Machine Learning Final Project

Nhat Hoang Pham

University of Colorado, Denver

MATH 5388

Professor Farhad Pourkamali

Abstract

In this project, I analyzed the problem , design a manche learning solution, implemented learning algorithms, and evaluated them on two data sets(Additive Manufacturing data set for regression and Telecom Churn prediction for classification). I developed several models to obtained around 0.7-0.8 accuracy scores and the performance of the trained models and their assumptions.

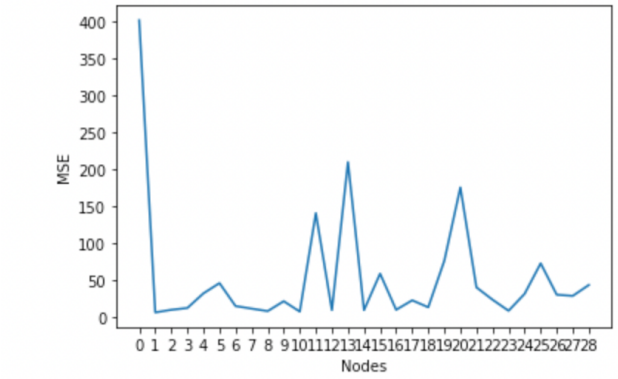
laws.

Keywords: Machine learning, Regression, Classification,

Regression with Additive Manufacturing Data Set.

Data set is 3D print data set of additive manufacturing test conditions is available for Polylactic Acid (PLA) and Acrylonitrile Butadiene Styrene (ABS). PLA can print at lower temperatures of 180°C compared to 250°C for ABS. PLA is more brittle than ABS and is not typically suitable for high strength applications. The data was collected by researchers in the Mechanical Engineering department at Selçuk Üniversitesi on a Ultimaker S5 3D printer. Nine parameters were adjusted for 3D printer Layer Height (mm)Wall Thickness (mm)Infill Density (%)Infill Pattern (Honeycomb or Grid),Nozzle Temperature (ºC), Bed Temperature (ºC), Print Speed (mm/s),Material (PLA or ABS),Fan Speed (%), and our response variable is Tension Strength (MPa). First, I transformed (%)Infill Pattern (Honeycomb or Grid) into two dummies variables, and Material (PLA or ABS) also into two dummy Material (PLA or ABS) variables using OnehotEncoder(). Using Boxplot (see Appendix), I identified that the observation with index 63 is an outlier. After that, I fit Linear Regression, Random Forest Regressor and Multi-layer Perceptron regressor for this dataset. For MLP Regressor, I decided to choose 1 hidden layer and with 7 nodes after testing all possible number of nodes (from 1 to 20) for 1 hidden layer. Below is the summary table of Mean Squared Errors and R-squared:

	MSE	*R-squared*
Linear Regression	5.658	0.906
Random Forest	12.847	0.7856
MLP (7nodes)	14.5	0.746

