Data Modeling

Data Science II (COSC 4337)

Dr. Ricardo Vilalta

Submitted by:

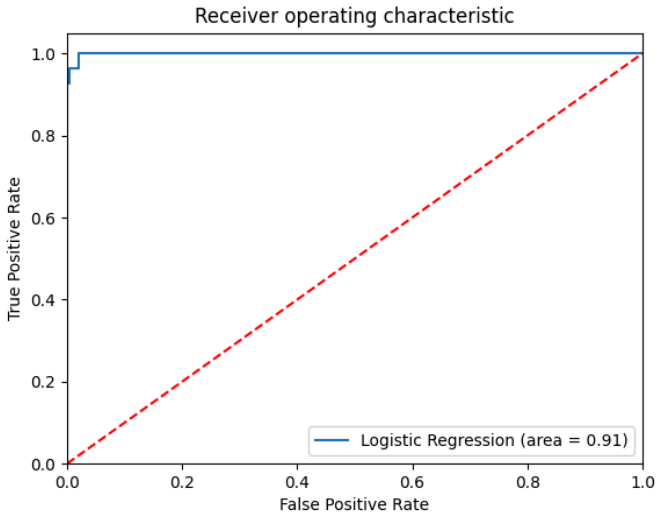
Brayan Gutierrez (1865588)

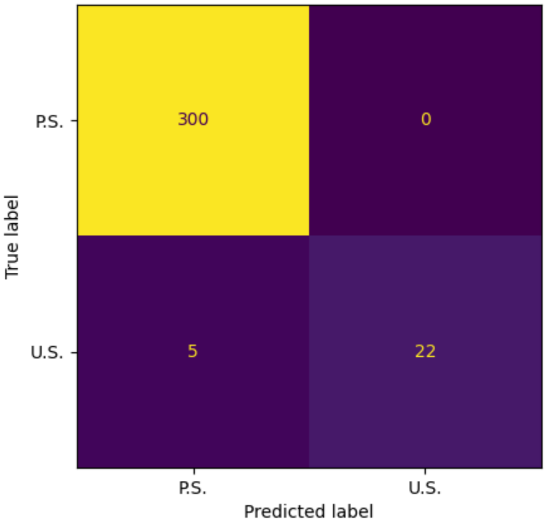
Jericka Ledezma (1968730)

Katie To (2026435)

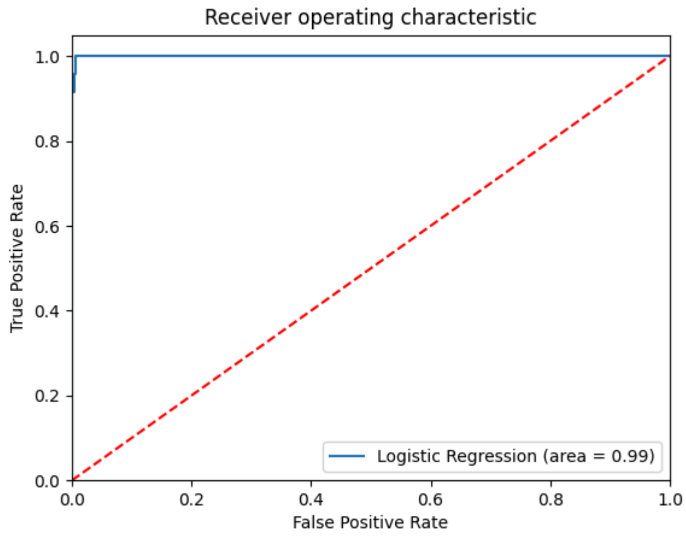
For this deliverable, each member of our team opted to do one supervised learning method. As our data already contained labels, we conducted supervised learning methods over unsupervised methods. In deliverable 1, we determined that the following features were to be kept: RSC, SAR, Na, E.C, TDS, HCO3, pH, Mandal, and Village, so these were the only features used in the following models.

