1. **Software language and libraries used -**

   R was primarily used for the data analysis and model creation. I used different libraries like 'rpart' for decision tree, 'randomForest', 'caret', 'ggplot' and 'rattle'.

2. **Data Preparation and Cleaning -**
   - I first wrote a script in R to merge the predictor and the features from 'train_features_DATE.csv' and 'train_salaries_DATE.csv' based on **'jobId'**.
   - Then I converted the categorical features into numeric factors based on the number of categories levels, as they could not be used in the linear regression model prediction.
   - There were a couple of instances with salary as 'zero' value, which were the outliers that needed to be removed.
   - 'Major' feature had almost 85% of the values as NONE, so it wasn't used as a feature for building our model.

3. **Algorithmic Methods -**
   - I started initially by dividing the data randomly into training and testing from the 'train_features_DATE.csv' into 70:30 ratio.
   - I applied RandomForest(ntree=100, mtry=2) on the training data. The accuracy seemed to increase with increasing number of trees=500 with given more time to build the model.
   - Other methods that I considered were Linear Regression that was used to find the base accuracy without any complex model use initially. Moreover, Decision tree was also used as a comparative analysis with RandomForest output.

4. **Feature Selection -**

   By finding the correlation matrix between the features and finding the %Inc In MSE (Applying RandomForest) we could get the most important features. Also, to cross verify I applied decision tree model and found the variables with highest 'Information Gain'. The finalized features were:

   > **yearsExperience**
   > **milesFromMetropolis**
   > **jobTypeFactor** (Job Type converted to numeric values)
   > **degreeFact** (degree converted to numeric values)
   > **industryFact** (industry converted to numeric values)

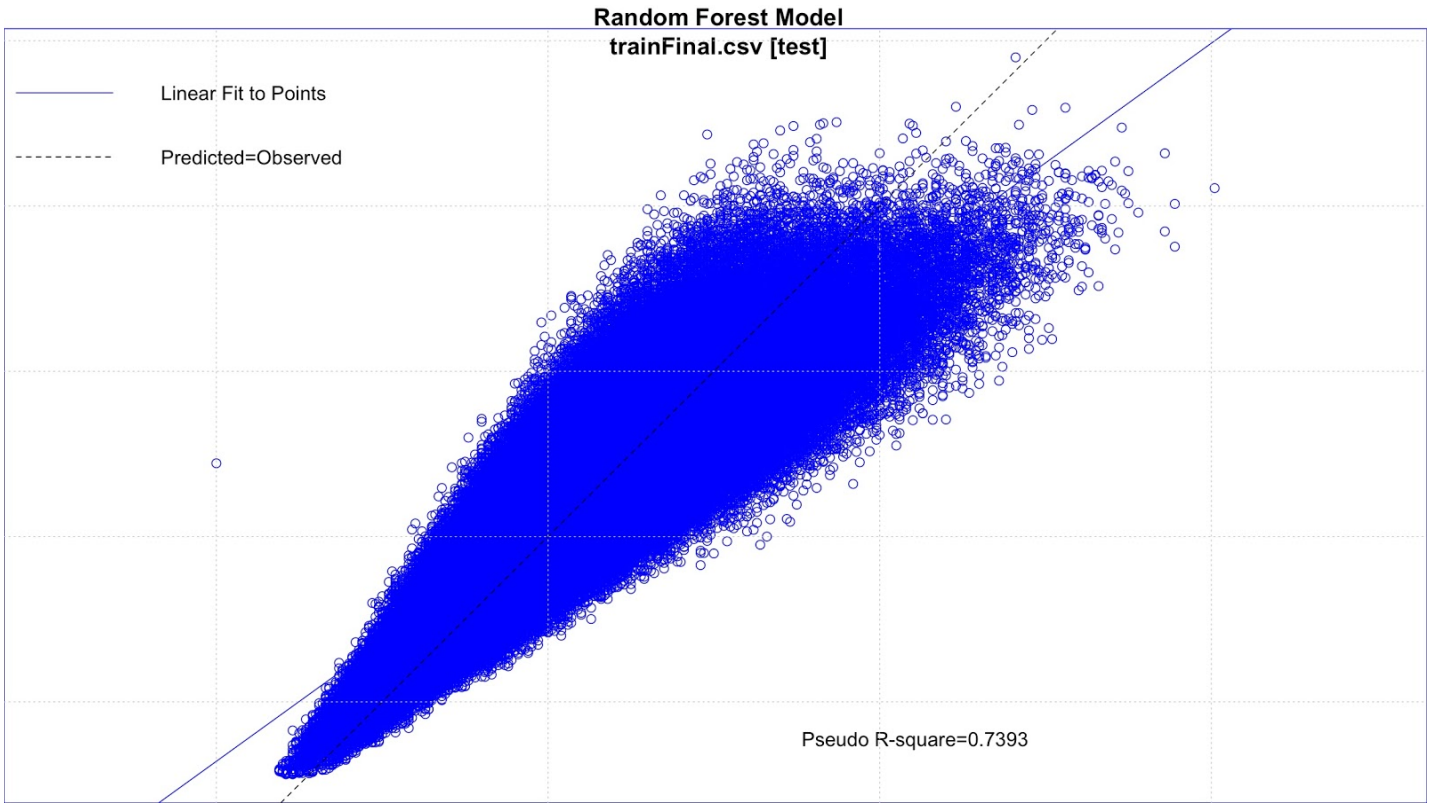
   Feature like 'jobId' were ignored which were used to identify the instances uniquely' and 'companyId' which had a lot of categories which were not correlated with the target variable.

