| REPORT ON  CREDIT CARD APPROVAL PREDICTION  PG-DBDA SEPTEMBER 2022  Submitted By :-  **PROJECT TEAM 3**    **Abhinav Prakash Mahendra Sonawane**  **Pratik Rasam Atharv Narvekar**  **Harshal Badgujar Vishal Gaikwad**  Project Mentor:  **Dhiraj Deore**  Course Coordinator:  **Vineeta Singh Mam**  **ABSTRACT**  Nowadays, banks receive a lot of applications for issuance of credit cards. Many of them are rejected for many reasons, like high-loan balances, low-income levels, or too many inquiries on an individual’s credit report.  Manually analyzing these applications is error-prone and a time-consuming process. Luckily, this task can be automated with the power of machine learning and pretty much every bank does so nowadays.    In this project, we will build an automatic credit card approval predictor using machine learning techniques, just like real banks do.  **Table of Contents**  1. Introduction………………………………………………………………………  2. Problem Statement………………………………………………………  3. Technology & Libraries Used …………………..…………………  4. Architecture & Methodology……………………………………  5. Project Implementation………………………………………………………  5. Conclusion ...……………………………………………………………    **INTRODUCTION**    The use of credit cards has become increasingly popular in today's society, and it is essential for financial institutions to have a reliable system to assess an individual's creditworthiness before approving their credit card application. A credit card approval prediction system helps financial institutions to streamline their application processes, reduce operational costs, and improve the overall customer experience.  By leveraging machine learning algorithms, the credit card approval prediction system can analyze large volumes of data and detect patterns that might be challenging for human experts to identify. This system can, therefore, provide more accurate and consistent decisions regarding credit card approvals, leading to better risk management and improved profitability for financial institutions.  A credit card approval prediction system is a machine learning model designed to predict whether an individual is likely to get their credit card application approved or not. This system utilizes a range of data inputs such as the applicant's credit history, income, employment status, and debt-to-income ratio, among others, to generate a prediction of the likelihood of approval.      **PROBLEM STATEMENT**  The current credit card approval process is often time-consuming, expensive, and may not be consistent due to human errors and biases. Financial institutions face challenges in assessing an individual's creditworthiness accurately, which can result in high risk and increased costs. Additionally, the traditional credit scoring models might not capture all relevant information, leading to suboptimal credit decisions.  Thus, the credit card approval prediction system aims to overcome these challenges by providing an automated and reliable solution for assessing creditworthiness. The system can analyze a large volume of data and generate accurate predictions, thereby improving the efficiency of the credit card approval process and reducing operational costs. The ultimate goal of this system is to provide a better customer experience by reducing the time and effort required for credit card applications and improving the overall credit decision-making process.  To develop a machine learning model with taking factors into consideration that can accurately predict the likelihood of an individual getting their credit card application approved or not based on various input variables such as credit history, income, employment status, debt-to-income ratio, among others.    **TECHNOLOGY & LIBRARIES USED**  Credit card approval prediction in machine learning involves the use of various technologies and libraries, including:  1.**Programming Languages:** Python, R commonly used programming languages for machine learning tasks.  2.**Machine Learning Frameworks**: Popular machine learning frameworks include Scikit-learn.  3.**Data Visualization Libraries:** Matplotlib, Seaborn, and Plotly are widely used data visualization libraries that help in the visualization of data patterns.  4**.Data Preprocessing Libraries:** Pandas and NumPy are data preprocessing libraries that are useful for data cleaning, transformation, and manipulation.  5.**Feature Engineering Libraries:** sklearn and PCA are feature engineering libraries that help to extract useful features from raw data.  6.**Model Evaluation Libraries:** Scikit-learn provides metrics such as accuracy, precision, recall, F1-score, and AUC-ROC to evaluate the performance of the model.  7.**Model Interpretation Libraries:** SHAP (SHapley Additive exPlanations), LIME (Local Interpretable Model-Agnostic Explanations), and ELI5 (Explain Like I'm Five) are model interpretation libraries that help in understanding the model's decision-making process.  8.**Data Visualization Tool:** PowerBi  9.**Interface** : DJango    **Project Architecture**    **METHODOLOGY & IMPLEMENTATION**    **A.Datasource & Dataset :**   * **Here we used a dataset from Kaggle composed of 2 files .**   + **Credit\_Record. csv** :Contains ID and status of credit of the respective ID's   + **Application\_Record.csv:**  This file contains other informations regarding the ID's such as Educational status, Job Profiles, family count, total income ,etc.       **B.Data Storage:**   * We have used Local Machine to store the Data.   **C.Libraries and packages used :**  import pandas as pd  import numpy as np  import matplotlib.pyplot as plt  import matplotlib  import seaborn as sns  from scipy.stats import probplot, chi2\_contingency, chi2  from sklearn.model\_selection import train\_test\_split, GridSearchCV, RandomizedSearchCV, cross\_val\_score, cross\_val\_predict  from sklearn.compose import ColumnTransformer  from sklearn.preprocessing import OneHotEncoder, MinMaxScaler, OrdinalEncoder  from sklearn.metrics import ConfusionMatrixDisplay, classification\_report, roc\_curve, roc\_auc\_score,accuracy\_score  from imblearn.over\_sampling import SMOTE  from sklearn.linear\_model import SGDClassifier, LogisticRegression  from sklearn.svm import SVC  from sklearn.tree import DecisionTreeClassifier  from sklearn.ensemble import RandomForestClassifier, GradientBoostingClassifier, BaggingClassifier, AdaBoostClassifier, ExtraTreesClassifier  from sklearn.naive\_bayes import GaussianNB  from sklearn.neighbors import KNeighborsClassifier  from sklearn.discriminant\_analysis import LinearDiscriminantAnalysis  from sklearn.neural\_network import MLPClassifier  from sklearn.inspection import permutation\_importance  import scikitplot as skplt  from yellowbrick.model\_selection import FeatureImportances  import scipy.stats as stats  import joblib  %matplotlib inline  **D.Data Cleaning and Preparation**  The credit card approval prediction dataset is a common dataset used in machine learning and predictive modeling. The dataset typically contains a variety of information about credit card applicants, such as their age, income, employment status, credit history, and other relevant factors that can be used to predict whether an applicant's credit card application will be approved or denied.  PySpark can be used to preprocess and clean the dataset by identifying missing or invalid data, encoding categorical variables, and scaling numerical features. Feature engineering can also be performed to create new features that might be predictive of credit card approval, such as debt-to-income ratio or credit utilization rate. The preprocessed data can then be used to train a machine learning model using PySpark, such as logistic regression, decision trees, or random forest, to predict whether a credit card application will be approved or denied.  By using PySpark for credit card approval prediction, it is possible to handle large-scale datasets and perform distributed processing, which can improve the efficiency and accuracy of the predictive models. This can be particularly important for credit card companies that receive large volumes of credit card applications and need to process them quickly and accurately.  **Handling Missing values**  Handling missing values is an important step in data preparation to ensure the accuracy and effectiveness of machine learning models. Here are some common approaches for handling missing values using PySpark:  Dropping missing values: This approach involves dropping any rows that contain missing values. While this approach is simple and easy to implement, it can result in loss of valuable information and may lead to biased results if the missing values are not randomly distributed across the dataset.  Imputation: Imputation involves filling in the missing values with some estimated value, such as the mean, median, or mode of the corresponding feature. PySpark provides several functions to perform imputation, such as fillna() or replace(). However, imputation can also introduce bias if the missing values are not missing at random or if the imputed values are not accurate.  Using machine learning algorithms: PySpark also provides algorithms that can handle missing values during training, such as decision trees or random forest. These algorithms can impute missing values based on the available data and the relationships between features. However, these methods may require more computation time and may not always be effective for handling large amounts of missing data.  In summary, handling missing values is an important step in data preparation and PySpark provides several options to address this issue. The best approach depends on the specific dataset and the goals of the analysis.  **Exploratory Data Analysis:**   * **Univariate Analysis:**  1. **Analysis of Gender**      * Most of the applicants (67%) are Female.  1. **Analysis of Age**      * The median Age of applicants is 43. Highest number of the applicants fall within the Age group of 30-40 Years.  1. **Analysis of Family Status**      * Highest number of applicants are from Married people.  1. **Analysis of Education**      * Highest number of applicants have completed Secondary Education  1. **Analysis of Target Variable**      * Most of the applicants are not considered as 'Risk' * There's a high imbalance in data which needs to be handled while training models |
| --- |