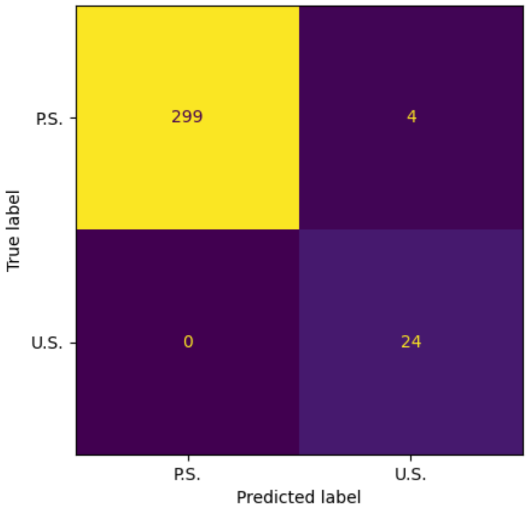
The first model was created with the logistic regression method by Katie. The first step taken was to scale the data; this was done to solve a “TOTAL NO. of ITERATIONS REACHED LIMIT” error that would occur when starting to fit the data to the model. Logistic regression, specifically the ROC curve, could not be done without changing all the features into numeric values, so a combination of label encoding and one hot encoding was done to achieve this. Label encoding was always going to be used on Classification.1 in order to keep the labels under one column since logistic regression only accepted one column for the target variable, however the same error from earlier would occur when using label encoding on the mandal and villages features. As such, one hot encoding was used to transform mandal and villages into numeric features. When using dummy variables for village and mandal and no labeling for Classification.1, the accuracy averaged to be about 99% with no AUC score as the Classification.1 feature was not made numeric yet. When using dummy variables for the village and mandal features then label encoding classification,1, the accuracy fell to 98% but returned an AUC score of 0.91.



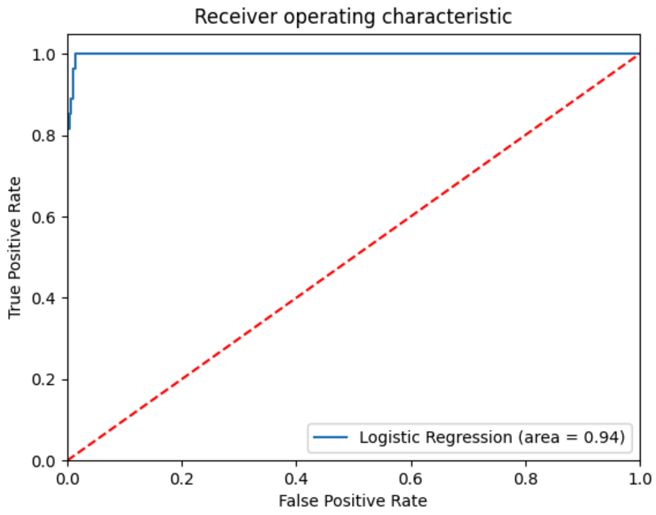


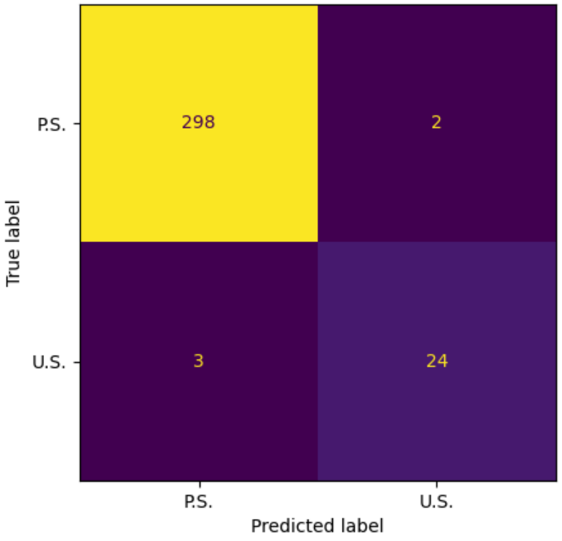
However, 0.91 was found to be an unsatisfactory AUC score, so further tuning was done. The ratio between P.S. (safe water) and U.S. (unsafe water) appeared to be disproportionate, with 750 P.S. values and 67 U.S. values total. Even in the confusion matrix above that contains 40% of the values, it is extremely disproportionate. To address this imbalance, SMOTE was used to oversample the minority class, then fit the model with this new dataset. The accuracy with resampling and both encoding methods used increased to 99% and the AUC score increased to 0.99.





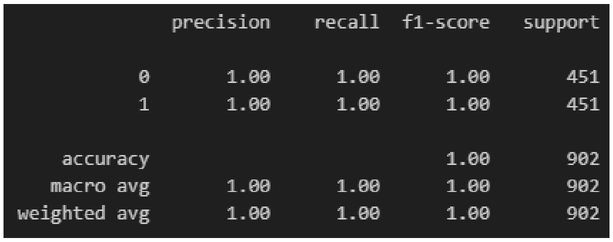
These results were satisfactory, but further tuning was done in order to see if it could be improved. Although logistic regression doesn't have many critical hyperparameters to finetune, a grid search was used to find the best solver, penalty, and c value. The grid returned the following results: "0.998881 using {'C': 100, 'penalty': 'l2', 'solver': 'newton-cg'}", and another model was made using these hyperparameters. The accuracy dropped to 98%, while the AUC dropped to 0.94.