5. **Accuracy and Evaluation criteria**

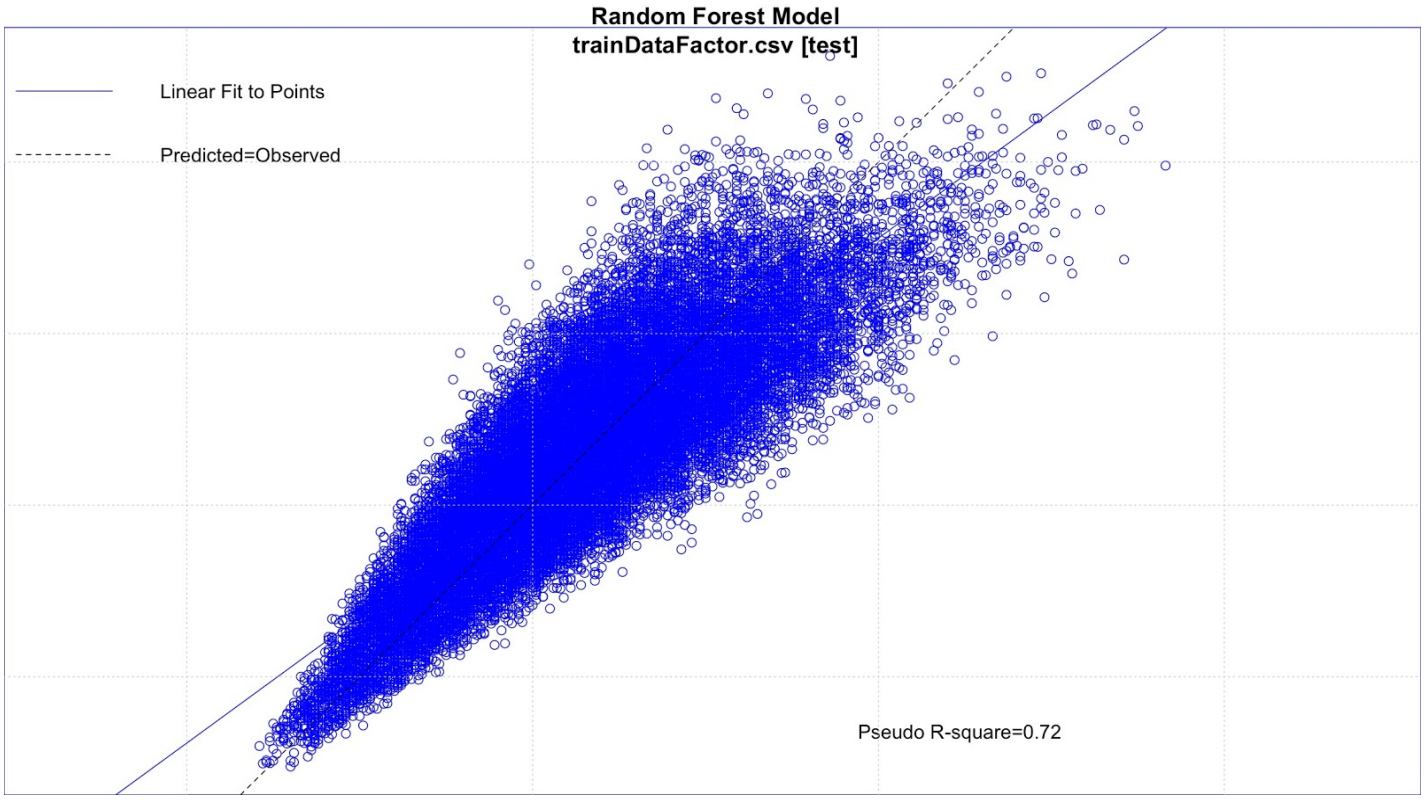
The accuracy was calculated using the mean using **Pseudo R Squared** value which is a metric that tries to mimic the R-Squared value. It is calculated as the square of correlation between predicted and observed value. It is better when it is closer to 1, but should not fit the data input. (R- square is the percent of response variation as explained by linear model)

