Application and Comparison of Machine Learning Techniques in Business

by

Shukhrat Khuseynov

A Dissertation Submitted to the
Graduate School of Sciences and Engineering
in Partial Fulfillment of the Requirements for
the Degree of

Master of Science

in

Data Science

KOÇ ÜNİVERSİTESİ

August 1, 2021

Application and Comparison of Machine Learning Techniques in Business

Koç University

Graduate School of Sciences and Engineering

This is to certify that I have examined this copy of a master's thesis by

Shukhrat Khuseynov

and have found that it is complete and satisfactory in all respects,

and that any and all revisions required by the final

examining committee have been made.

Committee Members:

Asst. Prof. David Carlson (Advisor)

Asst. Prof. Jeffrey Ziegler

Assoc. Prof. Özden Gür Ali

Date: December 2, 2021

ABSTRACT

Application and Comparison of Machine Learning Techniques in Business

Shukhrat Khuseynov

Master of Science in Data Science

August 1, 2021

Data science methodology and, particularly, machine learning techniques, are being widely used in many different fields today. There is the same trend in the private sector, where using machine learning is highly advantageous. The prediction of key variables in business given multiple input parameters is important and may affect the firm's profitability. This research aims to investigate several problems in business, being churn estimation, housing price prediction and sentiment analysis, which require a certain data framework and an algorithmic approach. The applied algorithms are compared for each case. The regarded cases involve an application of classification, regression, and text analysis types of machine learning models, demonstrating their diversity and useful application in the industry.

ÖZETÇE

Makine Öğrenmesi Tekniklerinin İşletmede Uygulanması ve Karşılaştırılması

Shukhrat Khuseynov

Veri Bilimleri, Yüksek Lisans

1 Ağustos 2021

Veri bilimi metodolojisi ve özellikle makine öğrenmesi teknikleri günümüzde birçok farklı alanda yaygın olarak kullanılmaktadır. Özel sektörde de kullanımı oldukça avantajlı olan makine öğrenmesini uygulama konusunda aynı eğilim bulunmaktadır. Birden fazla girdi parametresinin olduğu bilinen işletmedeki kilit değişkenlerin tahmini önemlidir ve bu, firmanın kar düzeyini etkileyebilmektedir. Bu araştırma, belirli bir veri çerçevesi ve algoritmik bir yaklaşım gerektiren iş dünyasındaki çeşitli sorunları; bu kapsamda müşteri kaybı tahmini, konut fiyat tahmini ve duygu analizini araştırmayı amaçlamaktadır. Uygulanan algoritmalar her bir vaka için karşılaştırılmaktadır. Ele alınan vakalar, makine öğrenmesi modellerinin sınıflandırma, regresyon ve metin analizi türlerinin birer uygulamasını içermekte ve bunların endüstrideki çeşitliliğiyle birlikte faydalı uygulamalarını da göstermektedir.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABBREVIATIONS

AUROC	Area Under the ROC curve
BoW	Bag-of-Words
KNN	K Nearest Neighbors
OLS	Ordinary Least Squares
RMSE	Root Mean Square Error
ROC	Receiver Operating Characteristic
SVM	Support Vector Machines
TF-IDF	Term Frequency – Inverse Document Frequency
XGBoost	Extreme Gradient Booster

Chapter 1:
INTRODUCTION

The application of data science and machine learning techniques is very important nowadays since it yields efficiencies and high profits in return, especially in the private sector. Other than direct business applications, artificial intelligence and machine learning can be observed in e-commerce, chatbot applications, banking, healthcare, cyber security, and commercial applications [Rao & Bhattacharyya, 2019]. The study also notes that this methodology brings gradually more and more security and profits to the industry, reducing the unneeded costs. Some of the recent examples for innovative business models include car-sharing and house-sharing services, and even manufacturing industry with Internet of things (IoT), utilizing big data and machine learning algorithms for better productivity [Jeong, 2018]. All different business tasks that use data science methodology generate vast amounts of data, sometimes even in real time, which then are taken as an input to a certain machine learning algorithm or a framework of algorithms to make some decision, prediction or detection. There is a wide range of learning models, depending on a type of problem. The main traditional models are regression analyses, Bayesian methods, random forests, and support vector machines. The more recent approaches also comprise deep learning algorithms, which outperform the classical models in some cases, especially for larger amounts of data [Kraus, Feuerriegel, & Oztekin, 2020]. Consequently, the comparison of algorithms for several business cases is worthwhile, emphasizing the diversity of machine learning algorithms and their beneficial application in the industry. Thus, this paper aims to compare a range of algorithms for three distinct business problems with a specific setup. The first chapter discusses the churn estimation, which involves predicting whether clients will stop doing business with a particular company, representing a binary classification task. Then, the second chapter describes the prediction of housing prices, given the key details of the apartments, using various regression models. Finally, the third chapter involves a sentiment analysis of e-commerce reviews, first implementing the text analysis, then the classification task to find a relationship between the reviews and the recommendations with the ratings given.

Chapter 2:
CHURN ESTIMATION

One of the common business problems in many companies is to find out which customers will potentially be using their products or services continuously in a long-term perspective, becoming loyal clients, and which ones will possibly churn. It is very important to identify these types of customers to manage the time and resources more efficiently, also to try to take precautionary measures to keep those who have the tendency to leave. The churn estimation is based on the machine learning algorithms of binary classification predicting whether a client will stop doing business with the company or not, given multiple parameters. The target variable, the churn of customers, usually represents a sparse vector due to the fact that only some customers tend to leave. The most common fields of churn estimation problem are telecom industry, insurance companies, banking sector, financial and subscription services [Siemes, 2016].

The data are obtained from a mobile company and anonymously shared on Kaggle[1]. The dataset consists of 66469 entries depicting the customers from a few months of 2013. There are 66 variables related to different measurements of calls, messages, durations and other information extracted via data mining techniques. The variables include the id, which is dropped before modeling since it is different for each customer and, therefore, useless in predictions, and the churn itself, which is the target variable in the models. The churn distribution can be observed in the data. Apparently, 79.1% of the customers are retained and 20.9% are exited customers. Such a distribution implies that guessing all customers are retained in the business regardless of any variable would result in almost 80% of accuracy score, which is the baseline to beat. The underrepresented group of exited clients has to be dealt with more carefully. For the same reason different measurements along with the accuracy score, such as the AUROC or, in other words, the area under the Receiver Operating Characteristic (ROC) curve, which constitutes a tradeoff between the true positive (TP) and the false positive (FP) rates, have to be used since the accuracy score can be misleading. The AUROC measures the overall performance of a classifier given any possible threshold for the probability between predicting one class or the other [Manďák & Hančlová, 2019].

[1] Can be retrieved from www.kaggle.com/dimitaryanev/mobilechurndataxlsx

Before the application of various machine learning models, the dataset is normalized using the min-max scaler, which normalizes by only keeping the relative distances between the data points. It is done to use the algorithms more efficiently and faster, because normalization allows some algorithms to approach the solution faster. Furthermore, after deciding the best parameters for each model, where it is required, in a grid search of 5-fold cross validation, maximizing the area under ROC (AUROC), 80% of the data are used for model training and the remaining 20% are for model testing purposes. In other words, if model requires, the cross validation is applied to the training set to choose the best hyperparameters with minimum overfitting, trying a range of reasonable values. Using these hyperparameters, the model is trained on 80% of the data and tested on the remaining 20%, being a holdout testing set, which is not touched before. The ROC curves of all models are summarized at the end.

The expected data generating process for the churn estimation assumes converting the cross-sectional data with various telecom parameters and dummy variables into either 0 or 1 for binary prediction, possibly using probability measure between 0 and 1 as a proxy.