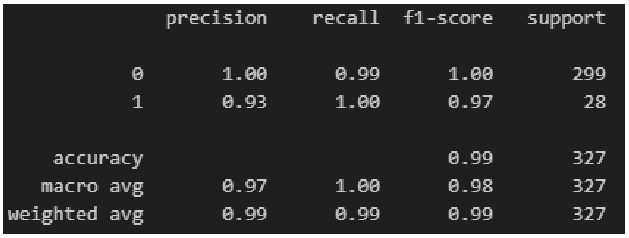


As both the AUC score and accuracy fell after changing the parameters, using a model fitted with just resampled data and both encoding labels would be best if we went with a logistic regression algorithm.

Regarding bias-variance trade off, we will only examine the model we are most likely to use, as we have already found which parameters and pre-processing methods are optimal.



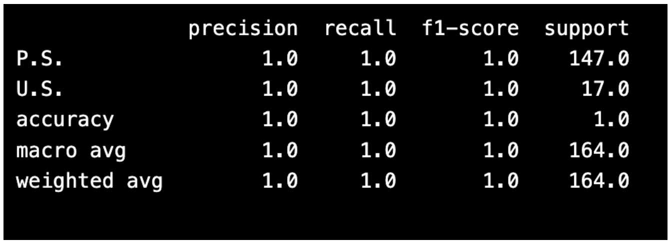
```
              precision    recall  f1-score   support

           0       1.00      1.00      1.00       451
           1       1.00      1.00      1.00       451

    accuracy                           1.00       902
   macro avg       1.00      1.00      1.00       902
weighted avg       1.00      1.00      1.00       902
```



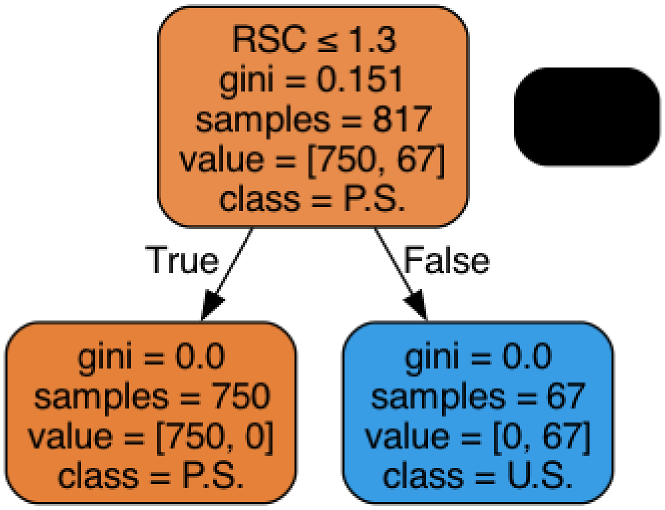
```
              precision    recall  f1-score   support

           0       1.00      0.99      1.00       299
           1       0.93      1.00      0.97        28

    accuracy                           0.99       327
   macro avg       0.97      1.00      0.98       327
weighted avg       0.99      0.99      0.99       327
```

To the left, we see the classification chart resulting from the training set in model two, while the right shows the classification chart from the test set. This model has a low score in precision and a middling f1-score when it comes to classifying U.S. objects in the test set, but otherwise it has high scores across the board when it comes to P.S. classification. As there's similar scores in both sets we can say that the data is generally being fitted pretty well and can make good predictions on unseen data. This is additionally an indication of a balanced variance and bias dynamic despite the fact most logistic regression models are high bias-low variance.

