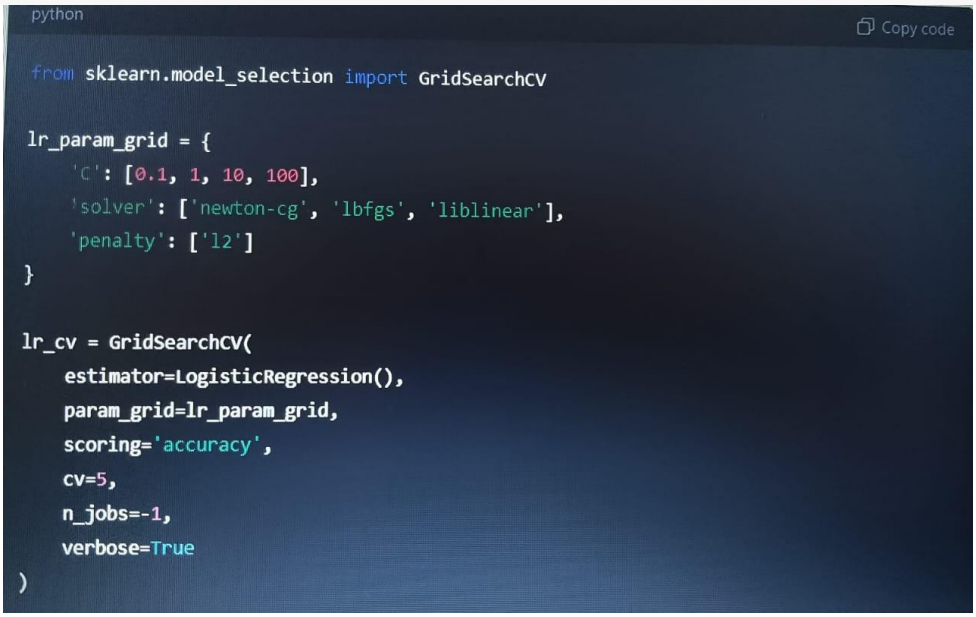
Model Optimization and Tuning Phase Template

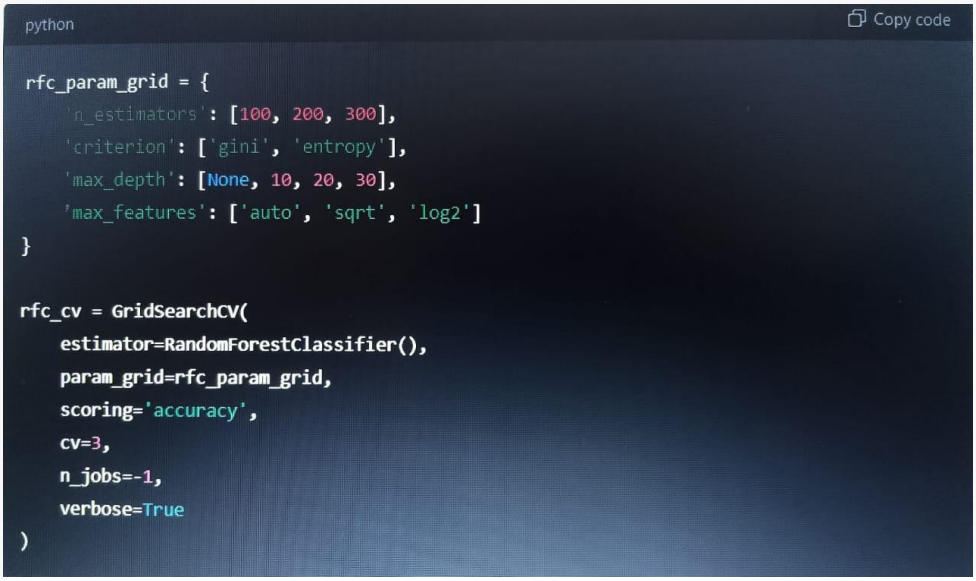
Date	July 2024
Team ID	740772
Project Title	Predictive Modeling for H1-B Visa Approval Using Machine Learning
Maximum Marks	10 Marks

Model Optimization and Tuning Phase for H1-B visa Approval

The Model Optimization and Tuning Phase aims to refine machine learning models to achieve peak performance. This phase involves optimizing model code, fine-tuning hyperparameters, comparing performance metrics, and justifying the final model selection to enhance predictive accuracy and efficiency. For the task of H1-B visa approval prediction, we focus on improving the performance of our selected Recurrent Neural Network (RNN) model.

