Model use for churn prediction problem:

**Multi-layer Perceptron (Neural Network)**

A **multilayer perceptron** (**MLP**) is a fully connected class of feedforward artificial neural network (ANN). The term MLP is used ambiguously, sometimes loosely to mean *any* feedforward ANN, sometimes strictly to refer to networks composed of multiple layers of perceptrons (with threshold activation). Multilayer perceptrons are sometimes colloquially referred to as "vanilla" neural networks, especially when they have a single hidden layer.

An MLP consists of at least three layers of nodes: an input layer, a hidden layer and an output layer. Except for the input nodes, each node is a neuron that uses a nonlinear activation function. MLP utilizes a supervised learning technique called backpropagation for training. Its multiple layers and non-linear activation distinguish MLP from a linear perceptron. It can distinguish data that is not linearly separable.

The MLP model include:

Activation function: Mainly ReLU or sigmoid function

Layers:

The MLP consists of three or more layers (an input and an output layer with one or more hidden layers) of nonlinearly-activating nodes.

Learning:

Using forward propagation, backpropagation and gradient descent, the model will learn the pattern of the training data and apply it onto the testing data.

**Logistic Regression**

**Logistic regression** (or **logit regression**) is estimating the parameters of a logistic model (the coefficients in the linear combination). Formally, in binary logistic regression there is a single binary dependent variable, coded by an indicator variable, where the two values are labeled "0" and "1", while the independent variables can each be a binary variable (two classes, coded by an indicator variable) or a continuous variable (any real value). The corresponding probability of the value labeled "1" can vary between 0 (certainly the value "0") and 1 (certainly the value "1"), the function that converts log-odds to probability is the logistic function.

**Random Forest**

**Random forests** or **random decision forests** is an ensemble learning method for classification, regression and other tasks that operates by constructing a multitude of decision trees at training time. For classification tasks, the output of the random forest is the class selected by most trees. For regression tasks, the mean or average prediction of the individual trees is returned.

**XGBoost**

**XGBoost** (eXtreme Gradient Boosting) is an open-source software library which provides a regularizing gradient boosting framework for C++, Java, Python, R, Julia, Perl, and Scala. It works on Linux, Windows, and macOS.

Salient features of XGBoost which make it different from other gradient boosting algorithms include:

* Clever penalization of trees
* A proportional shrinking of leaf nodes
* Newton Boosting
* Extra randomization parameter
* Implementation on single, distributed systems and out-of-core computation
* Automatic Feature selection

**Stacking Model**

Stacking or Stacked Generalization is an ensemble machine learning algorithm.

It uses a meta-learning algorithm to learn how to best combine the predictions from two or more base machine learning algorithms.

The benefit of stacking is that it can harness the capabilities of a range of well-performing models on a classification or regression task and make predictions that have better performance than any single model in the ensemble.

The architecture of a stacking model involves two or more base models, often referred to as level-0 models, and a meta-model that combines the predictions of the base models, referred to as a level-1 model.

* **Level-0 Models (*Base-Models*)**: Models fit on the training data and whose predictions are compiled.
* **Level-1 Model (*Meta-Model*)**: Model that learns how to best combine the predictions of the base models.

The meta-model is trained on the predictions made by base models on out-of-sample data. That is, data not used to train the base models is fed to the base models, predictions are made, and these predictions, along with the expected outputs, provide the input and output pairs of the training dataset used to fit the meta-model.

The outputs from the base models used as input to the meta-model may be real value in the case of regression, and probability values, probability like values, or class labels in the case of classification.

**Pycaret Model Detection (Best outcome: Catboost)**

PyCaret is an **open-source**, **low-code** machine learning library in Python that aims to reduce the hypothesis to insights cycle time in an ML experiment. It enables data scientists to perform end-to-end experiments quickly and efficiently. In comparison with the other open-source machine learning libraries, PyCaret is an alternate low-code library that can be used to perform complex machine learning tasks with only a few lines of code. PyCaret is **simple and** **easy to use**.

**Catboost**

CatBoost is a machine learning algorithm that uses gradient boosting on decision trees. It’s also had pandas and numpy integrated.