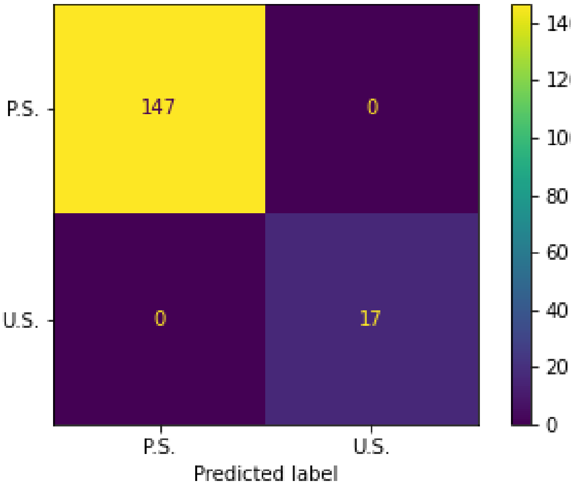
To get a different perspective on the data, Jericka was in charge of creating the second model: the decision tree. The groundwater dataset was loaded up in a new jupyter notebook. The 'marginal safe' classification is replaced with 'unsafe' using the pandas operation to keep the data between only 2 classifications. The code splits the dataset into features and target variables, and then converts categorical variables to numerical representations using one-hot encoding. This preprocessing is important for decision tree models as it enables better data splitting at each node, leading to more accurate and interpretable decision trees. During outlier treatment as part of the data preprocessing step, extreme values in numerical columns are identified using the IQR method and replaced with upper or lower whisker values based on a defined threshold. This step helps improve the dataset's reliability and ensures a more accurate statistical analysis. The decision tree is finally created with different tree depths (3, 7, 11, 15) with the training data, and the accuracy scores for each depth are reported as 100%. Similar results were found at each depth using the testing data. This suggests that the model performs well with unseen data and was not overfitted.



```
              precision  recall  f1-score  support
P.S.                1.0     1.0       1.0    147.0
U.S.                1.0     1.0       1.0     17.0
accuracy            1.0     1.0       1.0      1.0
macro avg           1.0     1.0       1.0    164.0
weighted avg        1.0     1.0       1.0    164.0
```



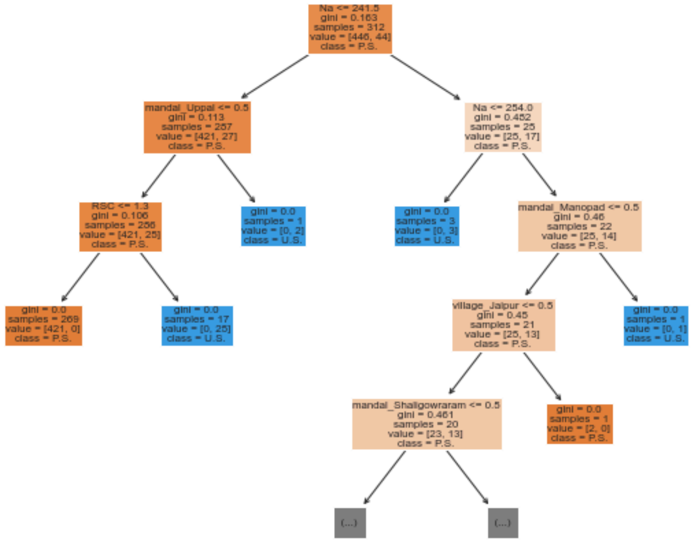
[redacted]