The first model is the Naïve Bayes classifier. The Gaussian Naïve Bayes algorithm is used for the binary classification in this manner:

$$P(y \mid x) = \frac{P(y)P(x \mid y)}{P(x)} = \frac{P(y)\prod_{i=1}^{N} P(x_i \mid y)}{P(x)} \propto$$

$$\propto P(y)\prod_{i=1}^{N} P(x_i \mid y) \tag{2.1}$$

given that $P(x_i \mid y) = \frac{1}{\sqrt{2\pi\sigma_y^2}} \exp\left(-\frac{(x_i - \mu_y)^2}{2\sigma_y^2}\right),$

where y and x are the target, which is churn, and the features, respectively, with the mean and the variance of y being obtained from maximum likelihood estimation [Scikit-learn, 1.9.1]. It is among the simplest and fastest models for classification, using a simple formula with a naïve assumption of no covariance between the variables. In other words, the independence of all features is assumed since the data generating process of Naïve

Bayes ignores prior distribution of parameters. As an advantage, there is no limitation in assuming any prior distributions. However, due to maximum likelihood estimation, the algorithm is not iterative, which makes it more likely to underfit. It is also sensitive to outliers. It sometimes even outperforms the more complicated models, especially when the dataset is relatively small. Here, the calculated accuracy score is 56.21%, which is quite low but better than random, and AUROC is 69.77%, which is higher due to better accuracy for the underrepresented group, as seen in the confusion matrix represented in Figure 2.1.

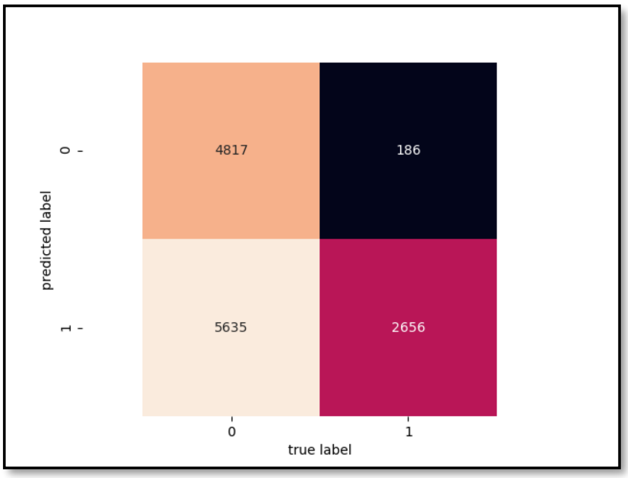

Figure 2.1: Confusion matrix for Naïve Bayes classifier.

The second model is Logistic Regression, which is classical for this type of problem. It uses a sigmoid-shaped activation function to produce binary output, either zero or one. The model is as follows:

$$\log\left(\frac{P(y\mid x)}{1-P(y\mid x)}\right)=w_0+w_1x_1+w_2x_2+\ldots+w_Nx_N\,, \qquad (2.2)$$

which I choose to solve with Stochastic Average Gradient descent (SAG), since it is faster for large datasets [Scikit-learn, 1.1.11]. Its data generating process assumes a certain fitting flexibility within the boundaries of linearity. The model is rather efficient and less likely to overfit due its linearity property. However, logistic regression is sensitive to outliers and subject to the curse of dimensionality, when high dimensional representation adversely affects its performance. Consequently, the accuracy score is 86.45% and the AUROC score is 78.18%. Its confusion matrix is summarized in Figure 2.2 below.

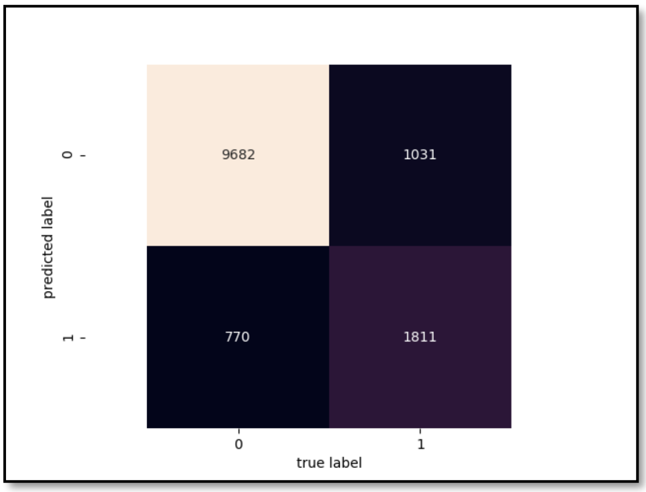

Figure 2.2: Confusion matrix for Logistic Regression classifier.

The next model is K Nearest Neighbors (KNN) classifier. It is a nonparametric method, using only the distances and closest neighbors to decide on prediction [Scikit-learn, 1.6.2]. It does not make any assumption about the data generating process. This allows the algorithm to fit flexible models, but leads to overfitting with small number of neighbors or insufficient amount of observations. Moreover, the KNN classifier is sensitive to outliers and subject to the curse of dimensionality. Subsequently, the number of neighbors is chosen to be 50, although more neighbors would improve the result; however, the change is not so significant, as tests involving different values for K, including 10 and 100, demonstrate. The observed accuracy score is 87.33% and AUROC is 79.6%. Figure 2.3 demonstrates the confusion matrix.

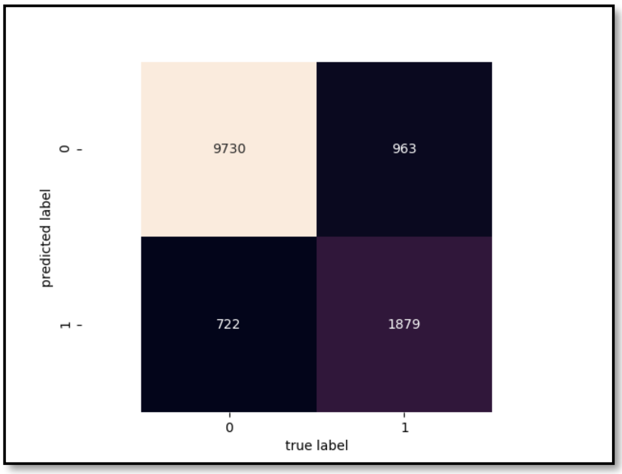

Figure 2.3: Confusion matrix for KNN classifier.

The fourth model is Random Forest classifier, which is an ensemble of decision trees. A decision tree left alone is a very weak nonparametric model; however, as a combination of many trees using the boosting aggregation, it is a powerful tool. The algorithm takes the average of probabilistic prediction made by each decision tree in a sample with replacement to find the classes [Scikit-learn, 1.11.2.1]. Being a nonparametric model, the Random Forest classifier has no assumption about the data generating process, making the model flexible in adapting nonlinear decision boundaries. It is scalable and robust to outliers due to its hierarchical structure. Although individual decision trees tend to overfit, the ensemble of trees employed resolves this problem. Thus, the number of estimators is chosen as 50 and not more because the improvement is insignificant and time consuming, as tests involving various numbers of estimators, including 10, 100, and 200, demonstrate. The accuracy and AUROC scores are 86.93% and 78.43%, respectively. The confusion matrix is given in Figure 2.4 below.

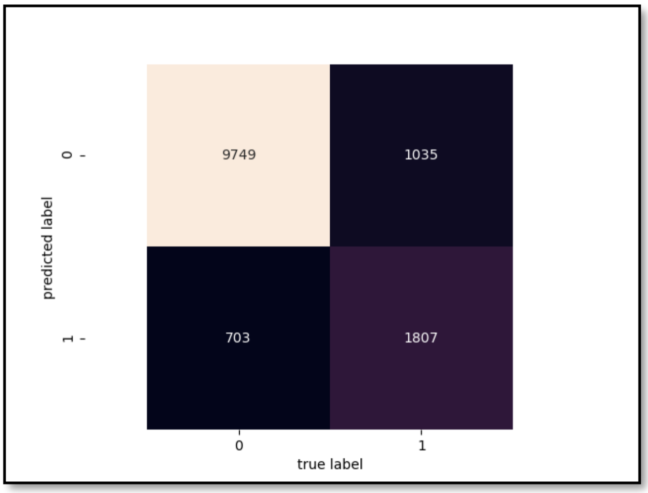

Figure 2.4: Confusion matrix for Random Forest classifier.