Hyperparameter Tuning Documentation (8 Marks):

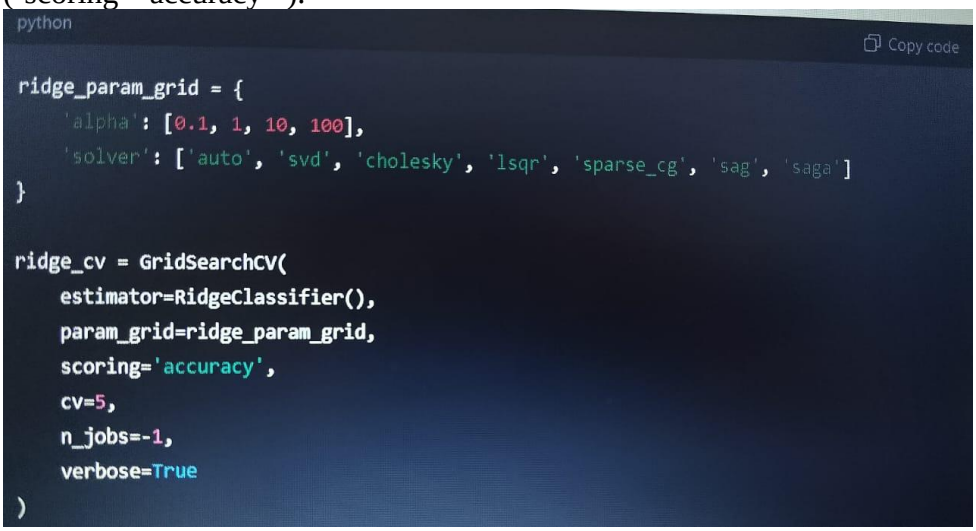
Model	Tuned Hyperparameters
Logistic Regression	The 'lr_param_grid' specifies different values for regularization strength (C), solvers (solver), and penalty types (penalty). GridSearchCV ('lr_cv') is employed with 5-fold cross-validation ('cv=5'), evaluating model performance based on accuracy ('scoring="accuracy"'). The process uses all available CPU cores ('n_jobs=-1') for parallel processing and provides verbose output ('verbose=True') to track progress. 

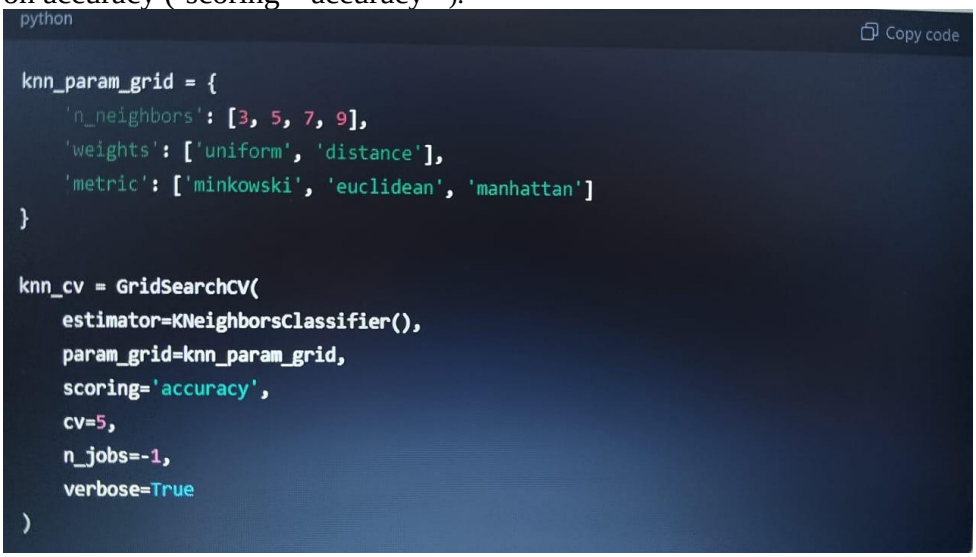
Random Forest	The parameter grid ('rfc_param_grid') specifies different values for the number of trees ('n_estimators'), splitting criterion ('criterion'), maximum depth of trees ('max_depth'), and maximum number of features considered for splitting ('max_features'). GridSearchCV ('rfc_cv') is employed with 3-fold cross-validation ('cv=3'), evaluating model performance based on accuracy ('scoring="accuracy"'). 

```python
rfc_param_grid = {
    'n_estimators': [100, 200, 300],
    'criterion': ['gini', 'entropy'],
    'max_depth': [None, 10, 20, 30],
    'max_features': ['auto', 'sqrt', 'log2']
}

rfc_cv = GridSearchCV(
    estimator=RandomForestClassifier(),
    param_grid=rfc_param_grid,
    scoring='accuracy',
    cv=3,
    n_jobs=-1,
    verbose=True
)
```