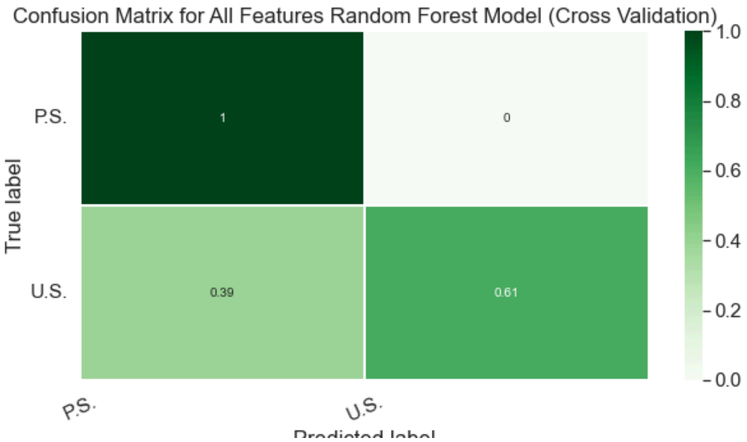


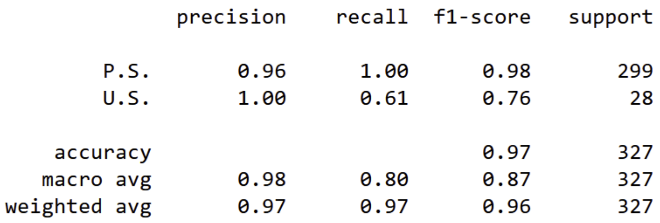
Evaluation metrics, starting with a confusion matrix, were employed to confirm the initial findings in the decision tree model using the testing data. The confusion matrix results revealed accurate predictions for 147 instances of the safe class (P.S.) and 17 instances of the unsafe class (U.S.), showcasing a scenario of perfect performance by the decision tree model. This kind of prediction suggests that the model performs very well with unseen data, leading to little to no challenges in generalizing well to unseen data. To confirm the model's performance from the confusion matrix, f-1 scoring was used as the second evaluation matrix. Using the classification report from the sklearn metrics library the f-1 report came back with the perfect score reaffirming the model's performance. However, since there is a good chance of high variance with a single decision tree we decided to look elsewhere for a data model that more accurately described the groundwater dataset.

To address the high variance issue with a single decision tree, the third model was the random forest model created by Brayan. The initial step involved reading the data into a working dataframe. After creating the data frame, outliers were treated by capping at the 75th percentile and flooring at the 25th percentile to avoid potential issues with the random forest model. Subsequently, a second data frame containing the selected features was created. Since random forest models cannot handle categorical string variables, such as Mandal and Village in our case, these features were encoded using the "get_dummies" function built into pandas. A third dataframe containing the dummy variables was then created and used throughout the rest of the model. A seed of 10 was set for the entire model. The data was split into 60% training and 40% testing, and this split remained consistent for the rest of the model.

The random forest model was evaluated in two ways: one using all of the selected features and the other using only the selected numerical features. Observing the random forest models with all of the selected features, two different variants were created. The first variant involved cross-validating the number of estimators and the maximum depth of the tree. The resulting model had a maximum depth of 17 and 83 estimators, as indicated by the tuned hyperparameters in the following tree snippets and confusion matrix.



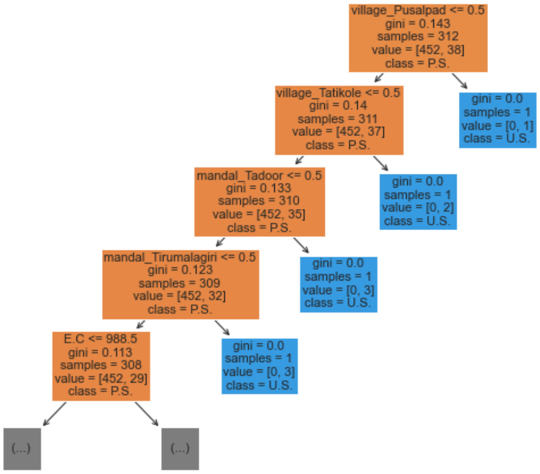


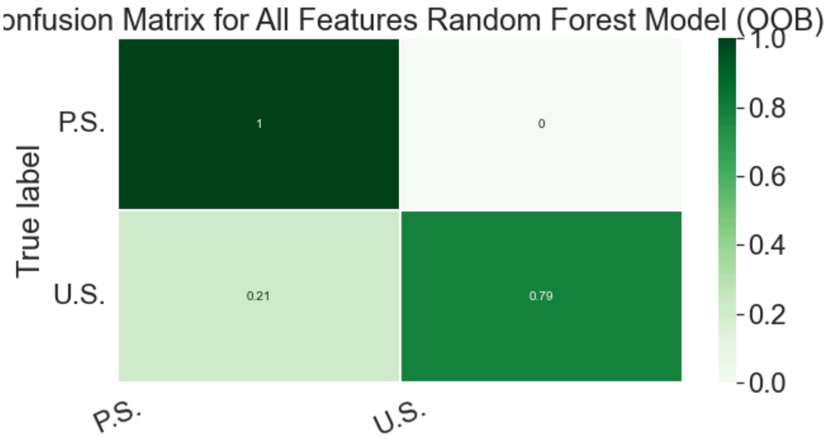


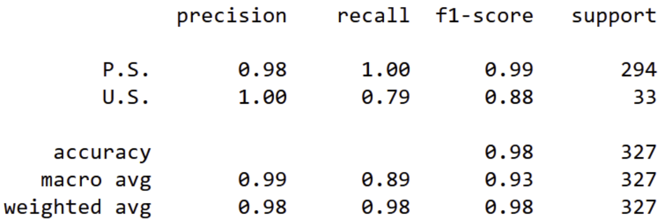
```
              precision    recall  f1-score   support

        P.S.       0.96      1.00      0.98       299
        U.S.       1.00      0.61      0.76        28

    accuracy                           0.97       327
   macro avg       0.98      0.80      0.87       327
weighted avg       0.97      0.97      0.96       327
```

Based on the confusion matrix and the accuracy score obtained from the accuracy_score function of sci-kit learn, this variant of the random forest model achieved an accuracy of 96.6% on the testing set. Although this is a good initial result, it is anticipated that further improvements will be seen with other variants. The F1 scores reveal that this variant of the model performs better at predicting the safe (P.S.) class compared to the unsafe (U.S.) class, potentially due to the significantly higher number of P.S. observations in the dataset.

