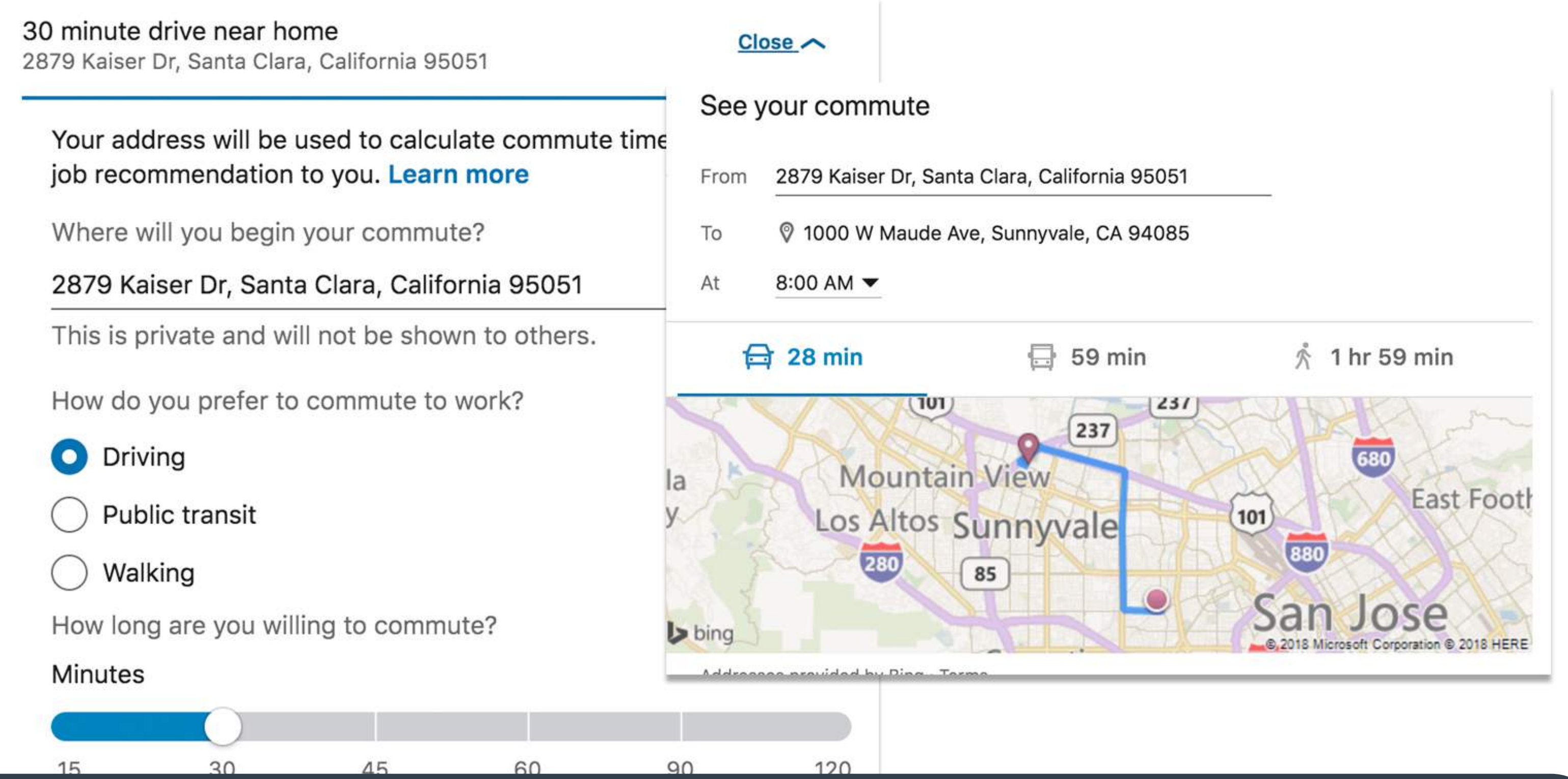
Dianxia Yang (dianxiay@Stanford.edu)

CS 229 Machine Learning

Background & Challenge

LInkedIn rolled out a feature to allow job seekers to provide us their job commute preference, including starting point, preferred way to commute and maximum commute time. And we will be using this data to provide more relevant job recommendations.