| * **Bivariate Analysis :**   **1)Scatterplots for columns such as Employment Length vs age, Account Age , Total Income, etc**     * We can see a positive linear correlation between the family member and the children count. This makes sense, the more the children someone have, the larger the family member count. This is a multicollinearity problem. Meaning that the features are highly correlated. We will need to drop one of them. * Another interesting trend is the Employment length and age. This also makes sense, the longer the employee has been working, the older they are.   2) Correlation Analysis with Heatmap     * There is no feature that is correlated with the target feature * Family member count is highly correlated with children count as previously discussed * Age has some positive correlation with the family member count and children count. The older a person is, the most likely he/she will have a larger family. * Another positive correlation is having a phone and having a work phone. * The final positive correlation is between the age and work phone. The younger someone is the less likely he/she will have a work phone. * 6.We also have a negative correlation between the employment length and the age as previously seen.   **3) ANOVA Test for Age vs Rest of categories**     * Female applicants are older than their male counterpart. * Those who don't own a car tend to be older. * Those who own a property tend to be older than those who don't. * Of course, the pensioners are older that those who are working (We also see that some have pensioned at a young age, those are outliers). * It is also interesting to see that those who hold an academic degree are younger in general than the other groups. * Obviously, the widows tend to be much older. We also see some outliers in their 30's as well. * With no surprise, those who live with parent tend to be younger. We also see some outlier as well. * Lastly, who work as cleaning staff tend to be older while those who work in IT and to be younger. * **Overall Business Findings:** * a Female in her early 40’s, married with a partner and no child. She has been employed for 5 years with a salary of 157500. She has completed her secondary education. She does not own a car but owns a property (a house/ apartment). Her account is 26 months old. Age and income do not have any effects on the target variable Those who are flagged as bad client, tend to have a shorter employment length and older accounts. They also constitute less than 2% of total applicants. * Most applicants are 20 to 45 years old and have an account that is 25 months old or less.   **E.Data Preparation & Model Building:**  **1)Tranformations on Each Future**   1. **ID, CNT\_CHILDREN,FLAG\_MOBIL,Account age:Dropping the feature**      1. **CODE\_GENDER,NAME\_FAMILY\_STATUS,NAME\_FAMILY\_STATUS, Housing type ,Employment status , FLAG\_WORK\_PHONE , FLAG\_PHONE , FLAG\_EMAIL , NAME\_EDUCATION\_TYPE: creating dummies columns**      1. **Age:Min-max scaling and Fix skewness**      1. **CNT\_FAM\_MEMBERS:Fix outliers**      1. **AMT\_INCOME\_TOTAL:Remove outliers and Fix skewness and Min-max scaling**      1. **OCCUPATION\_TYPE:One hot encoding and Impute missing values**      1. **Employment length:Remove outliers and Min-max scaling**      1. **FLAG\_OWN\_CAR, FLAG\_OWN\_REALTY :Change it numerical and get dummies and drop original.**     **t)Target:Change the data type to numerical and balance the data**    **F. Model Selection , Building , Evaluation**  **1)Logistic Regression:**   * **Logistic regression is a statistical technique used to analyze relationships between a categorical dependent variable and one or more independent variables. The dependent variable is typically binary (e.g., "yes" or "no," "0" or "1"), but it can be extended to handle multiple categories. The goal of logistic regression is to model the probability of the dependent variable, given the values of the independent variables.** * **In logistic regression, the independent variables are transformed using a logistic function, which maps the values onto the range [0,1]. The resulting values are interpreted as probabilities, representing the likelihood of the dependent variable being in a particular category. The logistic function allows for the model to be linear in the parameters, but nonlinear in the independent variables.** * **The logistic regression model is fitted by estimating the parameters that maximize the likelihood of the observed data, given the model. This is typically done using a method called maximum likelihood estimation. The model can then be used to make predictions about the probability of the dependent variable, given a set of values for the independent variables.** * **Logistic regression is widely used in many fields, including epidemiology, finance, and social sciences, to model binary outcomes and to understand the relationships between variables. It is also used as a building block for more complex models, such as neural networks and decision trees.** * **Some Key terms for evaluation:**   **a. Classification Report Related terms:**  **1.Precision: Precision measures the proportion of true positives among all the predicted positive instances. In other words, it measures how often the model correctly identifies a positive instance. Precision is defined as:**  **precision = true positives / (true positives + false positives)**  **2.Recall: Recall, also known as sensitivity or true positive rate, measures the proportion of true positives among all actual positive instances. In other words, it measures how often the model correctly identifies a positive instance from all the actual positive instances. Recall is defined as:**  **recall = true positives / (true positives + false negatives)**  **3.F1 score: F1 score is the harmonic mean of precision and recall. It is used to balance the tradeoff between precision and recall. F1 score is defined as:**  **F1 score = 2 \* (precision \* recall) / (precision + recall)**  **4.Accuracy: Accuracy measures the proportion of correct predictions among all predictions. It is defined as:**  **accuracy = (true positives + true negatives) / (true positives + true negatives + false positives + false negatives)**   * **In general, accuracy is not always a good metric to evaluate a model's performance, especially when the dataset is imbalanced or when there is a high cost associated with**   **misclassifying certain instances. In such cases, precision, recall, and F1 score provide a better understanding of the model's performance.