The second variant of the model used the out-of-bag parameter set to true in the hope of achieving better performance compared to the first variant. The number of estimators was left at its default value, but the maximum depth was set to 4. Below is a snippet of one of the fitted decision trees, along with the confusion matrix and classification report for the testing data







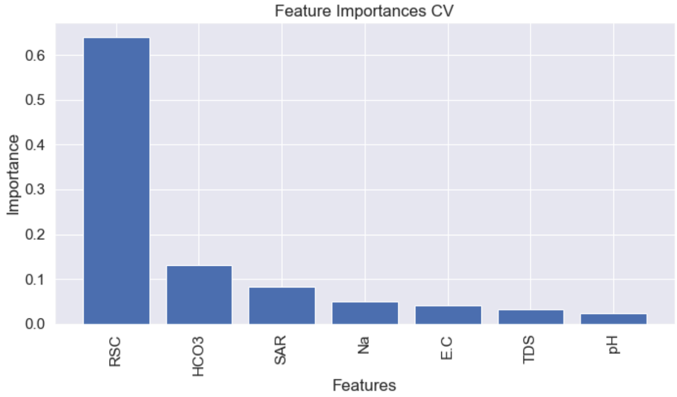
```
              precision    recall  f1-score   support

        P.S.       0.98      1.00      0.99       294
        U.S.       1.00      0.79      0.88        33

    accuracy                           0.98       327
   macro avg       0.99      0.89      0.93       327
weighted avg       0.98      0.98      0.98       327
```

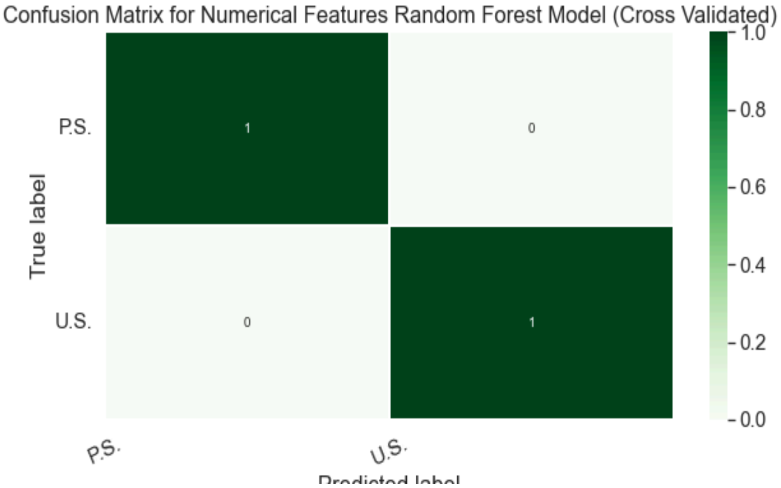
This second variant of the random forest model achieved a score of 97.9% on the testing set, a slight improvement over the previous variant. However, it was expected that the out-of-bag model would perform better compared to the other model variants. Similar to the previous variant, this model predicts P.S. classes better than U.S. classes, although in this case, it predicts U.S. classes 10% better than the last two variants based on the displayed F1 score.

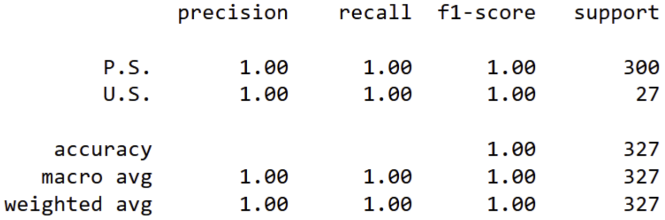
In both variants of the random forest model using all of the selected features, it is observed that most of the features used in the decision trees came from the Mandal and Village features, with little use of the other selected features. This suggests that the dummy variables for Mandal and Village are more important, as they were used for most of the splits, compared to the other numerical features. Additionally, in both variants, the performance in predicting U.S. classes was slightly lower than that for P.S. classes, although both variants still performed well with accuracies between 96% and 98%.