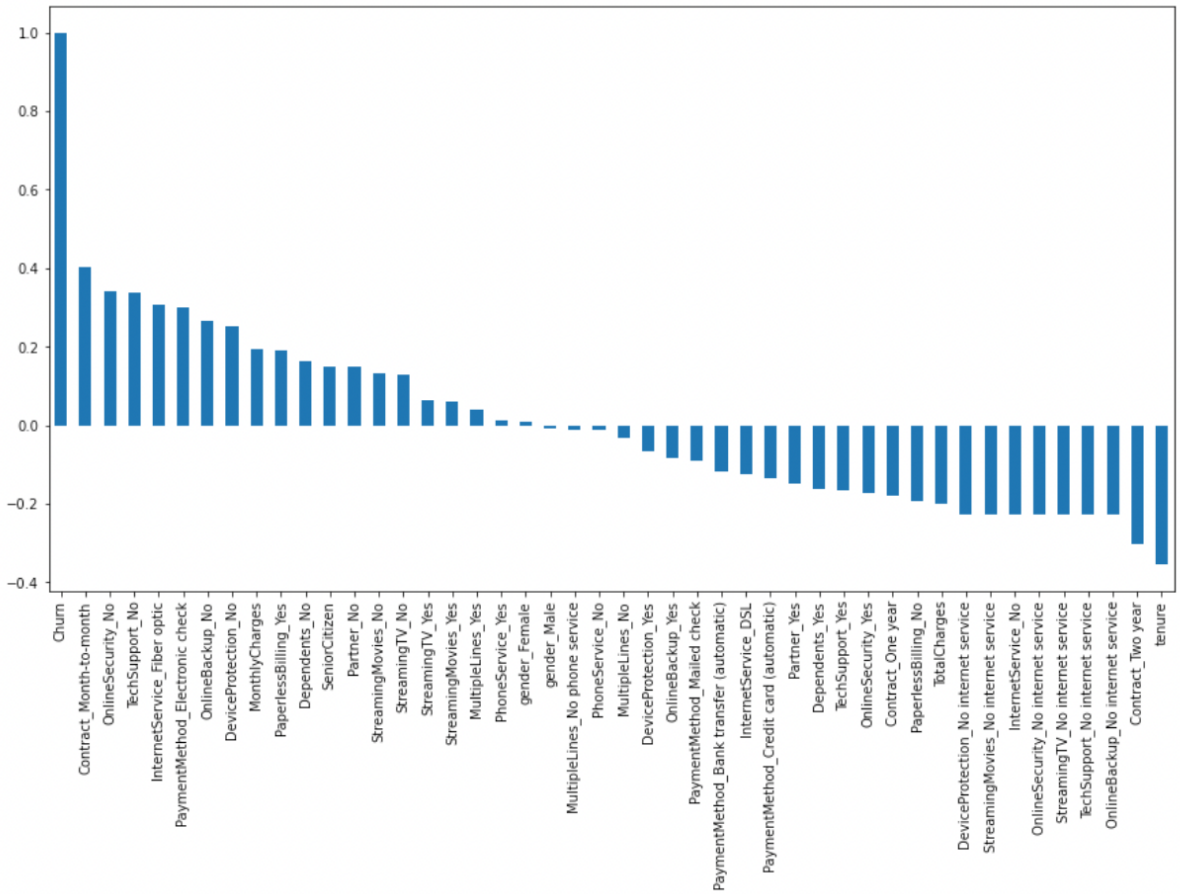


I believe the result make sense. It a good model with R-squared is 0.8 for testing data. It somehow underfit the model. However, a typical underfit model would be characterized by low R-squared values for both training and testing data. I used 100 random trees for random forest and 7 nodes with one hidden layer for MLP regressor. In short, linear regression performs the best among the three models, with the lowest MSE and the highest R-squared value. Random forest performs moderately well, while the MLP model with 7 nodes performs relatively poorer compared to the other two models in terms of both MSE and R-squared. This result suggests that there is strong evidence that a linear relationship between independent and dependent variables. In scenarios where the relationship is indeed linear, linear regression can capture it effectively. On the other hand, Random Forest and MLP are far more complex models that have high chances to overfit the data, especially when data set is small. This is exactly what we have here since we only have 70 observations and just 12 variables. However, the difference in mean squared errors and R-squared between 3 models are reasonable, not big enough for us to question the overfitting phenomenon.

Machine Learning Final Project

Classification with Telecom Churn Prediction Data Set.

Telecom churn prediction refers to the task of predicting which customers are likely to cancel or "churn" their telecom services. Churn prediction is important for telecom companies to identify and retain customers who are at risk of leaving, allowing them to take proactive measures to reduce customer attrition and improve customer retention. First, I turned Total Charges into a numerical data type. Secondly, I converted all the categorical variables into dummy variables. Below is the correlation graph between “Churn” Variables and other variables. As we can see, the more services we provided for customers, the lower the churn rate would be.



Before doing any model, I standardized the data. First of all, Standardizing data ensures that all features are on a similar scale, which is important for many machine learning algorithms. Scaling the data helps prevent features with larger numeric ranges from dominating the learning process and giving undue importance to certain features during model training. On the other hand, I used KNN as one of my classfier.K-nearest neighbors is a distance-based classifier that classifies new observations based on similar measures (e.g., distance metrics) with labeled observations of the training set. Standardization makes all variables contribute equally to the similarity measures.

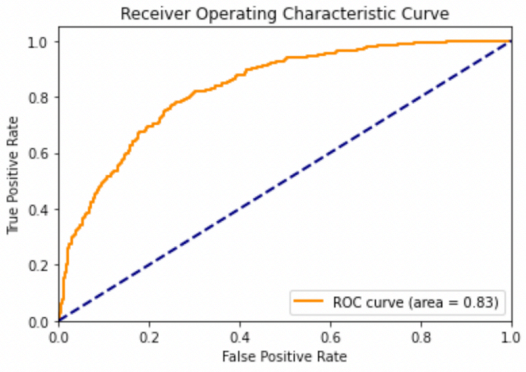
To predict churn in the telecom industry, various machine learning techniques I employed, including:

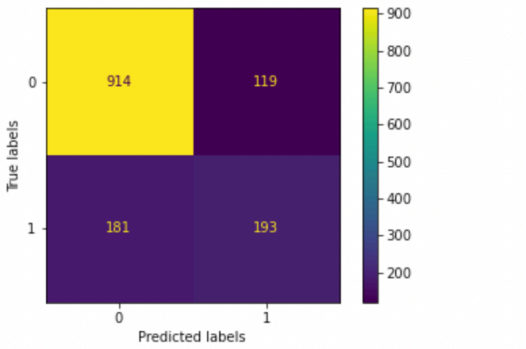
1. Logistic Regression:

   Logistic regression is a commonly used algorithm for churn prediction. It models the probability of churn based on customer features such as call duration, data usage, billing information, and customer demographics. Beside the metrics for Logistic Regression

	precision	recall	f1-score	support
0	0.83	0.88	0.86	1033
1	0.62	0.52	0.56	374
accuracy			0.79	1407
macro avg	0.73	0.70	0.71	1407
weighted avg	0.78	0.79	0.78	1407

AUC: 0.8319079986126282

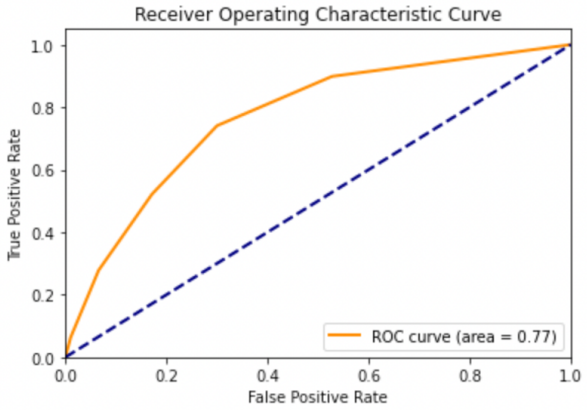


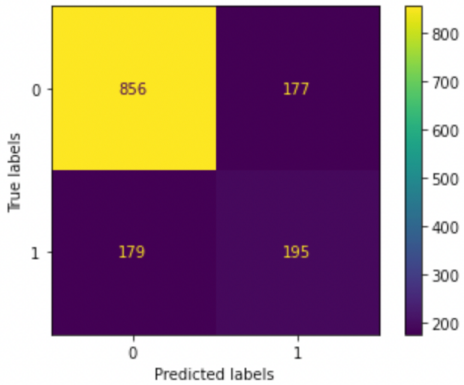


2. K- nearest- neighbor is a non-parametric supervised learning algorithm commonly used for classification tasks, including churn prediction in telecom. The algorithm determines the class of a new data point by considering the class labels of its k nearest neighbors in the feature space. The proximity or similarity between data points is typically measured using distance metrics such as Euclidean distance. By identifying the majority class among the k nearest neighbors, KNN can make predictions on whether a customer is likely to churn or not based on the characteristics of similar customers in the dataset.

	precision	recall	f1-score	support
0	0.83	0.83	0.83	1033
1	0.52	0.52	0.52	374
accuracy			0.75	1407
macro avg	0.68	0.68	0.68	1407
weighted avg	0.75	0.75	0.75	1407

AUC: 0.768052658007672

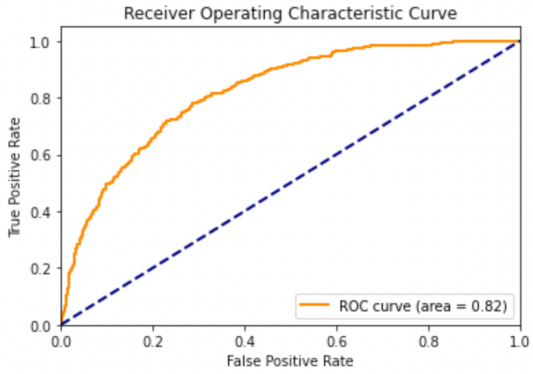


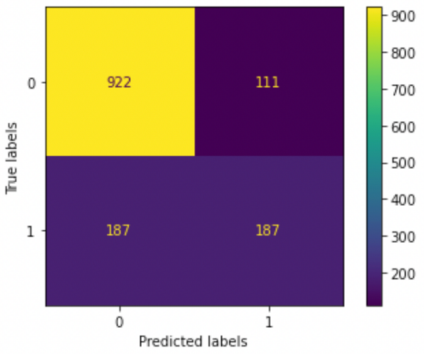


3. Multi-Layer Perceptron Classifier, which is a type of neural network model used for classification tasks. It consists of multiple layers of interconnected nodes (neurons) and is capable of learning complex patterns and relationships in the data.Since configuring the architecture and hyperparameters of the MLPClassifier, such as the number of hidden layers, number of neurons per layer, activation functions, and regularization techniques, may require experimentation and tuning to achieve optimal performance, I tried multiple nodes number for better performance, and chose 1 hidden layers with 7 nodes. Beside is the metrics for MLPClassfier,

	precision	recall	f1-score	support
0	0.83	0.89	0.86	1033
1	0.63	0.50	0.56	374
accuracy			0.79	1407
macro avg	0.73	0.70	0.71	1407
weighted avg	0.78	0.79	0.78	1407

AUC: 0.8205838350477039





In conclusion, there is no big difference between three classifiers. The accuracy score range is around 0.8, which is pretty good for the models. All other metrics pointed out that using 3 different Classifiers would lead us to a model. AUC number is around 0.8 indicate that we are not overfitting our data.