Predicting Purchase Behavior

Take your marketing to the next level, keep your customers coming back, and help your business grow.

By: Erica Yanoshak

Importance of Predicting Customer Purchase Behavior

- **Deep Understanding of Customer Behavior:** Predicting customer behavior helps businesses understand preferences, habits, and decision-making processes.
- **Personalized Marketing and Experiences:** Predictive analytics enables businesses to create tailored marketing strategies, delivering personalized customer experiences.
- **Improved Customer Acquisition and Retention:** By anticipating customer needs, businesses can engage, retain, and build loyalty among their customers more effectively.
- **Competitive Advantage:** Predicting buying behavior helps identify emerging trends and market opportunities, giving businesses an edge over competitors.
- **Enhanced Growth and Customer Satisfaction:** Accurate predictions allow businesses to refine strategies, optimize the customer journey, and drive growth through improved customer satisfaction.

Customer Purchase Prediction Model Plus Feature Importance

- **Problem to Solve:** Develop a predictive model to determine the likelihood of a customer making a purchase based on historical behavior and other demographic data.
  - **Model Selection:** supervised learning approach, logistic regression, decision trees, or random forest, to predict whether a customer will make a purchase.
  - **Evaluation:** Using metrics like accuracy, precision, recall, F1-score, and ROC-AUC to evaluate model performance.
  - **Optimization:** Through random search, SMOTE, and stacking models
- Analyze the features to learn what does and does not lead, to customers making a purchase
  - **Evaluation:** Using feature importance, identify how each variable impacts potential purchase behavior
  - **Technologies:** Scikit-learn for the model, Pandas and Matplotlib for data handling and visualization, and Python for scripting.

Data Preprocess

Features		
Age	*Customer's age*	20 - 70 yrs old
Gender	*Customer's gender*	0: Male 1: Female
Annual Income	*Customer's annual income in dollars*	20k - 140k
Number of Purchases	*Total purchases made*	0 - 20
Product Category	*Category of the purchased product*	0: Electronics 1: Clothing 2: Home Goods 3: Beauty 4: Sports
Time Spent on Website	*Time spent in minutes*	0 - 59
Loyalty Program	*Loyalty program participation*	0: No 1: Yes
Discounts Availed	*Number of discounts availed by the customer*	0, 1, 2, 3, 4, 5

Target Variable		
Purchase Status	*Likelihood of the customer making a purchase*	0: No 1: Yes

Dataset contained **recent** information on customer purchase behavior across various attributes, aiming to help data scientists and analysts **understand the factors influencing purchase decisions**

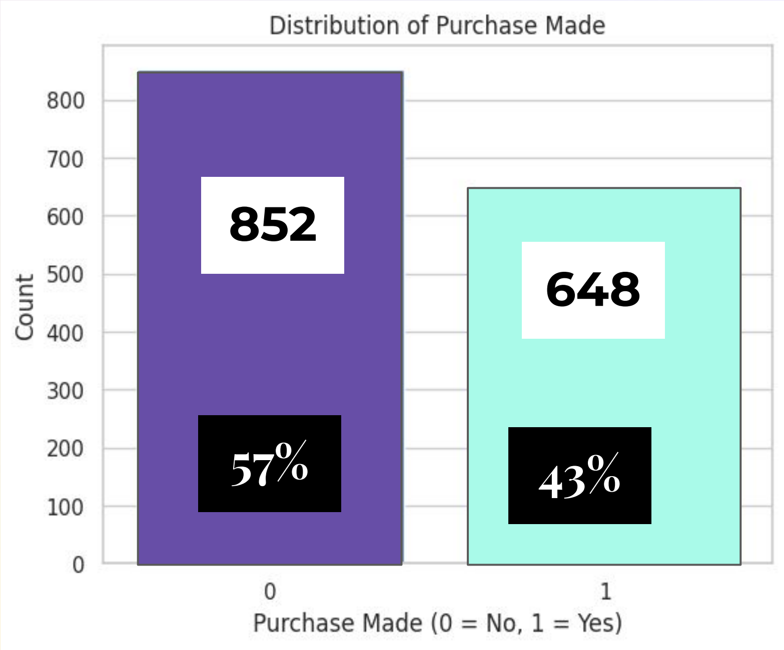
After removing Product Category and Gender, and with the exception of Loyal, we were left with numerical formats with no null values or rows