To assess the performance of the numerical features without the Mandal and Village features, two additional variants were created. These models were created in the same manner as the previous two. Due to space constraints, decision trees will not be displayed in these variants; instead, feature importance plots will be shown.

Similar to the all-features model, the first variant involved cross-validating the number of estimators and the maximum depth of the tree. The feature importance plots and confusion matrix indicate that the tuned hyperparameters for this variant were a maximum depth of 14 and 373 estimators.

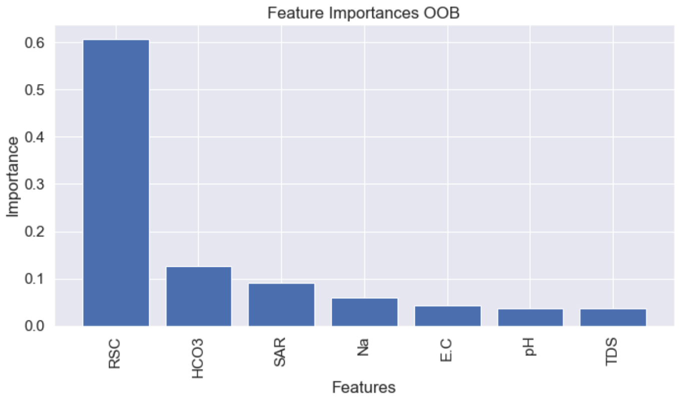


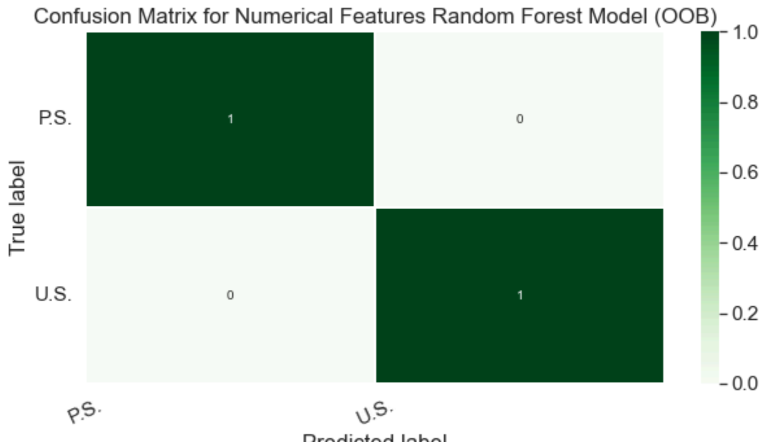


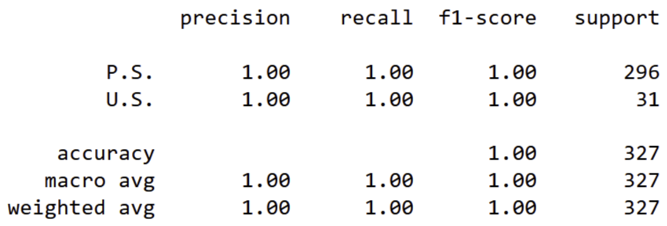

```
              precision    recall  f1-score   support

        P.S.       1.00      1.00      1.00       300
        U.S.       1.00      1.00      1.00        27

    accuracy                           1.00       327
   macro avg       1.00      1.00      1.00       327
weighted avg       1.00      1.00      1.00       327
```

Based on the confusion matrix and accuracy score, this first variant of the random forest model using only numerical features achieved a perfect score of 100% on the testing set. This represents a significant improvement over the previous all-features variants. This model outperformed the out-of-bag model for all features. The higher performance of this model may be attributed to the fact that the numerical features scored higher in the feature selection tests during the pre-processing stage of the project. Among all numerical features, RSC was the most important, consistent with the feature selection plots in the previous assignment, with HCO3 and SAR scoring second and third highest, respectively. Unlike the previous variants, this model predicts both P.S. and U.S. classes with 100% success.

Like the all-features models, the second variant of the numerical features model used the out-of-bag parameter set to true in the hope of achieving better performance compared to all previous variants. Below are the importance plot, confusion matrix, and classification report for the testing data.