XGBoost	The parameters(‘xgb_param_grid’) define a grid for hyperparameter tuning of the XGBoost Classifier (‘XGBClassifier’), including ‘min_child_weight’, ‘gamma’, ‘colsample_bytree’, and ‘max_depth’. The XGBClassifier is configured with a learning rate of 0.5, 100 estimators, using a binary logistic regression objective, and utilizing 3 threads for processing. GridSearchCV (‘xgb_cv’) is used with 5-fold cross-validation (‘cv=5’), refitting the best model (‘refit=True’), evaluating based on accuracy (‘scoring="accuracy"’).
	
	python Copy code
	`from xgboost import XGBClassifier`
	
	`xgb_param_grid = {`
	`'min_child_weight': [1, 5, 10],`
	`'gamma': [0.5, 1, 1.5],`
	`'colsample_bytree': [0.5, 0.7, 1.0],`
	`'max_depth': [3, 4, 5]`
	`}`
	
	`xgb_cv = GridSearchCV(`
	`estimator=XGBClassifier(learning_rate=0.5, n_estimators=100, objective='binary:logisti`
	`param_grid=xgb_param_grid,`
	`scoring='accuracy',`
	`cv=5,`
	`refit=True,`
	`n_jobs=-1,`
	`verbose=True`
	`)`

Decision Tree	The parameters (‘dec_param_grid’) define a grid for hyperparameter tuning of the Decision Tree Classifier (‘DecisionTreeClassifier’), including ‘max_depth’, ‘min_samples_leaf’, and ‘criterion’ ('gini' or 'entropy'). GridSearchCV (‘dec_cv’) is used with 5-fold cross-validation (‘cv=5’), evaluating model performance based on accuracy (‘scoring="accuracy"’). 
Ridge Classifier	The parameters (‘ridge_param_grid’) define a grid for hyperparameter tuning of the Ridge Classifier (‘RidgeClassifier’), including ‘alpha’ and ‘solver’. GridSearchCV (‘ridge_cv’) is used with 5-fold cross-validation (‘cv=5’), evaluating model performance based on accuracy (‘scoring="accuracy"’). 

K- Nearest Neighbors	The parameters (‘knn_param_grid’) define a grid for hyperparameter tuning of the K-Nearest Neighbors Classifier (‘KNeighborsClassifier’), including ‘n_neighbors’, ‘weights’ ('uniform' or 'distance'), and ‘metric’ ('minkowski', 'euclidean', or 'manhattan'). GridSearchCV (knn_cv) is used with 5-fold cross-validation (‘cv=5’), evaluating model performance based on accuracy (‘scoring="accuracy"’). 

Final Model Selection Justification (2 Marks):

Final Model	Reasoning
Random Forest Classifier	The Random Forest model is chosen for its robustness in handling complex datasets and its ability to mitigate overfitting while providing high predictive accuracy. This model leverages the power of ensemble learning by combining the strengths of multiple decision trees,which enhances performance and reliability. The ensemble approach reduces the risk of overfitting, making it particularly well-suited for the the varied and complex nature of H1-B visa approval data. Among all the models evaluated, the Random Forest model demonstrated the highest accuracy, making it the optimal choice for predicting H1-B visa approval outcomes.