All commute preference data are currently organically input by the active members who discovered this feature.



30 minute drive near home
2879 Kaiser Dr, Santa Clara, California 95051

Close

Your address will be used to calculate commute time job recommendation to you. Learn more

Where will you begin your commute?

2879 Kaiser Dr, Santa Clara, California 95051

This is private and will not be shown to others.

How do you prefer to commute to work?

● Driving
○ Public transit
○ Walking

How long are you willing to commute?

Minutes: 30

See your commute

From: 2879 Kaiser Dr, Santa Clara, California 95051
To: 1000 W Maude Ave, Sunnyvale, CA 94085
At: 8:00 AM

28 min | 59 min | 1 hr 59 min

Problem Statement

We believe there are some correlation between member’s job seeking activity and their implicit commute preference. The goal of the project is to find out:
Member’s Willingness: Whether a member would love to provide commute preference information.
Potential Commute Duration: What is member’s most possible commute duration if preference provided.
With this prediction, we could potentially 1. Promote feature with precise target and personalized copy. 2. Leverage implicit inferred data for job recommendation

Dataset

All data we are leveraging here are persisted in our HDFS
1. Standardiized LinkedIn member data, including derived member location.
2. LinkedIn member job activity tracking events, including job view / search / apply / save.
3. LinkedIn member’s careers preference data
4. Standardized geo data, with latitude/longitude of locations.
5. Other pre-derived member job related data

Feature Generation

1. Selected member derived data: industry, job activity score derectly as feature
2. With member’s standardized lat/long, generate distance vectors for **view/search/apply/save/preference location** data with following equation.

$$d = 2r \arcsin\left(\sqrt{\sin^2\left(\frac{\phi_2 - \phi_1}{2}\right) + \cos(\phi_1)\cos(\phi_2)\sin^2\left(\frac{\lambda_2 - \lambda_1}{2}\right)}\right)$$

3. For each generated distance vectors for action X, we produce the following features: X_count, X_avg_distance and X_std.

Data Preprocess

1. For Problem #1, the binary problem, generate nagative data label under the same envrionment for testing.
2. For Problem #2, the multi-class problem, rewrite the original 6 ranges of duration into 3 ranges:
[15,30] -> 0, [45,60] -> 1, [90,120] ->2
3. Deal with missing values for linear models. We removed samples without job view data, and then populate missing values of search/apply/save/preference data as a chain from business logic.

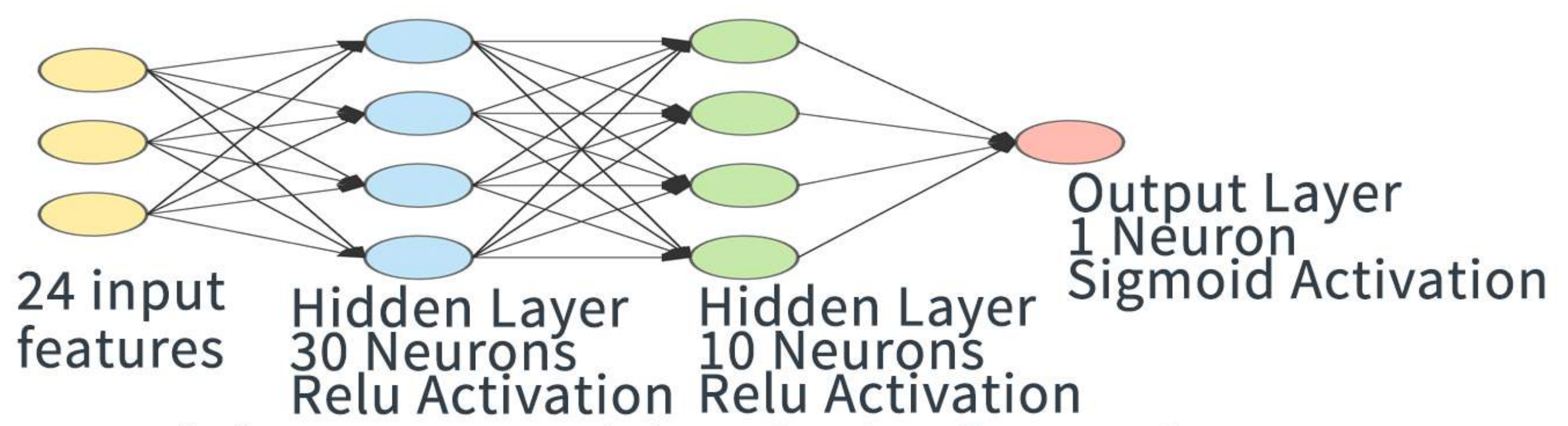
Models for Willingness Problem (Binary)

