Predicting loan default using the lending club

Problem Description :

The objective of this project is to build a machine learning model that can predict whether a borrower will default on a loan or not. The project will use the LendingClub dataset, which is available on Kaggle. The LendingClub is a peer-to-peer lending platform that connects borrowers with investors. The dataset contains information on loans issued by LendingClub between 2007 and 2018, including the borrower's credit score, employment history, loan amount, interest rate, and loan status.

Dataset Link:

The dataset can be downloaded from Kaggle at the following link: https://www.kaggle.com/wendykan/lending-club-loan-data.

Deliverables:

The deliverables of this project will include a trained machine learning model that can predict loan default, as well as an evaluation of the model's performance on a test set. The project will also include exploratory data analysis and data preprocessing steps to prepare the dataset for modeling.

Dataset Description:

The LendingClub dataset contains information on 2.2 million loans issued by LendingClub between 2007 and 2018. Each row in the dataset represents a single loan, and the columns contain information on the borrower's credit score, employment history, loan amount, interest rate, and loan status. The loan status is the target variable for this project, and it indicates whether the borrower has fully paid off the loan or has defaulted on the loan.

Framework for Predicting Loan Default:

1. **Exploratory Data Analysis (EDA):**

    - Load the dataset and check for missing values.
    - Explore the distribution of the target variable.
    - Analyze the distribution of each feature and check for outliers.
    - Visualize the relationship between features and the target variable.

2. **Data Preprocessing:**

    - Remove unnecessary columns and rows.
    - Handle missing values by either imputing or dropping them.
    - Encode categorical variables using one-hot encoding or label encoding.
    - Scale numerical features using standardization or normalization.
    - Split the data into training and test sets.

3. **Feature Engineering:**

    - Create new features from existing features.

- Transform features using logarithmic, exponential or power functions.
- Remove correlated features.
- Use domain knowledge to engineer new features.

4. **Model Selection and Training:**

- Choose a suitable machine learning algorithm such as Logistic Regression, Random Forest or Gradient Boosting.
- Tune hyperparameters of the chosen algorithm using GridSearchCV or RandomizedSearchCV.
- Train the model on the training set.

5. **Model Evaluation:**

- Evaluate the performance of the model on the test set using metrics such as accuracy, precision, recall, and F1 score.
- Plot the confusion matrix to visualize the performance of the model.
- Analyze the importance of each feature using feature importance plot.

6. **Model Deployment:**

- Deploy the trained model in a production environment.

- Create an API or web application that allows users to input loan information and receive a prediction of whether the loan is likely to default.

- Continuously monitor the model’s performance and update the model as necessary.

Code Explanation :

Here is the simple explanation for the code which is provided in the code.py file.

Section 1: Importing Libraries and Loading Data

In this section, we import the necessary libraries that we will use throughout the code, such as pandas, numpy, matplotlib, and sklearn. We also load the LendingClub dataset into a pandas DataFrame using the read_csv() function.

Section 2: Data Preprocessing

In this section, we first drop any columns that have more than 50% missing values since they are unlikely to provide much useful information for our model. We also drop any rows that have missing values in the remaining columns.

Next, we select the columns that we want to use as features in our model and the target column that we want to predict. We then convert any categorical variables into numerical variables using one-hot encoding, and scale the numerical features to have zero mean and unit variance using the StandardScaler from sklearn.

Finally, we split the preprocessed data into training and testing sets using the train_test_split() function from sklearn.

Section 3: Model Building and Training

In this section, we define a dictionary of machine learning models that we want to train, including Logistic Regression, Decision Tree, Random Forest, Support Vector Machine, and Gradient Boosting. We then loop through each model, train it on the training data using the fit() method, and calculate its accuracy score on the test data using the score() method. We store the accuracy scores in a dictionary called model_scores.

After training all the models, we plot the accuracy scores using matplotlib’s bar() function. The x-axis shows the names of the models, while the y-axis shows their accuracy scores. This helps us compare the performance of the different models and choose the best one for our problem.

Running the Code

To run this code, you need to have Python 3 and the necessary libraries installed, such as pandas, numpy, matplotlib, and sklearn. You can install these libraries using pip by running the following command in your terminal:

```
Pip install pandas numpy matplotlib sklearn
```

Once you have installed the necessary libraries, you can simply copy and paste the code into a Python script, and run the script. The script will load the LendingClub dataset, preprocess the data, train several machine learning models, and plot the accuracy scores of the models.

Note that the script assumes that the LendingClub dataset is saved in a CSV file called ‘lendingclub_loan.csv’ in the same directory as the script. If your dataset is saved in a different file or location, you will need to modify the code accordingly.