The fifth model is Support Vector Machines (SVM) classifier. It is a strong model, based on several optimizations, which require some time for computations. The algorithm involves primal and dual optimization problems with the following specifications:

$$* \, Primal:$$

$$\min_{w,b,\varsigma} \frac{1}{2} w^T w + C \sum_{i=1}^{n} \varsigma_i \tag{2.3}$$

$$\text{subject to} \quad y_i(w^T\phi(x_i)+b) \geq 1-\varsigma_i,$$
$$\varsigma_i \geq 0, \qquad i=1,\ldots,n$$

$$*\ Dual:$$
$$\min_{\alpha} \frac{1}{2}\alpha^T Q\alpha - e^T\alpha$$
$$\text{subject to} \quad y^T\alpha = 0, \qquad 0 \leq \alpha_i \leq C, \qquad i=1,\ldots,n$$

$$[\, Q_{ij} = y_i y_j K(x_i,x_j) \ \ \& \ \ K(x_i,x_j) = \phi(x_i)^T\phi(x_j)\,],$$

where e is a matrix of ones and $K(x_i,x_j)$ is a kernel function, and unlike other models, the classes are predicted by the sign of the decision function [Scikit-learn, 1.4.1]. The data generating process of the SVM classifier assumes an effective generalization with regularization parameters that prevent the model from underfitting and overfitting. The kernel function employed allows it to capture nonlinear decision boundaries. The model is sensitive to outliers, but robust against overfitting, scaling well to high dimensional data. Here, the value for C is chosen as 10 through available grid search cross validation function. A wide range of alternative values for C is tested, including adjacent candidates 9 and 11, which are outperformed. The resultant accuracy score is 86.70% and the AUROC is 77.87%. Its confusion matrix is summarized in Figure 2.5.

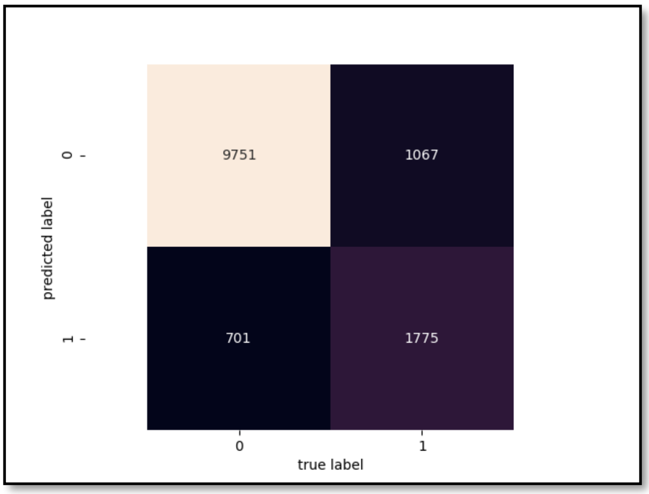

Figure 2.5: Confusion matrix for SVM classifier.

The following model is Neural Networks classifier. It is an interesting model which works as a black box but it also takes some time to compute the weights for each layer to produce the result. The multilayer perceptron trained with backpropagation is used in this algorithm [Scikit-learn, 1.17.2]. Predicting as a black box, the Neural Networks classifier does not have any assumptions about the data generating process. For the same reason, it may accidentally underfit or overfit. The model performs very well in practice, being computationally intensive. There is no specific rule for determining the structure of potential mapped networks, so the hyperparameters can be tuned by trial and error. The model is robust against outliers, if their proportion in the data is small enough. Hence, the parameters are tuned manually, choosing the two layer representation with 10 nodes each. Alternative parameters have lower scores. The accuracy score and AUROC are 87.21% and 78.97%, accordingly. The confusion matrix can be observed in Figure 2.6 below.

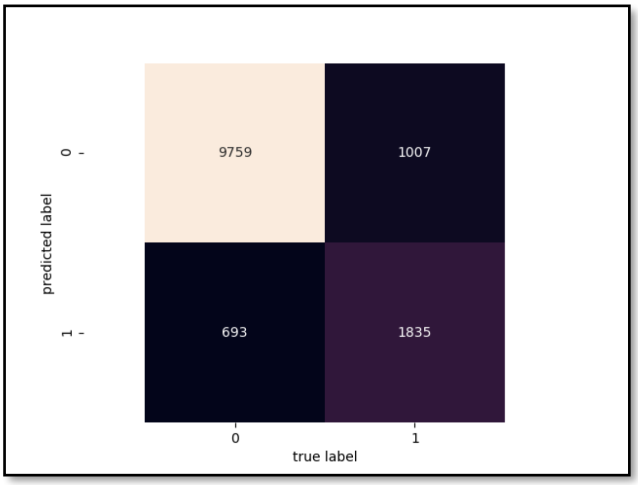

Figure 2.6: Confusion matrix for Neural Networks classifier.

The last model is Extreme Gradient Booster (XGBoost) classifier. It is also an ensemble learning model with boosting and regularization [Chen & Guestrin, 2016]. It has been the winner in many programming contests. Being nonparametric, XGBoost classifier has no specific assumption about the data generating process. It is flexible in capturing complex patterns in the data. Like other tree-based models, it does not perform very well with tasks involving extrapolation. Also, the regularization employed resolves the problem of overfitting. However, it is more prone to overfit, than random forests, which is rewarded with higher accuracy in a robust setup. The model is sensitive to outliers due to its boosting procedure. The number of estimators is tuned as 13 with the grid search cross validation function. Various numbers of estimators are tested, including adjacent candidates 12 and 14, which have worse results. Its accuracy score is 87.73% and AUROC score is 81.07%. Its confusion matrix is given in Figure 2.7.

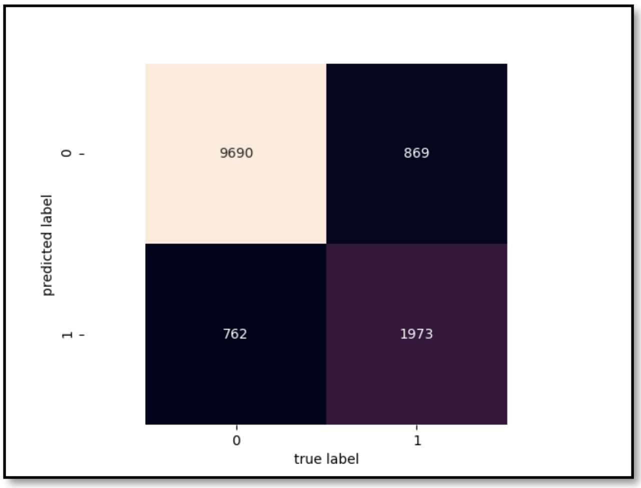

Figure 2.7: Confusion matrix for XGBoost classifier.

To conclude, among all the used models, both the highest accuracy and the highest area under the ROC curve (AUROC) belong to XGBoost classifier, although it is slightly better than its alternatives. To summarize the comparison even better, the ROC curve is demonstrated in Figure 2.8 below.

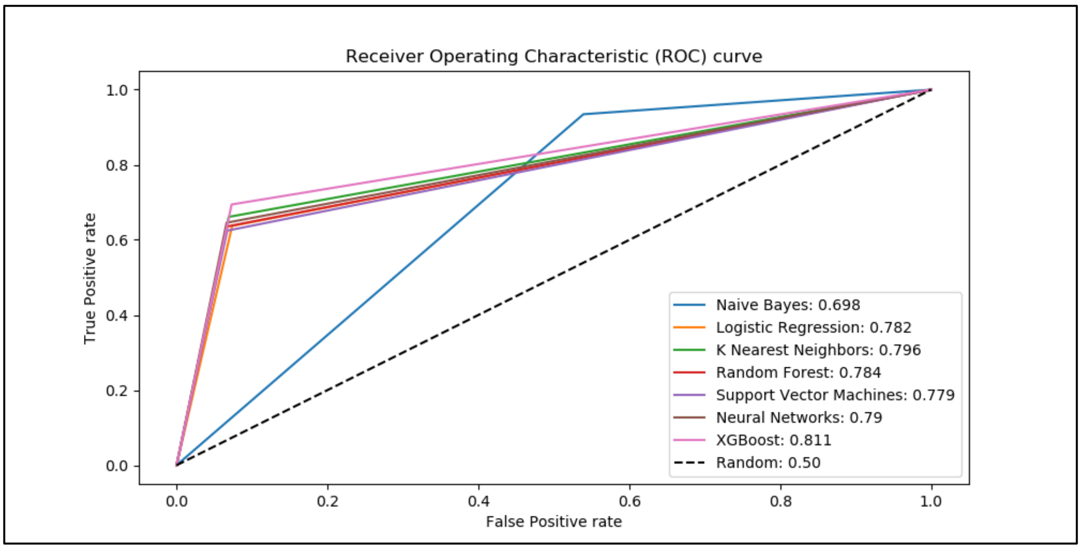

Figure 2.8: ROC curve for binary classification models of churn estimation.