**  **B. ROC Curve**   * **A Receiver Operating Characteristic (ROC) curve is a graphical representation of the performance of a binary classifier, as the discrimination threshold is varied.** * **The ROC curve plots the true positive rate (TPR) against the false positive rate (FPR) for different classification thresholds. TPR is also known as sensitivity or recall, and it is the proportion of actual positive instances that are correctly identified as positive by the model. FPR is the proportion of actual negative instances that are incorrectly identified as positive by the model.** * **The ROC curve shows the trade-off between TPR and FPR for different classification thresholds. A perfect classifier would have a TPR of 1 and an FPR of 0, which corresponds to the point (0,1) in the ROC space. A random classifier would have a diagonal line from (0,0) to (1,1), with an AUC (Area Under the Curve) of 0.5. The closer the ROC curve is to the top-left corner, the better the classifier's performance.** * **The area under the ROC curve (AUC) is a commonly used metric to evaluate the overall performance of a binary classifier. The AUC ranges from 0.5 (random classifier) to 1 (perfect classifier). An AUC of 0.5 indicates that the classifier is not better than random guessing, while an AUC of 1 indicates that the classifier makes perfect predictions.**   **C. Confusion Matrix**   * **A confusion matrix is a table that summarizes the performance of a binary or multi-class classification model. It shows the number of correct and incorrect predictions made by the model, broken down by class (or category) of the predicted and actual values.** * **For a binary classification problem, the confusion matrix has four possible outcomes:** * **True positive (TP): The model correctly predicted a positive instance (i.e., the predicted value is positive and the actual value is also positive).** * **False positive (FP): The model incorrectly predicted a positive instance (i.e., the predicted value is positive but the actual value is negative).** * **True negative (TN): The model correctly predicted a negative instance (i.e., the predicted value is negative and the actual value is also negative).** * **False negative (FN): The model incorrectly predicted a negative instance (i.e., the predicted value is negative but the actual value is positive).** * **The confusion matrix is usually presented in a tabular form, with the predicted values on the vertical axis and the actual values on the horizontal axis.** * **Model Building**      * **Model Evaluation**      * **ROC curve** * **Confusion Matrix**      * **Top 10 Features**       **2)Decision Tree Classifier**   * **A decision tree classifier is a type of supervised machine learning algorithm used for classification and regression analysis. It is a tree-structured model in which internal nodes represent tests on features, branches represent the outcomes of the tests, and leaf nodes represent the class labels or regression values.** * **The decision tree classifier works by recursively partitioning the input data into smaller and smaller subsets based on the values of the input features, until the subsets are as pure as possible in terms of the target variable. The purity of a subset is typically measured by a metric such as entropy or Gini impurity.** * **During the training process, the decision tree algorithm selects the best feature to split the data at each internal node, based on the purity of the resulting subsets. The splitting process continues until a stopping criterion is met, such as reaching a maximum depth or a minimum number of samples in a leaf node.** * **Once the decision tree is constructed, it can be used to make predictions on new input data by traversing the tree from the root node to a leaf node, based on the values of the input features. The class label or regression value of the leaf node is then used as the prediction.** * **Decision tree classifiers have several advantages, such as being easy to interpret and visualize, and being able to handle both categorical and numerical features. They also work well with both small and large datasets. However, they can suffer from overfitting if the tree is too complex, and they may not perform well if the input data contains noisy or irrelevant features.** * **Model Building**      * **Model Evaluation**        * **ROC curve**      * **Confusion Matrix**      * **Top 10 Features**      * **Bottom 10 Features**     **3)Random Forest Classifier**   * **Random Forest Classifier is an ensemble learning algorithm based on decision trees, used for classification and regression analysis. It builds multiple decision trees and combines their predictions to obtain a more accurate and stable prediction.** * **The random forest classifier works by building a large number of decision trees on different subsets of the input data and input features. Each tree is trained on a bootstrap sample of the data, which is a random sample of the data with replacement, and a subset of the input features, which is randomly selected for each tree. This randomness helps to reduce overfitting and improve generalization.