In general we tuned the parameters for the following algorithms via cross validation over the training set.

1. Logistic Regression (binary configuration with L2 regulariation) to optimize the following loss function:

$$J(\mathbf{w}) = \sum_{i=1}^{n}\left[-y^{(i)}\log\left(\phi(z^{(i)})\right) - (1 - y^{(i)})\log\left(1 - \phi(z^{(i)})\right)\right] + \frac{\lambda}{2}\|\mathbf{w}\|^2$$

2. Neural Network



24 input features
Hidden Layer 30 Neurons Relu Activation
Hidden Layer 10 Neurons Relu Activation
Output Layer 1 Neuron Sigmoid Activation

3. Simple Decision Tree with Min-leaf pruning. We are trying to maximize the reduction of Gini loss in each step:

$$H(X_m) = \sum_k p_{mk}(1 - p_{mk})$$

4. XGBosst, which follows this objective function in each step:

$$\sum_{i=1}^{n}[g_i f_t(x_i) + \frac{1}{2}h_i f_t^2(x_i)] + \Omega(f_t) \qquad g_i = \partial_{\hat{y}_i^{(t-1)}} l(y_i, \hat{y}_i^{(t-1)}) \quad h_i = \partial^2_{\hat{y}_i^{(t-1)}} l(y_i, \hat{y}_i^{(t-1)}) \qquad \Omega(f) = \gamma T + \frac{1}{2}\lambda\sum_{j=1}^{T} w_j^2$$

Models for Duration Problem (Multi Class)

Since we are using the same feature set for this problem, we are also leveraging similar models but with different settings.

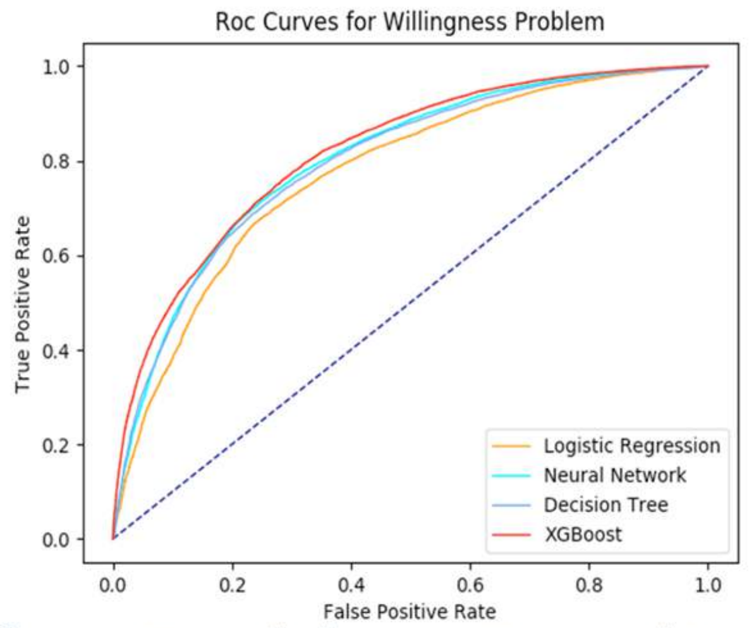
0. Deal with Imbalanced Data: The distribution of the data label is highly imbalanced, I’ve calculated the class weight with the below equation, which effectively panalizes more on minority sample mistake. $w_j = \frac{n}{kn_j}$