As seen in the graph, there are two peaks where ROC curves concentrate. On the left, the dominant model is XGBoost, followed by KNN, which is followed by the Neural

Networks model. The peak on the right represents the Naïve Bayes classifier, which is best in predicting the underrepresented group of exited clients, although its accuracy score in total is the lowest among the models used. All other accuracy scores are greater than 85%, which is far better than guessing all as retained clients (79.1% of accuracy) and this is a significant improvement. There are not enough comparable code submissions for this dataset on Kaggle, the present ones have lower accuracy and AUROC scores than the applied XGBoost classifier.

Chapter 3:
HOUSING PRICE PREDICTION

Predicting various prices is another ubiquitous application of statistics and data science in the private sector, since knowing future prices is quite essential for profitability. One of the most famous business cases is to forecast housing prices for potential valuation and investment purposes. Other than in the housing industry itself, real estate investment can be made by miscellaneous companies and institutions as a way to invest their funds. Being considered by managers and investors, real estate is a worthy asset group, which is an ideal hedge against risks and uncertainty [Pinnington, 2020]. In general, housing price prediction constitutes a specific machine learning model of regression with a target variable of the real estate price and independent variables of the related parameters to a housing unit.

The data consist of apartment prices in Moscow, provided by Higher School of Economics and shared on Kaggle in 2018[2]. The dataset has 2040 entries with apartment prices and few related variables. The prices are reported in thousands of US dollars. The independent variables comprise the total space of apartment, the living and kitchen spaces, the distances to the city center and nearest metro station, the categorization of apartments from 1 to 8 depending on observations of data collectors, and several binary variables for the proximity to metro station (within walking distance vs. requiring transportation), the type of building material (brick vs. monolith houses), and the floor indicator (first and last floor vs. the rest). The variables also include the numbering id, which is dropped before modeling since it is useless in predictions.

As a next step before the application of different machine learning algorithms, the data are normalized using the min-max scaler, normalizing by only keeping the relative distances between the data points, since normalization allows some algorithms to approach the solution more efficiently and faster among other benefits. Similarly, after deciding the best hyperparameters for each model in a grid search of 5-fold cross validation applied to the training set, minimizing the Root Mean Square Error (RMSE), as a holdout 20% of the data are used for model testing and the other 80% are for model training purposes. The Root Mean Square Error (RMSE) is a popular metric for regression in the literature [Chai & Draxler, 2014]. The bar plots of the RMSE and the correlation

[2] Can be retrieved from www.kaggle.com/hugoncosta/price-of-flats-in-moscow

coefficient between predicted and actual prices for all models are summarized at the end to compare and decide on the best algorithm.

Given the observed dataset, the assumed data generating process is expected to produce a positive integer from a set of housing parameters and dummy variables in order to predict the apartment prices, with a continuous nature of prediction.

The first model is Linear Regression, which is a traditional technique for regression problems. It uses the Ordinary Least Squares (OLS) method to minimize the sum of squared errors, i.e. differences between the actual and predicted target variables. The model is as follows:

$$\hat{y}(w, x) = w_0 + w_1 x_1 + w_2 x_2 + \ldots + w_N x_N \, , \qquad (3.1)$$

where $\hat{y}$ is the predicted target value, w_0 is the intercept term, and $w = (w_1, w_2, \ldots w_N)$ is the vector of coefficients [Scikit-learn, 1.1.1]. The data generating process of linear regression straightforwardly assumes linearity as a functional form. The model is sensitive to outliers, so overfitting is possible. Performing poorly in the case of more complex and nonlinear distributions, it is likely to underfit as well. But it is simple and does not suffer from the curse of dimensionality, being a baseline model for regression tasks. So, the RMSE error is 29.41 and the correlation coefficient is 82.29%.

The second model is the Gaussian Process regressor. This method utilizes Gaussian probabilistic processes for regression purposes. In this algorithm, the prior parameters and the noise level, alpha parameter, are the input for the optimization framework. The prior mean is expected to be constant and the prior covariance depends on a kernel to be chosen, such as basic kernel, radial-basis function (RBF) kernel, rational quadratic kernel, and dot-product kernel [Scikit-learn, 1.7.1]. As a result of manual tuning, the combination of rational quadratic kernel and alpha parameter of 1.7 minimizes the error term. This is the rational quadratic kernel:

$$K(x_i, x_j) = (1 + \frac{d(x_i, x_j)^2}{2\alpha l^2})^{-\alpha}, \qquad (3.2)$$

where *l>0* and *α>0* are length-scale and scale mixture parameters, respectively, and *d( )* is a Euclidean distance function [Scikit-learn, 1.7.5]. As a nonparametric approach, the Gaussian Process regressor does not have a certain distribution assumed for its data generating process. Although it is computationally expensive, the model generalizes powerfully beyond the training data, depending on the chosen kernel, iteratively capturing the uncertainty very well. Thus, the algorithm is flexible in adapting nonlinear forms, but also has the tendency to overfit. It is also scalable, but sensitive to outliers. Consecutively, the calculated RMSE is 27.77 and correlation is 84.45%.

The third model is K Nearest Neighbors regressor. Such nonparametric methods take the average of the closest neighbors by analyzing the distances [Scikit-learn, 1.6.3]. Although usually uniform weights are applied for the average, in this task the inverse distance is used to allow closer neighbors to contribute more to the weights, slightly improving the algorithm performance. Being nonparametric, the KNN regressor has no specific assumption regarding the data generating process. Such flexibility can suit well the nonlinear forms. The models with small amount of neighbors or few observations may lead to overfitting. Furthermore, the KNN regressor is sensitive to outliers and subject to the curse of dimensionality. Here, the number of neighbors is chosen to be 8, since it minimizes the RMSE score. Other potential numbers of neighbors above and below 8 are also tested, but have poorer results. Although the results are not very promising, the observed RMSE is 29.75 and the correlation coefficient is 82.33%.

The following model is Random Forest regressor, which is an ensemble of decision trees, predicting the target by taking the average of each leaf node. A combination of numerous decision trees using the boosting aggregation technique converts a single decision tree, a weak nonparametric model, into a powerful regressor. In a sample with replacement, the algorithm takes the average of target predictions made by each decision tree to make the final prediction [Scikit-learn, 1.11.2.1]. There is no specific distribution assumed for the data generating process of the Random Forest regressor. Its hierarchical structure allows to model nonlinear relationships, being scalable and robust to outliers. The ensemble of trees does not let the model to overfit. However, it is not as successful as Random Forest classifier, since the nature of prediction is not continuous. Thus, the number of estimators is chosen to be 115, since it minimizes the error term. A wide range of alternative numbers is also tested, including adjacent candidates 114 and 116, which are outperformed. The RMSE score and correlation are 26.18 and 86.42%, accordingly.

The fifth model is Support Vector Machines regression. Basing on framework of computed optimizations, the model involves primal and dual optimization steps with such specifications:

$$* \ Primal:$$

$$\min_{w,b,\varsigma,\varsigma^*} \frac{1}{2} w^T w + C \sum_{i=1}^{n} (\varsigma_i + \varsigma_i^*)$$

$$subject\ to \quad y_i - w^T \phi(x_i) - b \leq \varepsilon + \varsigma_i,$$

$$w^T \phi(x_i) + b - y_i \leq \varepsilon + \varsigma_i^*,$$

$$\varsigma_i, \varsigma_i^* \geq 0, \qquad i = 1, \ldots, n$$

(3.3)

$$* \ Dual:$$

$$\min_{\alpha,\alpha^*} \frac{1}{2} (\alpha - \alpha^*)^T Q (\alpha - \alpha^*) + \varepsilon e^T (\alpha + \alpha^*) - y^T (\alpha - \alpha^*)$$

$$subject\ to \quad e^T (\alpha - \alpha^*) = 0,$$

$$0 \leq \alpha_i, \alpha_i^* \leq C, \qquad i = 1, \ldots, n$$

$$[\, Q_{ij} = K\left(x_i, x_j\right) = \phi(x_i)^T \phi\left(x_j\right) \,],$$

where e is a matrix of ones and $K\left(x_i, x_j\right)$ is a kernel function [Scikit-learn, 1.4.2]. The data generating process of the SVM regressor forms an effective generalization along with regularization parameters. Its strength is in kernel trick, making the algorithm flexible in modeling nonlinear forms. The SVM regressor is robust to overfitting, especially in high dimensional spaces. It is also sensitive to outliers. Consequently, the C coefficient is chosen to be 10000 within the grid search cross validation function; greater values of C do not improve the performance significantly, as tests involving a wide range of candidates show. The resultant RMSE score is 25.29 and the correlation coefficient is 87.25%.