Model			Pseudo R Square Value
Random Forest	10% random sampled data	Ntree = 500, mtry = 3, time taken = 23.97 min	0.717
		Ntree = 100, mtry = 2, time taken = 3.11 min	0.725
	Complete training data	**Ntree = 100, mtry = 2, time taken = 5.34 hrs**	**0.7393**
Linear Regression(Anova)	Training data	Residual err=31.5 on 70%data	0.347
Decision Tree	Training data	Bucket size = 20 Min split = 7	0.478

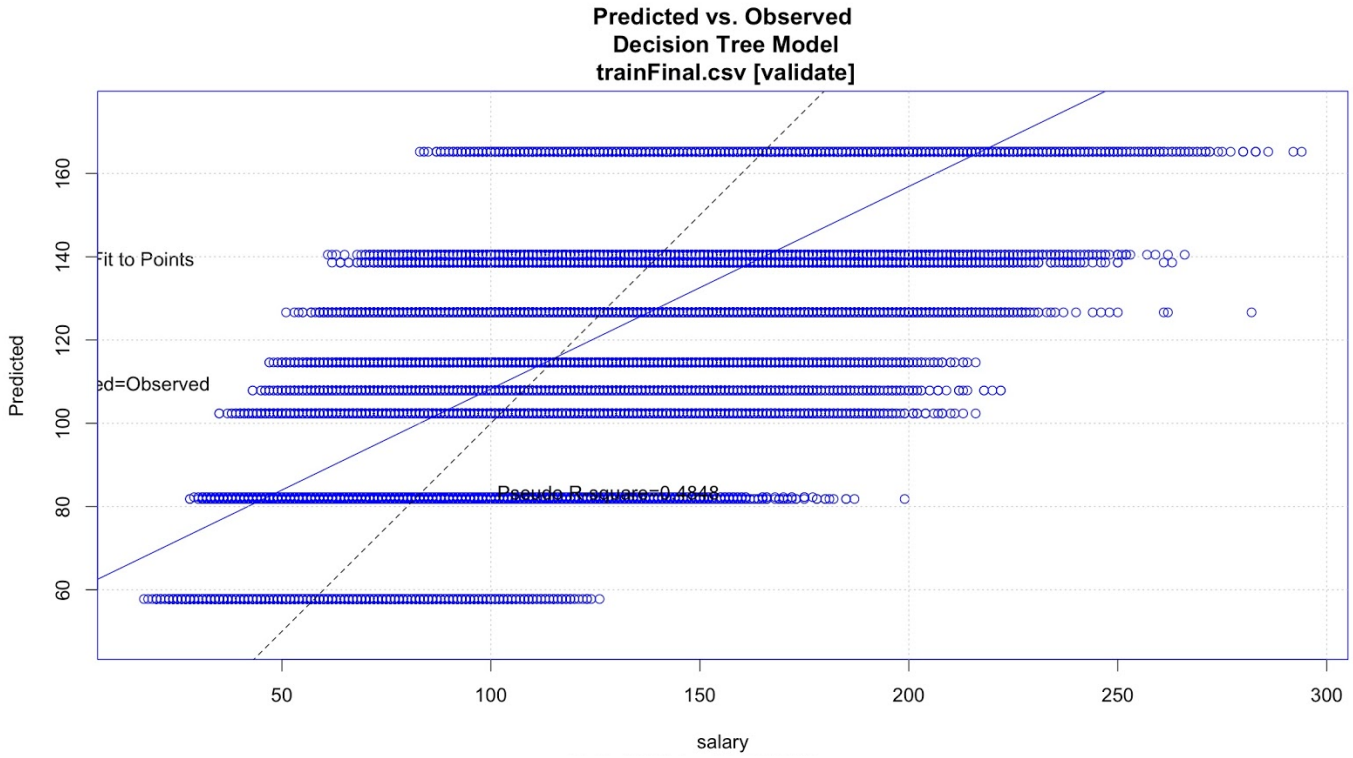
Random Forest Evaluation(complete data):-



