Simon Chau 1460077
Hiu Fung Kevin Chang 1443481

CMPUT 466 Mini Project

Problem Specification and Question

Given information about consumers on Black Friday relating to their gender, marital status, age, etc, can we accurately predict how much they will spend on Black Friday?

Description of the dataset:

Dataset of observations about the black Friday in a retail store, it contains different kinds of variables either numerical or categorical with missing values.

The missing values were in the Product Category columns. Based on our interpretation of the data, a missing value from these columns suggests a value of 0. So these missing values are filled with 0s.
Columns such as gender, age, City_category, Stay_In_Current_Years, were objects which we converted to numerical representations.
Gender: M = 0, F = 1
Age: bins changed to 0, 1, 2, …
City_Category, A=0, B=1, C=2
Stay_In_Current_Years = 0-4

We assume that Occupation and Product_Category were already discretized into numerical representations. When we say discretized, we mean for example, that if we see the value 17 under Occupation, then this person has occupation 17, which could be something like a doctor.

Number of Samples:

- 537577 samples (according to dataset.info(), despite Kaggle saying there's 550000)

Number of Features and their types:

- We'll look at 9 features from the dataset.

- Gender:	discrete, M or F
- Age:	discrete, bins of age ranges
- Occupation:	discrete, discretized ints
- City_category:	discrete, A, B, or C
- Stay_In_Current_City_Years	discrete 0-3, 4+
- Marital_Status:	discrete, 0 or 1
- Product_Category_1:	discrete, discretized ints
- Product_Category_2:	discrete, discretized ints
- Product_Category_3:	discrete, discretized ints

- User_ID and Product_ID are also in the dataset, but these are unique keys rather than features, so we'll disregard them.

Target Variable:

- Purchase: a continuous variable representing the purchase amount in dollars.

Brief description of the importance of the data:

This dataset came from Kaggle[1], and is meant to be used for learning purposes. The topic has relevance considering the proximity of Black Friday to the date this project is being done. It is also common for businesses to try and predict demographics' spending for marketing purposes.

Design of experiment

We will use internal cross validation to determine the best metaparameters for the algorithms. Cross validation function[2] provided by Scikit-learn will be used.

How the data will be split

We have a large amount of data so we can test on a sizable subset and see if it'll generalize to the rest of the data.

We shuffle our data, then separate the target variable Purchase from the other features. Then we plan to train on around 80% of the samples with 400000 rows and test on 100000 rows.

What statistical significance tests will be used

Given the size of the dataset, it seems reasonable to assume that the errors in our distributions will be distributed normally due to the Law of Large Numbers, and so we could use the paired t-test to compare our algorithms. We compare the mean absolute error between the predicted values made by the algorithms and the targets. We use the mean absolute error due to its ease of interpretation.

Algorithms chosen:

Since this a regression problem, we chose the following three algorithms.

1. Ridge
    - The output is continuous, and it's possible our features could have a linear relationship with our output. We assume the weights have a zero-mean Gaussian prior. Linear regression is typically fast to compute compared to other algorithms, and regularization will help us prevent overfitting.

2. Neural Network
    - Neural networks are known for being very strong in non-linear modelling, and performs well with larger datasets.

3. Random Forest Regression
    - Fits a number of decision trees on various subsets of the data and uses averages to improve predictive power and prevent overfitting
    - Random Forest Regression also determines feature importance in its calculations which is particularly useful for the Black Friday dataset as connections between the features and the targets aren't easily interpretable

A clear description of the parameters you tuned:
All our algorithms use weights that transform our inputs into predictions for our targets. Each algorithm has their own metaparameters, and we'll use cross-validation to adjust/tune said metaparameters.

The metaparameters that we will change for each algorithm are:

1. Ridge Regression[3]
    - Regularization parameter. A variable that determines the degree of regularization we'll use. (Alpha values)
        - We'll try these values { 0.1, 1.0, 10.0 }

2. Neural Network[4]
    - Number of nodes: the number of nodes in the hidden layer(s) for our NN
        - We'll try these values { 4, 8, 16 }
    - Step size, a.k.a learning rate, how much we shift our weights per iteration.
        - We'll try these values { 0.1, 0.01, 0.00001 }

