Intro
General

Machine learning allows the user to feed a computer algorithm an immense amount of data and have the computer analyze and make data-driven recommendations and decisions based on only the input data. In most of the situations we want to have a machine learning system to make **predictions**, so we have several categories of machine learning tasks depending on the type of prediction needed: **Classification, Regression, Clustering, Generation**, etc.

Classification is the task whose goal is the prediction of the label of the class to which the input belongs (e.g., Classification of images in two classes: cats and dogs). **Regression** is the task whose goal is the prediction of numerical value(s) related to the input (e.g., House rent prediction, Estimated time of arrival). **Generation** is the task whose goal is the creation of something new related to the input (e.g., Text translation, Audio beat generation, Image delousing). **Clustering** is the task of grouping a set of objects in such a way that objects in the same group (called a **cluster**) are more similar (in some sense) to each other than to those in other **clusters** (e.g., Clients cluttering).

In machine learning, there are learning paradigms that relate to one aspect of the dataset: **the presence of the label to be predicted**. **Supervised Learning** is the paradigm of learning that is applied when the dataset has the label variables to be predicted, known as y variables. **Unsupervised Learning** is the paradigm of learning that is applied when the dataset has not the label variables to be predicted. **Self-supervised Learning** is the paradigm of learning that is applied when part of the X dataset is considered as the label to be predicted (e.g., the Dataset is made of texts and the model try to predict the next word of each sentence).

Notebook overview

Telco companies Churn analytics provides valuable capabilities to predict customer churn and also define the underlying reasons that drive it. The churn metric is mostly shown as the percentage of customers that cancel a product or service within a given period (mostly months). A customer churn analysis is a typical classification problem within the domain of supervised learning.
In this Article, a basic machine learning pipeline based on a sample data set from Telco company is to build performance of different model types and compare. The pipeline used for this process consists of 8 steps:
NOTE:
For more info, please check on my GitHub account: https://github.com/Kennymaur/ML-Customer Churn LP3.git

Step 1: Problem Definition
Step 2: Data Collection
Step 3: Exploratory Data Analysis (EDA)
Step 4: Feature Engineering
Step 5: Train/Test Split
Step 6: Model Evaluation Metrics Definition
Step 7: Model Selection, Training, Prediction and Assessment

Step 8: Hyperparameter Tuning/Model Improvement

Step 1: Problem Definition

Based on the introduction the key challenge is to predict if an individual customer will churn or not. To accomplish that, machine learning models are trained based on 80% of the sample data. The remaining 20% are used to apply the trained models and assess their predictive power with regards to "churn / not churn". A side question will be, which features actually drive customer churn. That information can be used to identify customer "pain points" and resolve them by providing goodies to make customers stay.

To compare models and select the best for this task, the accuracy is measured. Based on other characteristics of the data, for example the balance between classes (number of "churners" vs. "non-churners" in data set) further metrics are considered if needed.

Step 2: Data Collection

The data set for this classification problem is taken from github.

The pipeline build-up is started with imports of some basic libraries that are needed throughout the case. This includes Pandas and Numpy for data handling and processing as well as Matplotlib and Seaborn for visualization.

For this exercise, the data set (.csv format) is downloaded to a local folder, read into the Jupyter notebook and stored in a Pandas DataFrame.

Step 3: Exploratory Data Analysis

After data collection, several steps are carried out to explore the data. Goal of this step is to get an understanding of the data structure, conduct initial preprocessing, clean the data, identify patterns and inconsistencies in the data (i.e. skewness, outliers, missing values) and build and validate hypotheses.

Understanding

In the first part of EDA the data frame is evaluated for structure, columns included and data types. The goals of this step are to get a general understanding of the data set, check domain knowledge and get first ideas on topics to investigate. In this step some standard Pandas functions are used:

The unique values for every feature are printed to the console to get a deeper understanding about the feature values.

