

## Model Development Phase Template

Date	31 June 2024
Team ID	739762
Project Title	Software Salary Prediction
Maximum Marks	4 Marks

### Initial Model Training Code, Model Validation and Evaluation Report

Initial Model Training Code, Model Validation and Evaluation Report The initial model training code will be showcased in the future through a screenshot. The model validation and evaluation report will include classification reports, accuracy, and confusion matrices for multiple models, presented through respective screenshots

### Linear Regression Model

```
[34] reg= LinearRegression()  
reg.fit(X_train,y_train)
```

```
LinearRegression  
LinearRegression()
```

```
[35] y_test_pred=reg.predict(X_test)  
y_train_pred=reg.predict(X_train)
```

```
train_r2 = r2_score(y_train, y_train_pred) * 100  
print("Training R²",train_r2)  
  
# Calculate and print the R² score for the testing data  
test_r2 = r2_score(y_test, y_test_pred) * 100  
print("Testing R²: ",test_r2)
```

```
Training R² 86.03987604146623
```

### Random forest Model

```
rfr = RandomForestRegressor(n_estimators=100,random_state=42)  
rfr.fit(X_train,y_train)  
y_test_pred=rfr.predict(X_test)  
y_train_pred=rfr.predict(X_train)
```

```
train_r2 = r2_score(y_train, y_train_pred) * 100  
print("Training R²:",train_r2)  
#Calculate and print the R² score for the testing data  
test_r2 = r2_score(y_test, y_test_pred) * 100  
print("Testing R²",test_r2)
```

```
Training R²: 86.03987604146623  
Testing R² 0.19943667460349257
```

```
[31] xg_reg = xgb.XGBRegressor()  
xg_reg.fit(X_train,y_train)
```



XGBRegressor

```
XGBRegressor(base_score=None, booster=None, callbacks=None,  
             colsample_bylevel=None, colsample_bynode=None,  
             colsample_bytrees=None, device=None, early_stopping_rounds=None,  
             enable_categorical=False, eval_metric=None, feature_types=None,  
             gamma=None, grow_policy=None, importance_type=None,  
             interaction_constraints=None, learning_rate=None, max_bin=None,  
             max_cat_threshold=None, max_cat_to_onehot=None,  
             max_delta_step=None, max_depth=None, max_leaves=None,  
             min_child_weight=None, missing=None, monotone_constraints=None,  
             multi_strategy=None, n_estimators=None, n_jobs=None,  
             num_parallel_tree=None, random_state=None, ...)
```

```
[32] y_test_pred=xg_reg.predict(X_test)  
y_train_pred=xg_reg.predict(X_train)
```

```
[33]  
train_r2 = r2_score(y_train, y_train_pred) * 100  
print("Training R²:",train_r2)  
#Calculate and print the R² score for the testing data  
test_r2 = r2_score(y_test, y_test_pred) * 100  
print("Testing R²: ",test_r2)
```

## Decision tree for training data

```
▶ y_train_pred = dtr.predict(X_train)
  y_test_pred = dtr.predict(X_test)

  r2_score(y_train, y_train_pred)*100
```

```
⇒ 99.88283394123113
```

## Model Validation and Evaluation Report:

Model	F1 Scor e
Random Forest	86%
Decision Tree	83%
KNN	64%
Gradient Boosting	78%