ASSIGNMENT:2

REPORT

1. Introduction

The objective of the project is to forecast whether a student is likely to be hired (placed) through campus placements based on the features available in the provided dataset.

Campus placements are important since they directly influence:

The career lives of students.

The reputation of the educational institution.

The attractiveness of the institution to potential applicants.

Here, we implement and train different machine learning models in order to predict placement scores. We also combine them in ensembling (Voting Classifier) form for improved performance.

2. Dataset Selection

Dataset used:

train.csv: training data to build and adjust the models.

test.csv: test data where we need to predict placement outcomes.

SampleSubmission.csv: template for submitting test predictions.

Target variable:

status:

Placed → student was recruited.

Not Placed → student was not recruited.

Dataset size and source:

Reasonable size for basic classification task.

Provided by instructor (presumably suitable and reliable).

3. Data Preprocessing

3.1 Exploratory Data Analysis (EDA)

EDA helps us determine the data structure and quality.

Used .head(), .info(), .describe() to check for data type and missing values.

Plotted the target variable (status) distribution using a countplot → this helps us check if there is class imbalance.

Used a correlation heatmap to check if there are strongly correlated features.

3.2 Handling Missing Values

Missing values existed in certain of the features.

Missing numerical values were imputed with the median of that feature.

Median is not affected by outliers and works well in most cases.

3.3 Categorical Variable Encoding

Categorical variables need to be numeric for use in machine learning algorithms, so LabelEncoder encoded the categorical variables.

Target variable status was encoded as follows:

Placed → 1

Not Placed → 0

3.4 Data Split

Data was split into:

70% training set → to train the models.

30% validation set → to test model performance.

The split was stratified so that the proportion of placed/not placed students in each set was identical.

4. Model Selection

We selected 3 models for comparison:

1. Logistic Regression

Interpretable linear model in simple form.

Good as a baseline for binary classification.

2. Random Forest Classifier

Ensemble model made of lots of decision trees.

Can be used to capture complex non-linear relationships.

Hyperparameters were tuned using GridSearchCV.

3. Support Vector Classifier (SVM)

Works well in high-dimensional space.

Good for binary classification.

Was set up with probability=True in order to allow soft voting.

Hyperparameters were tuned (Random Forest):

n_estimators: number of trees.

max_depth: max depth of each tree.

min_samples_split: minimum number of samples to split a node

4. Model Selection

We decided to compare 3 models:

1. Logistic Regression

Simple interpretable linear model.

Good to use as a baseline for binary classification.

2. Random Forest Classifier

Ensemble model made up of many decision trees.

Can be used to model complex non-linear relationships.

Hyperparameters were tuned using GridSearchCV.

3. Support Vector Classifier (SVM)

Does well in high-dimensional space.

Good to use for binary classification.

Was started with probability=True so that it could do soft voting.

Hyperparameters were tuned (Random Forest):

n_estimators: quantity of trees.

max_depth: maximum depth of the trees.

min_samples_split: minimum samples required to split a node

5. Model Training

Trained each model using the training set.

Trained models with code organization and documentation that was good.

Training steps included:

Scaling features where it was required.

Fitting the models to the data.

Saving and plotting the results.

Metrics used:

We evaluated each model on the basis of:

Accuracy → % of correct predictions.

Precision → when the model predicts "Placed", how often is it correct?

Recall → of the actually "Placed" students, how many did the model predict correctly?

F1 Score → harmonic average of Precision and Recall → balances the two.

Confusion Matrices:

Plotted confusion matrices for each model to visualize True Positives, False Positives, etc.

Model Comparison:

Model	Accuracy	Precision	Recall	F1 Score
Logistic Regression	0.7538	0.7959	0.8667	0.8298
Random Forest	0.9385	0.9362	0.9778	0.9565
SVM	0.6923	0.6923	1.0000	0.8182
Voting Classifier	0.8462	0.8302	0.9778	0.8980

Observations:

The voting classifier has given balanced performance.

Random Forest gave very good performance when optimized.

SVM provided good Recall and F1 Score values.

Logistic Regression was simple but performed very well nevertheless.

Random Forest is the best performing of all:

Best Accuracy: 93.85% → very few total errors.

Best F1 Score: 0.9565 → very good at balancing Precision and Recall.

Both Precision and Recall are gravy-high; it means that it not only captures most of the placed students but is also highly accurate in predicting the placement of these students.

Voting Classifier also performs very well:

84.62% Accuracy is lower than Random Forest but still very good.

Very High Recall (0.9778) → it basically captures all the placed students like Random Forest.

Good Precision (0.8302) and F1 Score (0.8980) though slightly less than Random Forest but superior to Logistic Regression and SVM.

Voting Classifier benefits from the strengths of the models used in its favor.

Logistic Regression as a good baseline:

Accuracy 75.38% → decent performance of a simple model.

F1 Score 0.8298 → good come-off though not as strong as the one provided by Random Forest or

Voting Classifier.

Good for Precision and Recall but clearly beaten by Random Forest.

SVM behaves strangely with:

Perfect Recall (1.0) → placed all the students correctly (no false negatives).

But Precision was low (0.6923) because of so many false positives → tends to over-estimate "Placed".

So, it ended up with low Accuracy and a lower F1 Score than Random Forest or Voting Classifier.

This, in practice, will see it marking numerous students as "Placed" when they are not.