The next model is Neural Networks regressor. Being a black box model, it can achieve great results without detailed interpretations. Here, the multilayer perceptron is trained using backpropagation with no activation function at the output level to have the regression format in predictions [Scikit-learn, 1.17.3]. Working as a black box, the Neural Networks regressor does not make any assumptions regarding the data generating process. Both underfitting and overfitting are possible within such framework, which can be overcome with an effective tuning of hyperparameters. Even though it requires certain hardware resources, the model is rather flexible and able to capture extremely complex forms. It is not affected by outliers, if their amount is small. The parameters are tuned manually by trial and error, choosing the neural network of five layers with 150 nodes each. Other potential parameters have higher error terms. The RMSE error and correlation are 25.74 and 86.88%, respectively.

The last model is Extreme Gradient Booster (XGBoost) regressor, a popular technique. It is a powerful ensemble algorithm with regularized boosting of regression trees [Chen & Guestrin, 2016]. More as a black box, XGBoost regressor has no assumption about the data generating process. Such flexibility allows to model nonlinear relationships in the data. The problem of overfitting is dealt with ensembles and regularization. Even though, its tree-based structure leads to difficulties with extrapolation and the boosting process causes sensitivity to outliers, the model is scalable and quite successful in practice. So, the number of estimators is tuned as 21 with the help of grid search cross validation function. Other candidates being tested, including adjacent values of 20, 22, and 25, are outperformed. Its RMSE score is 26.09 and correlation coefficient is 86.30%. The comparison of models can be clearly seen within the bar plots summarized in Figures 3.1-3.2 below.

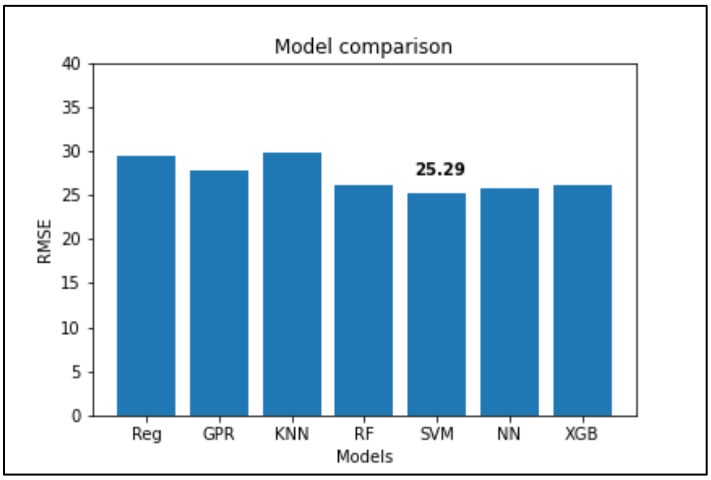

Figure 3.1: RMSE scores across regression models of housing price prediction.

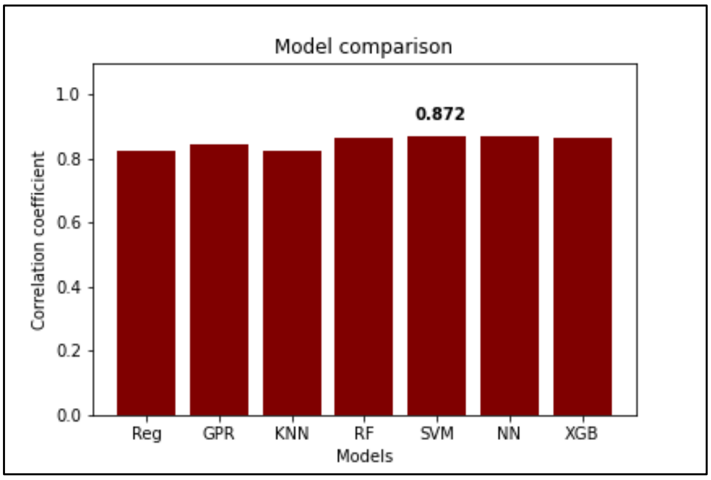

Figure 3.2: Correlations across regression models of housing price prediction.

In sum, among all the applied algorithmic models, both the lowest error term (RMSE score) and the best linear relationship between predictions and actual prices (correlation coefficient) belong to SVM regressor, although it is slightly better than its alternatives, such as Neural Networks, XGBoost, and Random Forest models. Furthermore, a powerful model, the Gaussian Process regressor may need a larger dataset to have a better performance. Apparently, the last models Linear Regression and KNN, being simple and straightforward, are observed to have the worst results. There are insufficient comparable code submissions for this dataset on Kaggle, the present ones have higher error and lower correlation measures than the applied SVM regressor.

Chapter 4:
SENTIMENT ANALYSIS

Another notable application of machine learning methodology in business is sentiment analysis technique, also known as opinion mining. It is a type of natural language processing (NLP) algorithms, commonly constituting text analysis. Given a piece of writing, usually voice of the customer material such as a comment or a review, the algorithm aims to understand its overall tone and classify it according to some acceptable measure. For instance, an opinion could be classified as positive, negative, or neutral. Hence, it is usually a classification task with various number of targeted classes and preprocessed text as an input, obtained through the feature extraction process. In practice, the sentiment analysis is deployed in a range of aspects in business, such as customer support and service, market research, public relations, and human resources management in numerous industries [Puschmann & Powell, 2018].

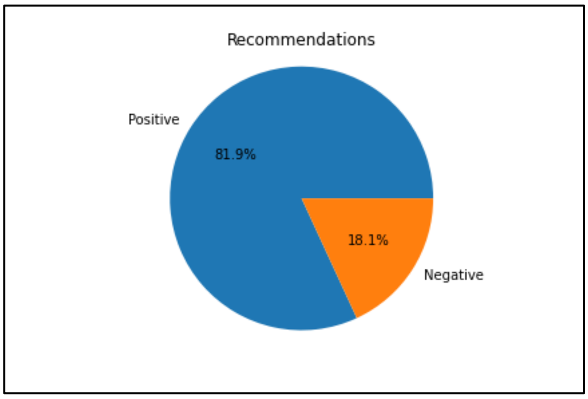

Figure 4.1: Distribution of customer recommendations.

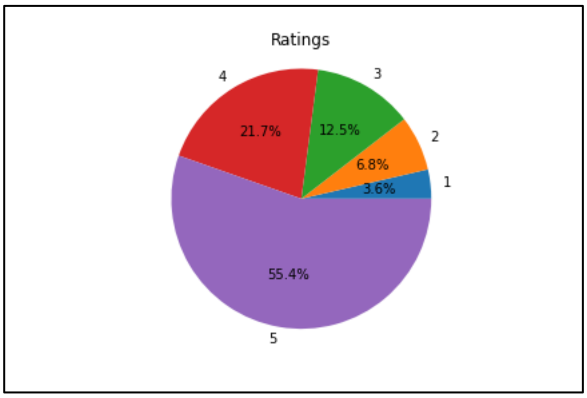

Figure 4.2: Distribution of customer ratings.