Meaning of Features

By inspecting the columns and their unique values, a general understanding about the features can be built. The features can also be clustered into different categories:

Classification labels

Churn — Whether the customer churned or not (Yes or No)

Customer services booked

PhoneService — Whether the customer has a phone service (Yes, No)

MultipleLines — Whether the customer has multiple lines (Yes, No, No phone service)

InternetService — Customer's internet service provider (DSL, Fiber optic, No)

OnlineSecurity — Whether the customer has online security (Yes, No, No internet service)
OnlineBackup — Whether the customer has online backup (Yes, No, No internet service)
DeviceProtection — Whether the customer has device protection (Yes, No, No internet service)
TechSupport — Whether the customer has tech support (Yes, No, No internet service)
StreamingTV — Whether the customer has streaming TV (Yes, No, No internet service)
StreamingMovies — Whether the customer has streaming movies (Yes, No, No internet service)
Customer account information
Tenure — Number of months the customer has stayed with the company
Contract — The contract term of the customer (Month-to-month, One year, Two year)
PaperlessBilling — Whether the customer has paperless billing (Yes, No)
PaymentMethod — The customer's payment method (Electronic check, Mailed check, Bank transfer (automatic), Credit card (automatic))
MonthlyCharges — The amount charged to the customer monthly
TotalCharges — The total amount charged to the customer

Customers demographic info
customerID — Customer ID
Gender — Whether the customer is a male or a female
SeniorCitizen — Whether the customer is a senior citizen or not (1, 0)
Partner — Whether the customer has a partner or not (Yes, No)
Dependents — Whether the customer has dependents or not (Yes, No)

Data Preprocessing for EDA
The analysis shows 11 missing values for "TotalCharges".

Hypothesis Building
Looking at the features included in data and connecting them to their potential influence on customer churn, the following hypotheses can be made:

- The longer the contract duration the less likely it is that the customer will churn as he/she is less frequently confronted with the termination/prolongation decision and potentially values contracts with reduced effort.
- Customers are willing to cancel simple contracts with few associated product components quicker and more often than complex product bundles — for bundles customers value the reduced administrative complexity. They might also be hesitant to cancel a contract, when they depend on the additional service components (e.g. security packages).
- Customers with patners might churn less to keep the services running for their family.
- Tenure, contract duration terms and number of additional services are assumed to be among the most important drivers of churn.
- More expensive contracts lead to increased churn as the chances to save money by changing providers might be higher.
- Senior citizens tend to churn less due to the extended effort associated with terminating contracts.

Data Exploration
The plot shows a class imbalance of the data between churners and non-churners. To address this, resampling would be a suitable approach. To keep this case simple, the imbalance is kept forward and specific metrics are chosen for model evaluations.

Senior citizens churn rate is much higher than non-senior churn rate.
Churn rate for month-to-month contracts much higher than that of other contract durations.
Moderately higher churn rate for customers without partners.
Payment method electronic check shows much higher churn rate than other payment methods.
Customers with Internet Service fiber optic as part of their contract have much higher churn rate.
No outliers in numerical features detected with the IQR method — no adjustments made.

Data Cleaning

Feature Engineering Actions
Based on the data types and the values, following actions are defined to preprocess/engineer the features for machine readability and further analysis:

Columns removed
customerID: not relevant

No action
SeniorCitizen
Label encoding The following features are categorical and each take on 2 values (mostly yes/no) — therefore are transformed to binary integers

gender
Partner
Dependents
Churn
PhoneService
PaperlessBilling
One-Hot Encoding The following features are categorical, yet not ordinal (no ranking) but take on more than 2 values. For each value, a new variable is created with a binary integer indicating if the value occured in a data entry or not (1 or 0).

MultipleLines
InternetService
OnlineSecurity
OnlineBackup
DeviceProtection
TechSupport
StreamingTV
StreamingMovies
Contract

PaymentMethod
Min-Max Scaling Values of numerical features are rescaled between a range of 0 and 1. Min-max scaler is the standard approach for scaling. For normally distributed features standard scaler could be used, which scales values around a mean of 0 and a standard deviation of 1. For simplicity we use min-max scaler for all numerical features.

Tenure
TotalCharges
MonthlyCharges

Step 4: Feature Engineering

In feature engineering, the steps identified at the end of EDA are executed. Additionally, a new feature is generated from existing features and a correlation analysis is conducted after all features have been transformed to numerical.

Step 5: Train-Test-Split

For conduction of model training and testing steps, the data set is splitted into 80% training data and 20% test data. The "Churn" column is defined as the class (the "y"), the remaining columns as the features (the "X").