1. Logistic Regression with configuration of softmax with L2 regularization
2. Neural network with softmax output layer
3. Simple decision tree with tuned similar prunings
4. XGboost with tuned similar settings

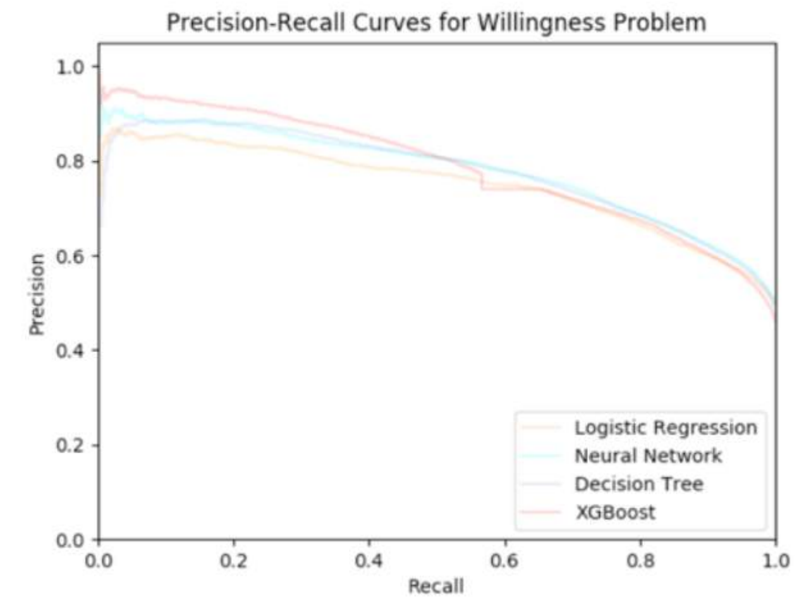
Test Result

WIllingness Problem Result: Note that we are reporting average preceision score because this problem is subject to precision-recall analysis to make product decision.

Model	Training Accuracy	Test Accuracy	Test Average Precision Score	Test ROC AUC Score
Logistic Regression	0.7102	0.7120	0.7482	0.7730
Neutral Network	0.7297	0.7324	0.7810	0.8018
Simple Decision Tree	0.7494	0.7258	0.7758	0.7968
XGBoost	0.7429	0.7367	0.7859	0.8165



Roc Curves for Willingness Problem



Precision-Recall Curves for Willingness Problem

Duration Problem Result:

Model	Training Accuracy	Test Accuracy	Precisions	Recalls	F1-Scores
Logistic Regression	0.4766	0.4758	0 – 0.5694 1 – 0.6335 2 – 0.1085	0 – 0.6228 1 – 0.3278 2 – 0.3774	0 – 0.5950 1 – 0.4320 2 – 0.1686
Neutral Network	0.5014	0.4954	0 – 0.5711 1 – 0.6142 2 – 0.1149	0 – 0.6702 1 – 0.3281 2 – 0.3205	0 – 0.6167 1 – 0.4278 2 – 0.1692
Simple Decision Tree	0.4779	0.4539	0 – 0.5955 1 – 0.6003 2 – 0.1082	0 – 0.5893 1 – 0.3015 2 – 0.4691	0 – 0.5923 1 – 0.4014 2 – 0.1758
XGBoost	0.6023	0.5845	0 – 0.5727 1 – 0.6115 2 – 0.2500	0 – 0.8084 1 – 0.4224 2 – 0.0032	0 – 0.6704 1 – 0.4997 2 – 0.0063

Discussion & Future Work

1. The binary willingness problem illustrated a good result, and we could explore productionize it.
2. The multi-class duration problem’s performance is poor. Potential reasons are: 1. Not enough data for minority classes. 2. The features we’ve chosen could not effectively distinguish commute duration in minutes granularity.
3. We should explore leveraging RNN for the vector features.
4. We should explore populating missing values with regreassion algorithms