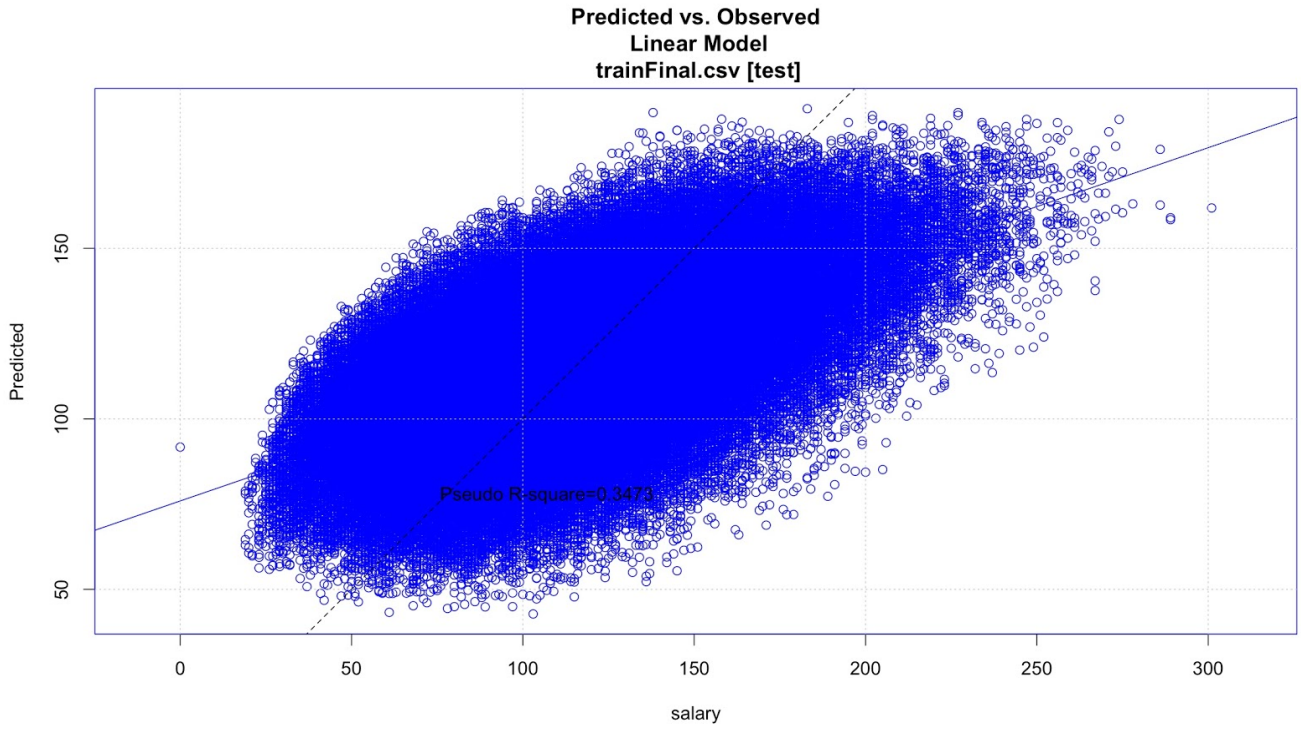
Random Forest Evaluation(10% data):-



Decision Tree Evaluation:-



Linear Regression Evaluation:-



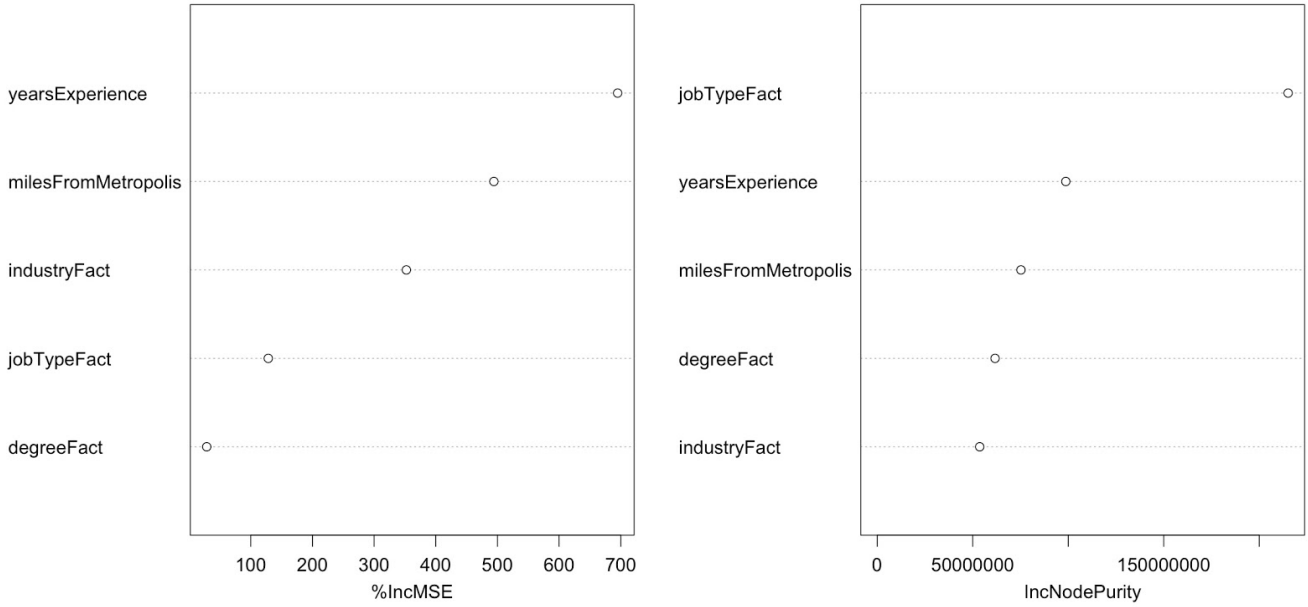
6. **Which features had the greatest impact on salary? How did you identify these to be most significant? Which features had the least impact on salary? How did you identify these?**

Features having most impact on salary were:

> yearsExperience
> milesFromMetropolis
> JobType
> industry
> degree

I calculated the above by finding the important variable plot by applying random forest on complete dataset and Finding the top variables used in the decision tree model which uses information gain ratio to calculate the significant features.

Variable Importance Random Forest trainFinal.csv



Features having low impact on salary were:

> Major (as more than 50% of the values were unknown)

> jobId

> companyId (uncorrelated with other features and target)