The dataset is the collection of women's e-commerce clothing reviews from a real but anonymous source, shared on Kaggle in 2018[3]. The data initially consisted of 23486 non-null entries depicting the customer reviews with related information, including product and customer descriptions. Other columns than text reviews, customer recommendations and rankings were dropped to concentrate on a sole sentiment analysis task. The text reviews column is the main variable for the learning model. The two others are possible classification targets with the customer recommendations, being either yes or no in a binary representation, and the rankings on a scale from 1 to 5. Since text reviews were omitted in some entries, they were dropped, resulting in a total of 22641 rows. The distribution for recommendations and rankings are represented as pie charts in Figures 4.1-4.2 above. Only 18.1% of all reviews have negative recommendations. Similarly, the majority of reviews have maximum ranking with 55.4% of proportion. In a consistent manner, each of next ranking level has lesser proportion than the previous. Since there are more classes, a multiclass classification task with a target of rankings is expected to be less successful in predictions than a binary classification task with a target of recommendations, in a general algorithmic setup.

First of all, the text reviews have to be preprocessed. Thus, the text reviews are filtered from numbers and special characters, since most of them are irrelevant and do not possess any important information. The cases of remaining alphabetical characters are lowered to enforce uniformity. Taking out the stopwords from the current data, the tag clouds are produced for both positive and negative recommendation labels. Both clouds are demonstrated in Figures 4.3-4.4 below.

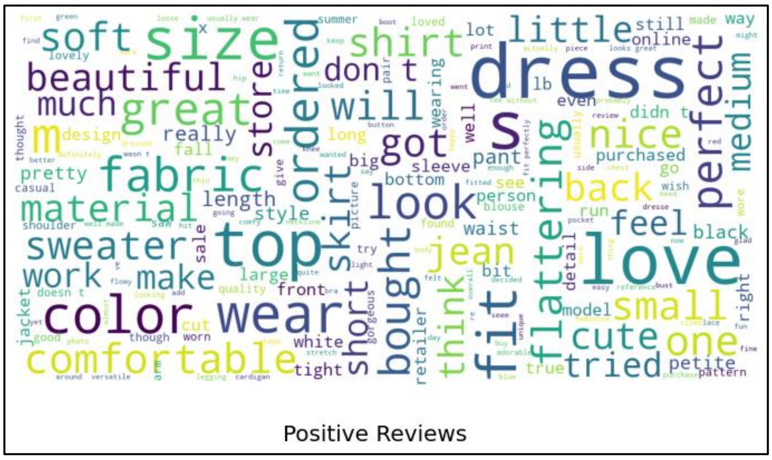

Figure 4.3: Tag cloud of text reviews with positive recommendation.

[3] Can be retrieved from www.kaggle.com/nicapotato/womens-ecommerce-clothing-reviews

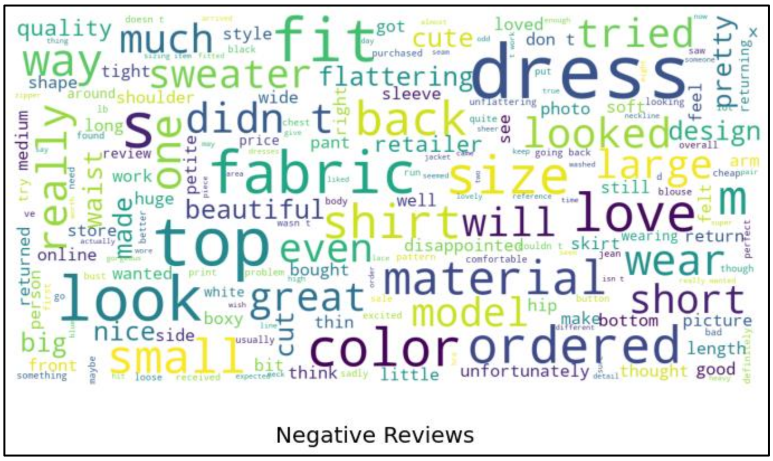

Figure 4.4: Tag cloud of text reviews with negative recommendation.

The stopwords, such as prepositions and conjunctions, are omitted because they are usually meaningless in such an analysis. In both tag clouds, although there are many common words related to the e-commerce of women's clothing in general, some words related to either positive or negative reaction can also be observed. As a next step, the text reviews with omitted stopwords are tokenized into bag-of-words (BoW) representation, which is a simplified matrix with the word counts. Then, the BoW representation is converted to term frequency – inverse document frequency (TF-IDF) measure, which is quite popular for text analysis. This measure reflects the relevance of words to a document in a collection of documents, i.e. corpus [Qaiser & Ali, 2018]. The unigram version of BoW and TF-IDF measures are compared, where unigram is a single word representation from the original text. Alternatively, their bigram version, which is a representation as pairs of consecutive words, being also considered, is not included in the analysis due to low effectiveness in comparison.

The same list of classification models, described in chapter 1, is utilized for binary classifications of sentiment analysis. The most successful ones are also used for the multiclass classification task. For the sake of simplicity, the latter uses standard independent classification with nominal nature of predictions, although it does not take into consideration the proximity of the predicted class to the actual one. Since accuracy metric is not sufficient for both cases, it is necessary to use a more reliable measure in addition. As before, AUROC measure is employed for the binary classification. For the multiclass classification, however, there is no well-designed ROC curve for this case. One common measure is Macro F1 score, a macro averaged F1 measure of each class, which combines both popular precision and recall measures at once [Grandini, Bagli, & Visani, 2020]. The precision measures how good the model is in predicting a particular class and the recall measures how well the particular class is predicted overall [Grandini, Bagli, & Visani, 2020]. The F1 score, which is a tradeoff of both, geometrically averaged for each class, is a good substitute for the AUROC measure. The macro average, rather than the weighted one, is used to deem each class equally; otherwise, guessing the most probable class at random would increase the score, as it does for accuracy measure. As before, having decided the best hyperparameters for each model, 80% of the data are employed for model training and the remaining 20% are for model testing purposes. The grid search of 5-fold cross validation applied to the training set is used to decide for the best model parameters, where it is required.

The assumed data generating process for the sentiment analysis is expected to translate the text input in the form of processed matrices into outputs of either binary classification being 0 or 1 for customer recommendations, or multiclass classification on a discrete scale from 1 to 5 for customer rankings. Since the same classification models from previous chapters are reintroduced for the sentiment analysis, the same assumptions regarding their data generating processes are made.

All seven models are applied for binary classification task of sentiment analysis. Firstly, the Naïve Bayes classifier [Scikit-learn, 1.9.1] and Logistic regression [Scikit-learn, 1.1.11] models do not have any decision variables, as before. The number of neighbors is chosen to be 50 for K Nearest Neighbors classifier [Scikit-learn, 1.6.2]. Conducted tests involve different values for K, including 40 and 100, demonstrating no significant improvement for values higher than 50. Moreover, the number of estimators for Random Forest classifier [Scikit-learn, 1.11.2.1] is chosen as 100; more estimators do not improve the results significantly, as tests involving lower and higher amounts of estimators depict. Then, the Support Vector Machines classifier [Scikit-learn, 1.4.1] maximizes the AUROC score at the C value of 100 for the BoW version and C of 10000 for the TF-IDF one. A wide range of candidate values for C is tested in both directions, showing no superiority over the chosen parameters. Being tuned manually, the Neural Networks classifier [Scikit-learn, 1.17.2] is decided to have six layer representation with 5 nodes each for BoW and four layers with 7 nodes per each for TF-IDF. Lastly, the Extreme Gradient Booster classifier [Chen & Guestrin, 2016] is chosen to have 1000 estimators for both cases. Other candidates, such as 900 and 3000, have lower scores.

Table 4.1: Comparison of binary classification models of sentiment analysis.

Model	BoW		TF-IDF	
	accuracy	AUROC	accuracy	AUROC
Naïve Bayes	45.9%	55.15%	45.93%	55.03%
Logistic Regression	88.54%	78.29%	88.23%	74.38%
K Nearest Neighbors	80.95%	51.49%	82.69%	56.13%
Random Forest	82.65%	55.66%	82.73%	55.89%
Support Vector Machines	87.99%	74.88%	*88.7%*	77.87%
Neural Networks	87.46%	**82.48%**	85.45%	**80.8%**
XGBoost	*88.87%*	77.72%	87.88%	77.33%

The outputs of the binary classification modeling are summarized in Table 4.1 above. As seen in the table, the outputs for BoW and TF-IDF input versions are more or less similar, especially for the accuracy scores. In the case of Logistic Regression and Neural Networks, the AUROC score is significantly better for BoW. On the other hand, KNN and SVM models perform better for TF-IDF. The AUROC scores, in general, demonstrate the relative weakness of simpler models of Naïve Bayes and KNN. For some reason, a more complex model Random Forest classifier also performs poorly. The ROC curves for both BoW and TF-IDF versions are demonstrated as well in Figures 4.5-4.6.