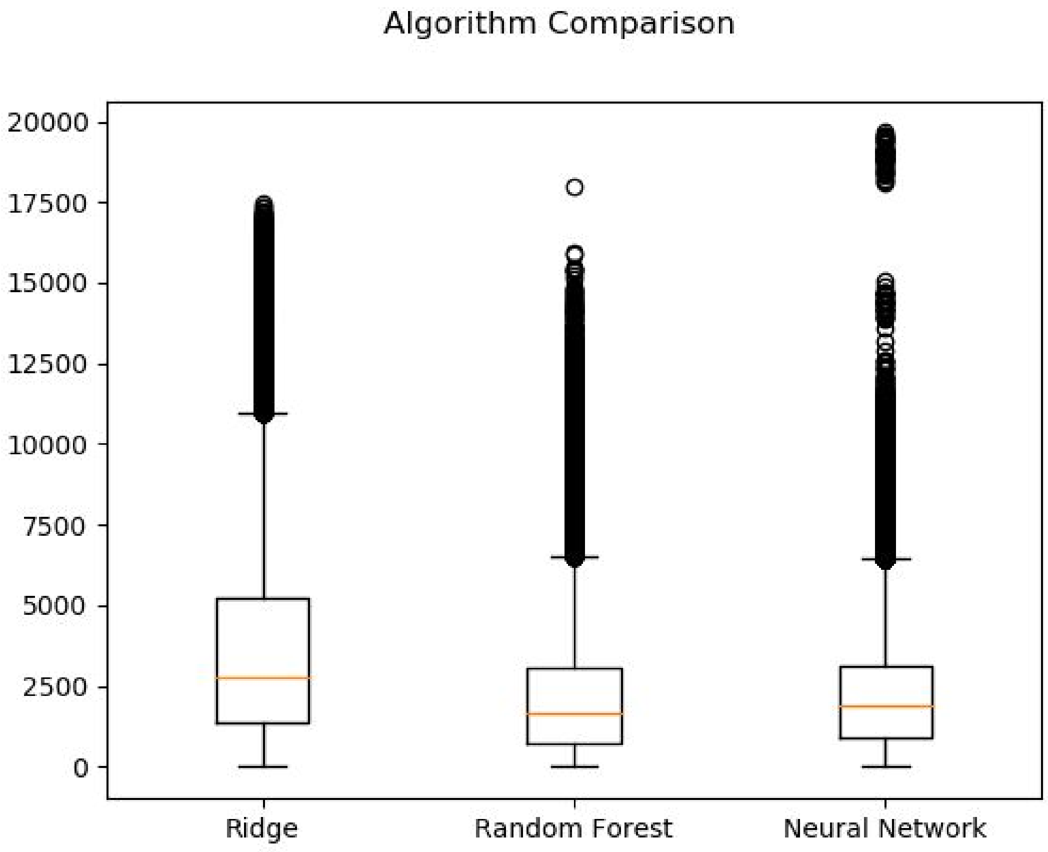
3. Random Forest Regression[5]
    - Number of estimators: the number of "trees" in the "forest"
        - We'll try these values { 20, 100, 200 }

Other metaparameters are set to the default settings as determined by Scikit-learn.

A detailed description of the methodology followed:

1. Randomize the data.
2. Separate the data into its features and the target variable 'Purchase'
3. Separate a subset of the data into an approximate 80% of the data training set, and a 20% holdout test set.
4. Perform cross validation using 5 folds on the training set to decide on good metaparameters for Ridge Regression, Random Forest Regression, and the Neural Network to determine what the best metaparameters are for each algorithm.
5. Start learning the model for each of the three algorithms over 5 runs using the determined best metaparameters, then determine the accuracy in predicting outputs using the test set by examining the mean absolute error.
6. The best model would be chosen by performing a paired t-test on every pair of models, and plotting a boxplot with the mean and standard deviation of the errors. The model with its best parameters that has the lowest mean absolute error is our "winner", and we'll examine the p-values to determine if the difference in our models is statistically significant.

Presentation of the results in a table/graph showing statistical confidence:
Cross-validation selected 10.0 as the best alpha parameter for Ridge Regression.
80 estimators as the best number of estimators for RFR.
And 16 hidden nodes with a step size of 0.1 for the Neural Network.
We compare the mean absolute errors in the predictions made by each algorithm on the test set in the box plot below.



Statistical Significance Analysis

Using the paired-t test:
We used scipy.stats.ttest_rel[6] to compute the p-values, which default to 0 if it’s too small.
We use a confidence level of 95%.

Null hypothesis: Ridge Regression error and Random Forest Regression error are the same
Ridge Regression Mean Absolute Error: 3561.305246711606
Random Forest Regression Mean Absolute Error: 2235.2593782263243
p-value: 0.0
Conclusion: Reject null hypothesis.

Null hypothesis: Ridge Regression error and Neural Network error are the same
Ridge Regression Mean Absolute Error: 3561.305246711606
Neural Network Mean Absolute Error: 2436.1929997935786
p-value: 0.0
Conclusion: Reject null hypothesis.

Null hypothesis: Random Forest Regressor error and Neural Network error are the same
Neural Network Mean Absolute Error: 2436.1929997935786
Random Forest Regression Mean Absolute Error: 2235.2593782263243
p-value: 8.709148150616216e-238
Conclusion: Reject null hypothesis

An analysis stating the “winner” algorithm and why that might be the case:

The Random Forest Regressor with 80 estimators is our “winner”. We believe this to be the case to due to RFR having implicit feature selection, which may have “eliminated” some unnecessary features, leading to a better model than the others.

We believe Ridge Regression performed poorly due to the data not being very linear.
And we believe the Neural Network performed worse than RFR due to an inability to converge during some runs in the training with its given parameters.

References:

1. Dataset
   https://www.kaggle.com/mehdidag/black-friday
2. Library for cross validation
   https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.cross_validate.html
3. Library for Ridge regression
   https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.Ridge.html
4. Library for Neural Network
   https://scikit-learn.org/stable/modules/generated/sklearn.neural_network.MLPRegressor.html#sklearn.neural_network.MLPRegressor
5. Library for Random Forest Regression
   https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestRegressor.html
6. Library for statistical test
   https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.ttest_rel.html