Step 6: Model Evaluation Metrics

For performance assessment of the chosen models, various metrics are used:

Feature weights: Indicates the top features used by the model to generate the predictions
Confusion matrix: Shows a grid of true and false predictions compared to the actual values
Accuracy score: Shows the overall accuracy of the model for training set and test set
Precision-Recall-Curve: Shows the diagnostic ability by comparing false positive rate (FPR) and false negative rate (FNR) for different thresholds of class predictions. It is suitable for data sets with high class imbalances (negative values overrepresented) as it focuses on precision and recall, which are not dependent on the number of true negatives and thereby excludes the imbalance
F1 Score: Builds the harmonic mean of precision and recall and thereby measures the compromise between both.
AUC (for PRC): Measures the overall separability between classes of the model related to the Precision-Recall curve

Step 7: Model Selection, Training, Prediction and Assessment

In the beginning we will test out several models and measure their performance by several metrics. Those models will be optimized in a later step by tuning their hyperparameters. The models used include:

K Nearest Neighbors — fast, simple and instance-based
Logistic Regression — fast and linear model
Random Forest — slower but accurate ensemble model based on decision trees
ADA boost Model

Decision Tree Model

Step 8: Hyper parameter Tuning/Model Improvement

To address a potential biased stemming from the specific split of the data in the train-test-split part, cross-validation is used during hyper parameter tuning with Grid Search and Randomized Search. Cross validations split the training data into in a specified amount of folds. For each iteration one fold is held out as "training-dev" set and the other folds are used as training set. Result of cross-validation is k values for all metrics on the k-fold CV.

K Nearest Neighbors (Optimized)
For KNN GridSearch CV is used to determine the optimal number of neighbors (k) leading to the best model performance.

Logistic Regression (Optimized)
For Logistic Regression GridSearchCV is used to determine the best model while applying different values of L1 or L2 regularization to turn the impact of non-meaningful feature to zero (L1) or to simplify the model by relativizing strong patterns that are picked up during training (L2).

Random Forest (Optimized)
For the Random Forest model RandomizedSearchCV is used to optimize for several hyperparameters including n_estimators, max_features, max_depth, criterion and bootstrap.

Support Vector Machine (optimized)
For SVM GridSearchCV is used to determine the C value for the optimal margin around the support vector.

Although the data set is relatively small and neural networks generally require lots of training data to develop meaningful prediction capabilities, a simple neural network is employed for a quick comparison to the other approaches.

Summary

Model Summary

Looking at model results, the best accuracy on the test set is achieved by the neural network with 0,7996. Given the high imbalance of the data towards non-churners, it makes sense to compare F1 scores to get the model with the best score on jointly precision and recall. This would also be the neural network with a F1 score of 0,5948.

Given the scores of the best performing models, it can be observed that F1 scores are not much above 50%. Further optimization efforts should be carried out to achieve a higher scores and thereby increase prediction power for more business value.

Hypotheses Check

Looking at the evaluation results, specifically the feature weights from the logistic regression, the hypotheses can be directionally supported or refused:

Contract duration: Contract duration month-to-month is the second biggest driver of churn → supported

Number of additional services: This feature does not rank among the top features → refused

Partners and children: Having children ranks as the fourth feature that drives not churning, but strength is relatively low → partially supported

Tenure: High tenure ranks as the strongest factor for not churning and the strongest feature overall. This is also supported by the boxplot in the EDA step. → supported

Monthly payment: Total payments, which is the product of tenure and monthly payment ranks as the strongest factor for churn. Indirectly, high monthly payments lead to churn. However, tenure is the highest driver of not churning → refused

Senior citizens: Senior citizens does not have high feature weights. Also the ratio of senior citizens who churn is much higher than that of non-churners → refused

Outlook

Telcos typically have much more data available that could be included in the analysis, like extended customer and transaction data from CRM systems and operational data around network services provided. Also they typically have much larger amounts of churn/non-churn events at their disposal than the ca. 7000 in this case example. With those, neural networks could be properly trained to detect more complex patterns in data and achieve higher accuracies. A high accuracy is needed to be able to identify promising customer cases where churn can be avoided as, eventually, the customer returns protected need to outweigh the costs of related retention campaigns.