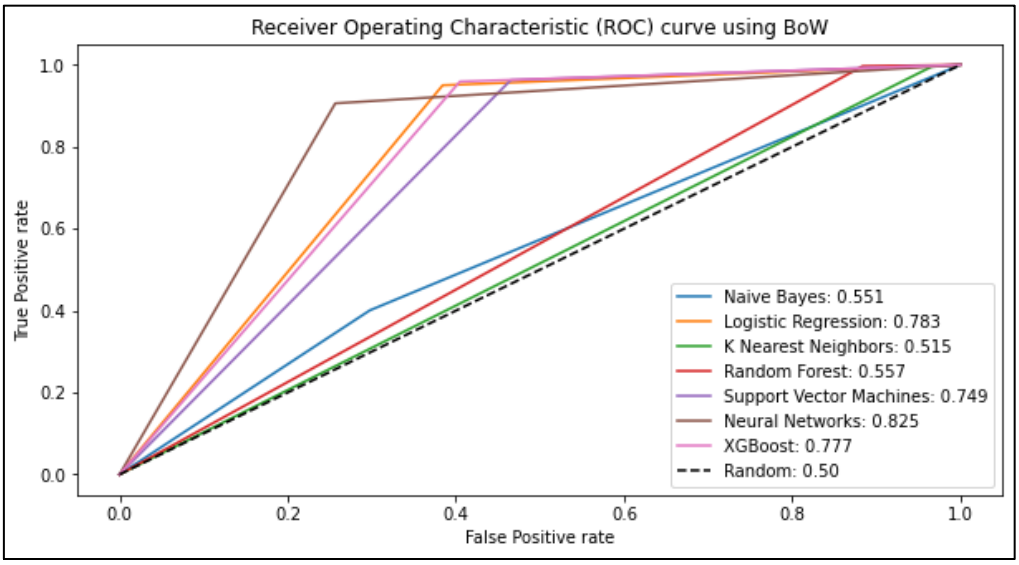

Figure 4.5: ROC curve for binary classification models using BoW.

As a result, despite the indication of higher accuracy scores for some models, which is less important, according to AUROC measures, the best performing model for both BoW and TF-IDF inputs is Neural Network classifier.

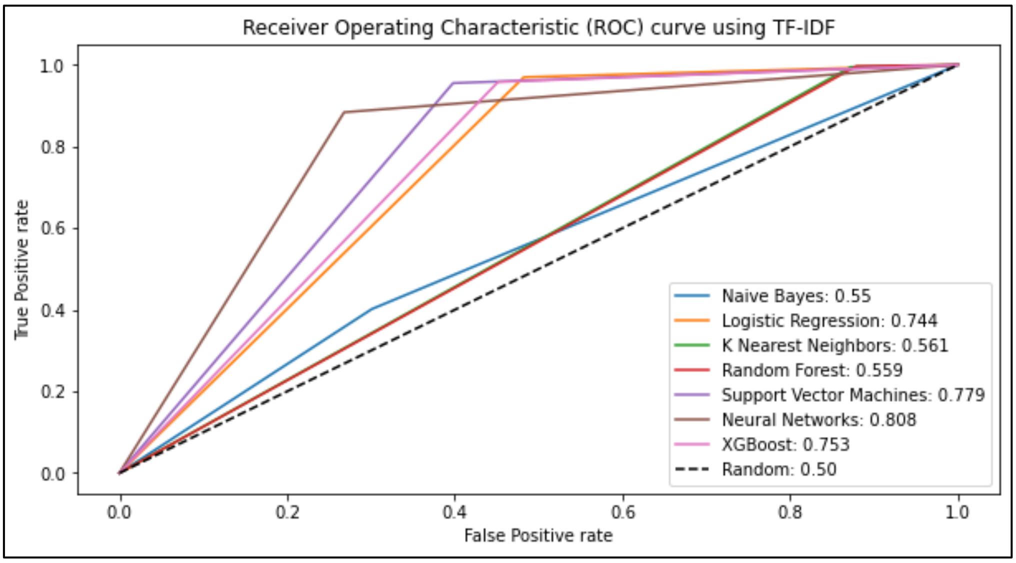

Figure 4.6: ROC curve for binary classification models using TF-IDF.

The classifiers such as Naïve Bayes, Random Forest, and K Nearest Neighbors, show poor performance for this type of problem with AUROC score closer to random guessing and, therefore, are omitted in the multiclass classification case. There are many comparable Kaggle submissions for this dataset performing a similar task, some of which use more complex methods including RNN (Recurrent Neural Network), which outperform relatively simpler approaches of this research.

Then, four best models are applied for multiclass classification task of sentiment analysis. The Logistic regression [Scikit-learn, 1.1.11] classifier is used without any decision variable. Secondly, the SVM classifier [Scikit-learn, 1.4.1] maximizes the Macro F1 score at C value of 1000 for the BoW case and C of 10000 for TF-IDF, as tests with various candidates demonstrate. With a manual tuning, the Neural Networks classifier [Scikit-learn, 1.17.2] is chosen to have one layer representation with 100 nodes for both cases. Lastly, the XGBoost classifier [Chen & Guestrin, 2016] is decided to have 150 estimators for both BoW and TF-IDF input versions, as a result of conducted tests that eliminated outperformed candidates. Table 4.2 summarizes the resultant accuracy and macro averaged F1 scores.

Table 4.2: Comparison of multiclass classification models of sentiment analysis.

Model	BoW		TF-IDF	
	accuracy	Macro F1 score	accuracy	Macro F1 score
Logistic Regression	61.54%	0.42	*63.66%*	0.39
Support Vector Machines	**62.4%**	**0.42**	**63.52%**	**0.42**
Neural Networks	58.8%	0.40	57.52%	0.39
XGBoost	62.35%	0.40	61.87%	0.38

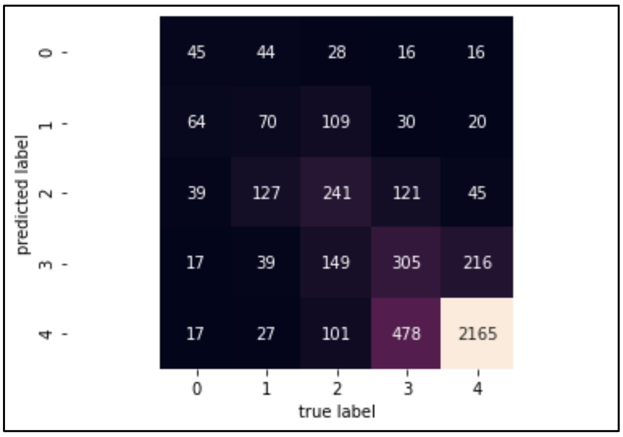

Figure 4.7: Confusion matrix for SVM classifier using BoW.

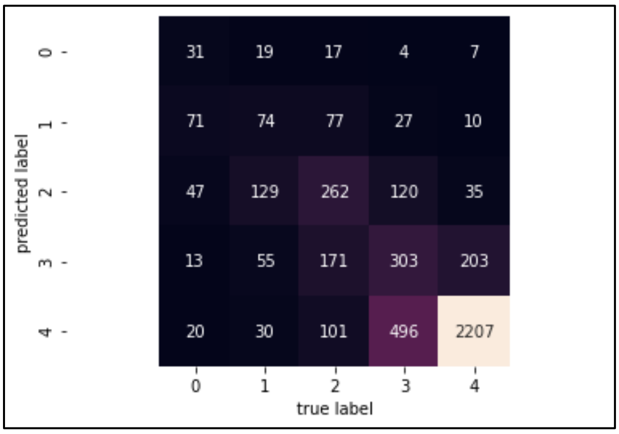

Figure 4.8: Confusion matrix for SVM classifier using TF-IDF.