** * **During the training process, each tree in the random forest classifier is constructed by recursively partitioning the input data into smaller and smaller subsets based on the values of the input features, using a similar process as the decision tree classifier. The difference is that the random forest classifier randomly selects a subset of the input features to consider at each node of the tree, which reduces the correlation among the trees.** * **Once the decision trees are constructed, the random forest classifier combines their predictions by taking the majority vote (for classification) or the average (for regression). This aggregation of predictions helps to reduce the variance and improve the accuracy and robustness of the model.** * **Random forest classifier has several advantages, such as being able to handle large datasets with high dimensionality, being less prone to overfitting than a single decision tree, and being able to estimate the importance of the input features. It is widely used in various applications, such as image classification, speech recognition, and financial forecasting.** * **Model Building**      * **Model Evaluation**      * **Confusion Matrix**      * **ROC Curve**      * **Top 10 Features**      * **Bottom 10 Features**     **4)Gaussian Naïve Bayes Classifier**   * **Model Building**      * **Model Evaluation**      * **ROC Curve**      * **Confusion Matrix**     **5)K-Nearest Neighbour Classifier**   * **K-Nearest Neighbor (K-NN) Classifier is a type of supervised machine learning algorithm used for classification tasks. It is a simple and intuitive method that uses a distance metric to identify the K closest training examples (i.e., neighbors) to a new input sample and then predicts the label of the sample based on the majority label of its K neighbors.** * **During the training phase, the K-NN classifier stores all of the training data samples and their corresponding class labels. When a new input sample is presented to the classifier for classification, it calculates the distance between the input sample and all of the training samples using a distance metric such as Euclidean distance, Manhattan distance, or cosine similarity.** * **Once the distances are calculated, the K-NN algorithm selects the K training samples that are closest to the input sample, based on the distance metric. The algorithm then assigns the class label to the input sample based on the majority class among its K nearest neighbors.** * **The value of K is an important parameter in the K-NN algorithm and should be chosen carefully. A small value of K may lead to overfitting, while a large value of K may lead to underfitting. Therefore, the value of K is often determined using cross-validation or other model selection techniques.** * **K-NN classifier has several advantages, such as being simple and easy to implement, being able to handle multi-class classification problems, and being able to adapt to complex decision boundaries. However, it can be computationally expensive to calculate distances for large datasets and high-dimensional feature spaces. Additionally, it may not perform well when the data contains noisy or irrelevant features.** * **Model Building**      * **Model Evaluation**      * **ROC Curve**      * **Confusion Matrix**     **6)Gradient Boosting Classifier**   * **Gradient Boosting Classifier is an ensemble learning algorithm that combines multiple weak learning models, such as decision trees, to create a stronger prediction model. It works by iteratively improving the model by adding new weak learners to the model, which are trained to correct the errors of the previous learners.** * **During the training phase, the Gradient Boosting Classifier starts by building an initial weak learner, such as a decision tree, on the input data. The subsequent weak learners are then trained to predict the residual errors of the previous learners, in order to improve the overall prediction accuracy of the model.** * **In each iteration, the Gradient Boosting Classifier calculates the negative gradient of the loss function with respect to the predicted values of the previous learners, and uses this as the target variable for the new learner. The new learner is then trained to minimize this target variable, using the input data and the residuals of the previous learners.** * **Once the new learner is trained, its prediction is added to the predictions of the previous learners, weighted by a learning rate parameter that controls the contribution of each learner to the final prediction. The process of adding new learners and updating the prediction continues until a stopping criterion is met, such as reaching a maximum number of iterations or a minimum improvement in performance.** * **Gradient Boosting Classifier has several advantages, such as being able to handle heterogeneous data types, being less prone to overfitting than a single decision tree, and being able to capture complex non-linear relationships between the input variables and the target variable. However, it can be computationally expensive and may require careful tuning of hyperparameters such as the learning rate, the maximum depth of the decision trees, and the number of learners.