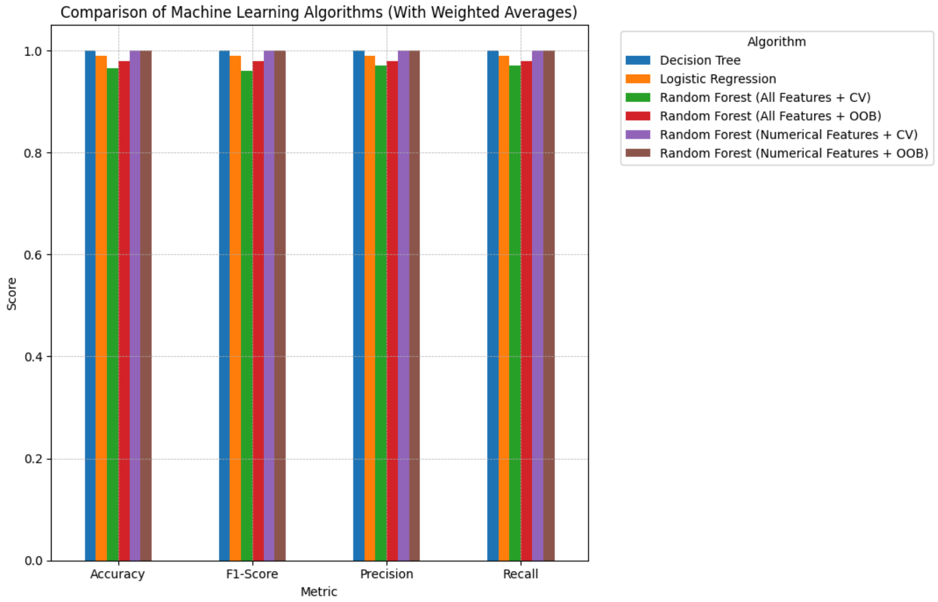


```
              precision    recall  f1-score   support

        P.S.       1.00      1.00      1.00       296
        U.S.       1.00      1.00      1.00        31

    accuracy                           1.00       327
   macro avg       1.00      1.00      1.00       327
weighted avg       1.00      1.00      1.00       327
```

This variant of the numerical features model performed equally well as the previous variant, achieving an accuracy of 100%. Similar to the previous variant, the out-of-bag variant of the model predicts both P.S. and U.S. classes with 100% success and has the exact same importance plot. The only difference between the two is the distribution between P.S. and U.S. predictions.

It is observed that both models consisting only of the selected numerical features performed significantly better on the testing data than both models consisting of all selected features across the board.

Turning to the bias-variance tradeoff, we'll evaluate each model variant using precision and recall. Beginning with the all-features cross-validation model, we observe comparatively low precision but high recall for the P.S. class and high precision but low recall for the U.S. class. This suggests high variance for the U.S. class, indicated by the tradeoff between precision and recall. Similarly, the P.S. class may exhibit high variance due to an increased number of false positives. One potential explanation is slight overfitting to the training data, leading to misclassification of some unseen testing data. However, this overfitting might not significantly impact accuracy, which remains high. Moving to the all-features Out-of-Bag (OOB) model, it shares the same issues as the cross-validated model: low precision but high recall for P.S. and high precision but low recall for U.S. Finally, examining the last two numerical random forest models, we find that both achieve 100% precision and recall. This indicates low bias and low variance, suggesting they can correctly classify any unseen data not included in the testing data set. The high F1 scores of 100% support this conclusion. If a choice between random forest variants is required, the numerical OOB variant is preferable. Previous evidence suggests that OOB models tend to outperform regular random forest models, and the numerical OOB variant exhibits optimal performance across accuracy, precision, recall, and F1 score metrics. One thing to note is that the scores for the all-feature models may change, as expected with the random forest model and the changing of training and testing sets. However, the same conclusion will be reached.



In comparing the logistic regression, decision tree, and random forest models developed by Katie, Jericka, and Brayan respectively, it becomes clear that each model has its strengths and weaknesses. Katie's logistic regression model initially achieved high accuracy of 99% but struggled with an unsatisfactory AUC score, which prompted further tuning. Despite optimization efforts, the model's performance slightly declined, showing limitations in improving its predictive power beyond a certain threshold. Jericka's decision tree model exhibited perfect performance in predicting both safe and unsafe classes, suggesting excellent performance to unseen data. However, concerns about high variance lead to exploration of alternative models. Brayan's random forest models, specifically those focusing solely on numerical features, showed exceptional performance with 100% accuracy, precision, recall, and F1 scores. These models exhibited low bias and variance, suggesting robustness in handling unseen data. Considering the comprehensive evaluation metrics and performance consistency, the numerical out-of-bag random forest variant emerges as the most favorable choice, offering optimal predictive capability and generalization ability for the groundwater dataset. Have all modules installed for the notebook to work.