As expected, the resultant scores are much lower than in the binary classification case, since there are five different classes. With a priority given to Macro F1 score, the best performing model here is Support Vector Machines classifier. Its confusion matrices for two versions are depicted in Figures 4.7-4.8 above. In general, the BoW input representation has slightly better performance for multiclass classification, although in the case of SVM classifier, both BoW and TF-IDF have equal macro averaged F1 scores. Thus, the sentiment analysis problem with classifications of different forms has multiple aspects, showing a range of machine learning tools and methods.

Chapter 5:
CONCLUSION

To conclude, the machine learning methodology of three distinct business applications is analyzed and compared in the paper. The first chapter applies the binary classification models for the churn estimation task. The second chapter deploys the regression analysis techniques to predict the housing prices. The third chapter incorporates the sentiment analysis tools along with models of binary and multiclass classification for e-commerce reviews and ratings. Hence, the diverse range of machine learning and data science methodology is very beneficial to be applied in business.

Theoretically, given datasets with some data generating process in the form of cross-sectional data from telecom, housing, and e-commerce industries with respective distributions and statistical assumptions, all the models employed for classification and regression tasks have different data generating processes depending on their mathematical formulations or nonparametric forms with various a priori expectations. The heterogeneity of data in the aggregate may lead to nonlinear deviant expectations, causing unwilling deviations. Moreover, other than for preprocessing purposes, the knowledge of the data is not used for modeling decisions, relying more on the general practical approaches of model tuning instead. There are also possible outliers in the data, which are partly resolved with data normalization as a first step. While the main goal of a machine learning model is to generalize from the training data to any data of the same problem type, two unwilling scenarios may occur in the modeling: underfitting when a model is not trained well enough and overfitting when a model is trained too well that it learns the noise of the data, which is even worse. To ensure the robustness against overfitting, outliers, and deviations of the underlying assumptions, the cross validation technique is applied and the holdout (testing) set is allocated. Being trained with a different data split, the model is being tested on a holdout, as if it is unknown prior to modeling to imitate real world prediction problem. Then, the common metrics are compared for each task as the main measurement of goodness of fit, since the practical aspect is the main goal of this research. In some cases the more complex models are outperformed by simpler ones, which is common, especially if the model is too complex that it leads to overfitting.

Even though the lack of a solid theoretical basis and a certain dependence of results on the specific datasets employed are possible drawbacks of this research, it presents a practical application of common machine learning algorithms, demonstrating their relative strengths and weaknesses in particular types of problem. Optimizing the feature selection method and adding more complex and recent techniques would potentially improve the results. Future works in this direction are very promising and inspiring.

BIBLIOGRAPHY

[Chai & Draxler, 2014] Chai, T., & Draxler, R. R. (2014). Root mean square error (RMSE) or mean absolute error (MAE)? *Geoscientific Model Development, 7*(3), 1247-1250. https://doi.org/10.5194/gmdd-7-1525-2014

[Chen & Guestrin, 2016] Chen, T., & Guestrin, C. (2016). XGBoost: A Scalable Tree Boosting System. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 785–794). New York, NY, USA: ACM. https://doi.org/10.1145/2939672.2939785

[Grandini, Bagli, & Visani, 2020] Grandini, M., Bagli, E., & Visani, G. (2020). Metrics for Multi-Class Classification: an Overview. *ArXiv*, abs/2008.05756. Retrieved from https://arxiv.org/pdf/2008.05756.pdf

[Jeong, 2018] Jeong, T. (2018). Adding Value With Artificial Intelligence. In *Building A Smart Partnership For The Fourth Industrial Revolution* (pp. 15-21). Atlantic Council. Retrieved from www.jstor.org/stable/resrep20947.5

[Kraus, Feuerriegel, & Oztekin, 2020] Kraus, M., Feuerriegel, S., & Oztekin, A. (2020). Deep learning in business analytics and operations research: Models, applications and managerial implications. *European Journal of Operational Research, 281*(3), 628-641. https://doi.org/10.1016/j.ejor.2019.09.018

[Manďák & Hančlová, 2019] Manďák, J., & Hančlová, J. (2019). Use of Logistic Regression for Understanding and Prediction of Customer Churn in Telecommunications. *Statistika: Statistics and Economy Journal, 99*(2), 129-141. Retrieved from https://www.czso.cz/documents/10180/88506448/32019719q2_129_mandak_analyses.pdf

[Pedregosa et al., 2011] Pedregosa, F., Varoquaux, Ga"el, Gramfort, A., Michel, V., Thirion, B., Grisel, O., … others. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, *12*(Oct), 2825–2830. Retrieved from https://arxiv.org/abs/1201.0490

[Pinnington, 2020] Pinnington, S. (2020). *Real Estate Investing* (Rep.). Retrieved from https://iqeq.com/sites/default/files/2020-03/IQ-EQ%20Real%20Estate%20Investing%20White%20Paper.pdf

[Puschmann & Powell, 2018] Puschmann, C., & Powell, A. (2018). Turning Words Into Consumer Preferences: How Sentiment Analysis Is Framed in Research and the News Media. *Social Media + Society*. https://doi.org/10.1177/2056305118797724

[Rao & Bhattacharyya, 2019] Rao, N. T., & Bhattacharyya, D. (2019). Applications of Artificial Intelligence and ML in Business. *International Journal of Innovative Technology and Exploring Engineering (IJITEE)*, *8*(7), 2734–2738. Retrieved from https://www.ijitee.org/wp-content/uploads/papers/v8i7/G6392058719.pdf

[Scikit-learn, 1.1.1] Scikit-learn. (2020). 1.1.1. Ordinary Least Squares. Retrieved from https://scikit-learn.org/stable/modules/linear_model.html

[Scikit-learn, 1.1.11] Scikit-learn. (2020). 1.1.11. Logistic regression. Retrieved from https://scikit-learn.org/stable/modules/linear_model.html

[Scikit-learn, 1.4.1] Scikit-learn. (2020). 1.4.1. Classification. Retrieved from https://scikit-learn.org/stable/modules/svm.html

[Scikit-learn, 1.4.2] Scikit-learn. (2020). 1.4.2. Regression. Retrieved from https://scikit-learn.org/stable/modules/svm.html

[Scikit-learn, 1.6.2] Scikit-learn. (2020). 1.6.2. Nearest Neighbors Classification. Retrieved from https://scikit-learn.org/stable/modules/neighbors.html

[Scikit-learn, 1.6.3] Scikit-learn. (2020). 1.6.3. Nearest Neighbors Regression. Retrieved from https://scikit-learn.org/stable/modules/neighbors.html

[Scikit-learn, 1.7.1] Scikit-learn. (2020). 1.7.1. Gaussian Process Regression (GPR). Retrieved from https://scikit-learn.org/stable/modules/gaussian_process.html

[Scikit-learn, 1.7.5] Scikit-learn. (2020). 1.7.5. Kernels for Gaussian Processes. Retrieved from https://scikit-learn.org/stable/modules/gaussian_process.html

[Scikit-learn, 1.9.1] Scikit-learn. (2020). 1.9.1. Gaussian Naive Bayes. Retrieved from https://scikit-learn.org/stable/modules/naive_bayes.html

[Scikit-learn, 1.11.2.1] Scikit-learn. (2020). 1.11.2.1. Random Forests. Retrieved from https://scikit-learn.org/stable/modules/ensemble.html

[Scikit-learn, 1.17.2] Scikit-learn. (2020). 1.17.2. Classification. Retrieved from https://scikit-learn.org/stable/modules/neural_networks_supervised.html

[Scikit-learn, 1.17.3] Scikit-learn. (2020). 1.17.3. Regression. Retrieved from https://scikit-learn.org/stable/modules/neural_networks_supervised.html

[Siemes, 2016] Siemes, T. (2016). *Churn prediction models tested and evaluated in the Dutch indemnity industry* (Master's thesis). Open University of the Netherlands. Retrieved from https://core.ac.uk/download/pdf/80496548.pdf

[Qaiser & Ali, 2018] Qaiser, S., & Ali, R. (2018). Text Mining: Use of TF-IDF to Examine the Relevance of Words to Documents. *International Journal of Computer Applications*, *181*(1). https://www.doi.org/10.5120/ijca2018917395