** * **Model Building**      * **Model Evaluation**      * **ROC Curve**     **Confusion Matrix**     * **Top 10 Features**      * **Bottom 10 Features**     **7)Linear Discremenant Analysis**   * **Linear Discriminant Analysis (LDA) is a supervised learning algorithm used for classification tasks. It is a dimensionality reduction technique that projects the high-dimensional input data into a lower-dimensional space while maximizing the separation between the classes.** * **LDA works by finding the linear combinations of the input features that best separate the classes. It does this by maximizing the ratio of the between-class variance to the within-class variance, which measures the distance between the means of the classes relative to the scatter within each class.** * **During the training phase, LDA learns the projection matrix that maps the input data onto the lower-dimensional space. This projection matrix is obtained by solving the generalized eigenvalue problem between the within-class scatter matrix and the between-class scatter matrix.** * **Once the projection matrix is learned, it can be used to project new input data into the lower-dimensional space. The LDA classifier then uses a decision rule to classify the input data based on the location of the projected data points in the lower-dimensional space.** * **LDA has several advantages, such as being a linear classifier, being robust to overfitting, and being able to handle high-dimensional input data. Additionally, LDA can be used for data visualization, as the lower-dimensional space can be easily visualized. However, LDA may not perform well when the classes are highly overlapping or when the assumptions of normality and equal covariance matrices are violated.**      * **Efficiency of finding Curse of Dimensionality**      * **ROC Curve**      * **Confusion Matrix**      * **Top 10 Important Feature Columns**      * **Top 10 Bottom Important Feature Columns**     **8)AdaBoost Classifier**   * **AdaBoost (Adaptive Boosting) Classifier is an ensemble learning algorithm used for classification tasks. It works by combining multiple weak learners, such as decision trees or stumps, to create a strong prediction model.** * **During the training phase, AdaBoost assigns equal weights to each training example and trains a weak learner on the input data. The weak learner is then evaluated on the training data, and the weights of the misclassified examples are increased to emphasize their importance in the subsequent iterations.** * **In each iteration, AdaBoost trains a new weak learner on the re-weighted training data and combines the predictions of the previous learners with a weighted sum, where the weights are based on the performance of the previous learners on the training data.** * **The final prediction is then made by combining the predictions of all the weak learners, weighted by their performance on the training data. The weights assigned to each weak learner depend on its accuracy in the training phase, with more accurate models receiving higher weights.** * **AdaBoost has several advantages, such as being able to handle complex decision boundaries, being able to combine different types of weak learners, and being less prone to overfitting than a single decision tree. Additionally, AdaBoost can be used for both binary and multi-class classification tasks. However, AdaBoost may be sensitive to noisy or mislabeled data and may require careful tuning of hyperparameters such as the number of learners and the learning rate.** * **Model Building**      * **Model Evaluation**      * **ROC Curve**      * **Confusion Matrix**      * **Top 10 Important Features**     **DEPLOYMENT ON DJango**  Django is a high-level web framework written in Python that enables rapid development of secure and maintainable web applications. It follows the Model-View-Controller (MVC) architectural pattern and includes a powerful Object-Relational Mapping (ORM) system for working with databases.  Django is commonly used in the deployment of Machine Learning (ML) models due to its ability to handle heavy traffic and its support for various data formats. When deploying an ML model, Django can be used to build a web API that provides a simple and secure way to access the model. The API can be used to receive input data, preprocess it if necessary, and pass it to the model for prediction. The output from the model can then be returned to the user in a structured format.  In summary, Django is a powerful web framework that can be used to deploy Machine Learning models through its support for building APIs, managing data security, and integrating with various ML libraries and frameworks.  We created a new project in django with a virtual environment and installed the necessary libraries like joblib, scikit-learn, pandas etc.  We have used two basic templates (html files) for the home page where users can enter personal information which will be used for prediction and a result page which displays the model’s prediction.  **Home Page:-**    Results Page:-      Next step is making the views which will render these html templates.  In Django, a view is a Python function or class that takes a web request and returns a web response. Views are used to handle the logic of a web application and render the appropriate HTML, JSON, or other content to be displayed to the user.    Home function here renders the Home Page.    **Result function: Takes the input data and uses it to predict the output.**      **Transform Function: Makes the transformations required on the input data as per the model.**    **CONCLUSION** |