Future Work :

Introduction: The aim of this project is to predict the likelihood of a loan default using the LendingClub dataset. The future work for this project involves exploring various other techniques and models to improve the performance of the existing model. Here we provide a step-by-step guide on how to implement the future work for this project.

1. **Feature Engineering:** The first step in improving the model's performance is to explore additional feature engineering techniques. We can generate additional features from existing ones or use external data sources to create new features. Some examples of feature engineering techniques include:

    - One-hot encoding categorical variables

    - Creating new variables based on domain knowledge

    - Transforming variables using mathematical functions

2. **Model Selection:** We can explore additional models to improve the performance of the existing model. Some models to explore include:

    - Gradient Boosting Machines (GBMs)

    - Random Forest

    - Neural Networks

    - Support Vector Machines (SVMs)

3. **Hyperparameter Tuning:** After selecting additional models, we can fine-tune their hyperparameters to optimize their performance. We can use techniques such as grid search and random search to find the best combination of hyperparameters.

4. **Ensemble Techniques:** We can explore ensemble techniques such as:

   - Stacking
   - Blending
   - Bagging
   - Boosting

These techniques involve combining multiple models to improve their overall performance.

5. **Interpretability:** We can also explore techniques to explain the model’s predictions. Some techniques to explore include:

   - Partial dependence plots
   - SHAP values
   - LIME

These techniques help to understand how the model is making predictions and can be useful in gaining insights into the data.

Step-by-Step Implementation Guide:

1. **Feature Engineering:**

    - Use one-hot encoding to encode categorical variables
    - Create new variables based on domain knowledge
    - Transform variables using mathematical functions
    - Use external data sources to create new features

2. **Model Selection:**

    - Explore additional models such as GBMs, Random Forest, Neural Networks, SVMs
    - Train and evaluate the models on the dataset

3. **Hyperparameter Tuning:**

    - Fine-tune the hyperparameters of the models using grid search or random search

4. **Ensemble Techniques:**

- Explore ensemble techniques such as Stacking, Blending, Bagging, Boosting
- Combine the models using these techniques and evaluate their performance

5. **Interpretability:**

- Use techniques such as Partial dependence plots, SHAP values, LIME to explain the model's predictions and gain insights into the data.

Requirements:

- Python 3
- Jupyter Notebook or any Python IDE
- Required libraries: pandas, numpy, scikit-learn, xgboost, matplotlib, seaborn, eli5

Conclusion: Implementing these future work steps will help to improve the performance of the existing model and gain more insights into the data. Using the appropriate techniques and models, we can create a robust and accurate loan default prediction model that can be useful in various industries.

Exercise :

Try to answers the following questions by yourself to check your understanding for this project. If stuck, detailed answers for the questions are also provided.

1. **What is the purpose of performing one-hot encoding on categorical variables in this loan default prediction project?**
   Answer: The purpose of performing one-hot encoding on categorical variables is to transform them into numerical variables that can be used in the machine learning algorithms for prediction. One-hot encoding is a process that converts categorical variables into a series of binary variables, where each variable represents a distinct category. This allows the machine learning algorithms to understand and use the information in the categorical variables.

2. **What is the purpose of performing feature scaling on the numerical variables in this loan default prediction project?**
   Answer: The purpose of performing feature scaling on the numerical variables is to ensure that all variables are on a similar scale, so that the machine learning algorithms can better understand the relationships between them. Feature scaling is a process of scaling the values of a variable to a specific range, usually between 0 and 1 or -1 and 1. This is important because some machine learning algorithms are sensitive to the scale of the input features.

3. **What is the purpose of splitting the dataset into training and testing sets in this loan default prediction project?**
   Answer: The purpose of splitting the dataset into training and testing sets is to evaluate the performance of the machine learning algorithm on unseen data. The training set is used to train the machine learning model, while the testing set is used to evaluate the model's performance. This is important to ensure that the model is not overfitting to the training data and can generalize well to new data.

4. **What is the purpose of using a grid search with cross-validation to tune the hyperparameters of the machine learning model in this loan default prediction project?**

Answer: The purpose of using a grid search with cross-validation to tune the hyperparameters of the machine learning model is to find the optimal combination of hyperparameters that maximizes the performance of the model on the testing set. Grid search is a method of searching through a range of hyperparameters to find the best combination, and cross-validation is a technique that helps to reduce the risk of overfitting by evaluating the model's performance on multiple folds of the training data.

5. **How can the performance of the machine learning model be further improved in this loan default prediction project?**
   Answer: The performance of the machine learning model can be further improved in several ways. One approach is to try different machine learning algorithms and compare their performance. Another approach is to use more advanced feature engineering techniques to create new features that capture more information from the dataset. Additionally, using more data or data from different sources can also improve the performance of the model. Finally, ensembling multiple models together can help to improve the predictive power of the model.