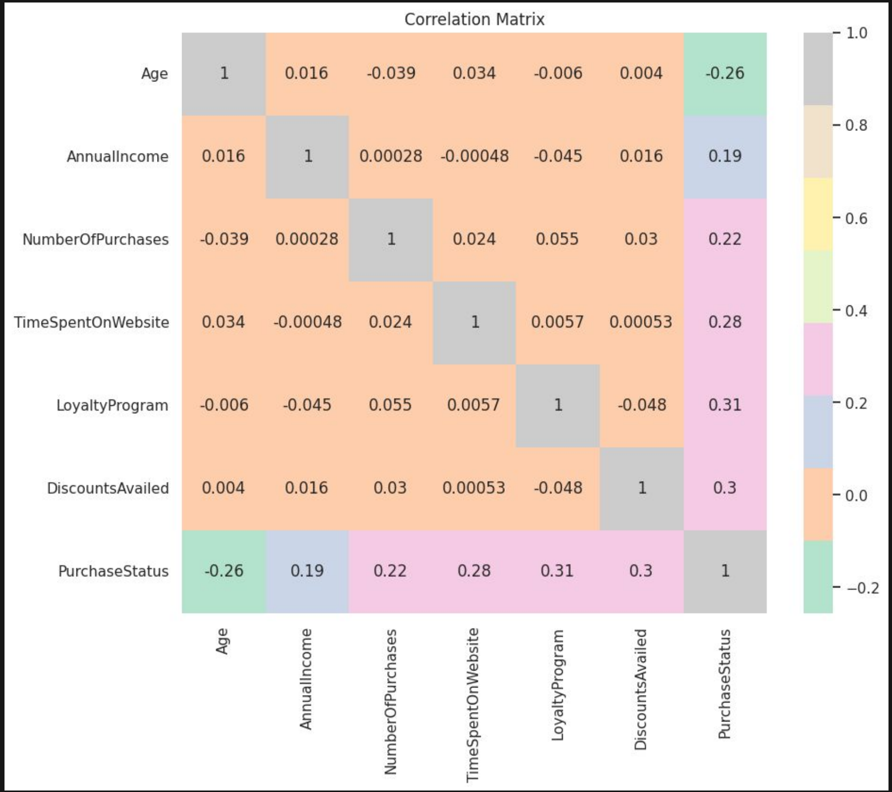
- **Scaled** numerical features to bring them to a similar range using StandardScaler.
- Conducted **Exploratory Data Analysis** (EDA) to understand the relationships in data
- **Split** Data into Training and Testing Sets

Rabie El Kharoua. (2024). Predict Customer Purchase Behavior Dataset [Data set]. Kaggle. https://doi.org/10.34740/KAGGLE/DSV/8725150

Exploratory Data Analysis (EDA)

Conducted EDA to understand the relationships in data





Implementation & Evaluation

Model Implementation

Model Selection

- **Logistic Regression:** Started with a simple model to establish a baseline performance
- **Decision Trees:** Great for understanding decision-making processes
- **Random Forest:** Often provides the best performance by averaging out the decisions of multiple trees

Model Evaluation

Evaluated the performance of the selected models using specific metrics

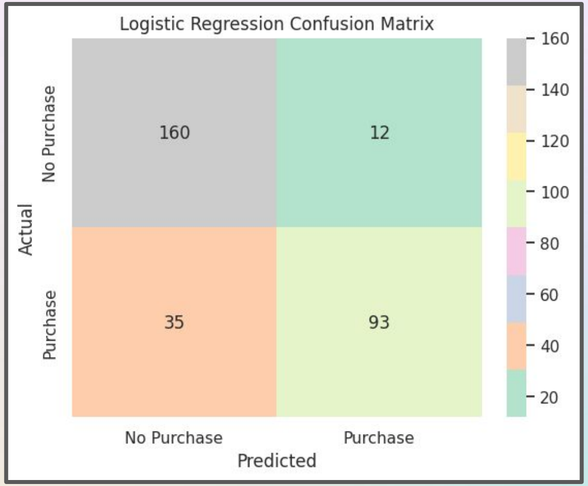
Defined a Function for Evaluation

To avoid repetitive code, we created a function that evaluates and prints these metrics for any model.

1. Accuracy
2. Precision
3. Recall
4. F1-score
5. ROC-AUC

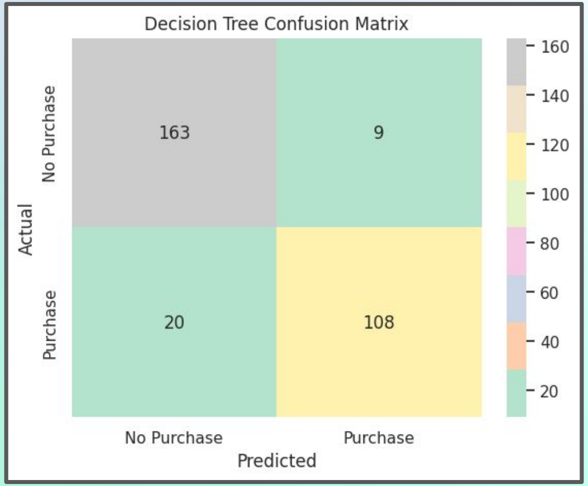
```python
from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score, roc_auc_score, roc_curve, confusion_matrix


def evaluate_model(y_true, y_pred, model_name):
    print(f"Evaluation Metrics for {model_name}:")
    print(f"Accuracy: {accuracy_score(y_true, y_pred):.4f}")
    print(f"Precision: {precision_score(y_true, y_pred):.4f}")
    print(f"Recall: {recall_score(y_true, y_pred):.4f}")
    print(f"F1 Score: {f1_score(y_true, y_pred):.4f}")
    print(f"ROC AUC Score: {roc_auc_score(y_true, y_pred):.4f}")

    # Confusion Matrix
    cm = confusion_matrix(y_true, y_pred)
    sns.heatmap(cm, annot=True, fmt='d', cmap='Pastel2', xticklabels=['No Purchase', 'Purchase'], yticklabels=['No Purchase', 'Purchase'])
    plt.title(f'{model_name} Confusion Matrix')
    plt.xlabel('Predicted')
    plt.ylabel('Actual')
    plt.show()

# Evaluate Logistic Regression
evaluate_model(y_test, y_pred_log_reg, "Logistic Regression")

# Evaluate Decision Tree
evaluate_model(y_test, y_pred_tree, "Decision Tree")

# Evaluate Random Forest
evaluate_model(y_test, y_pred_forest, "Random Forest")
```

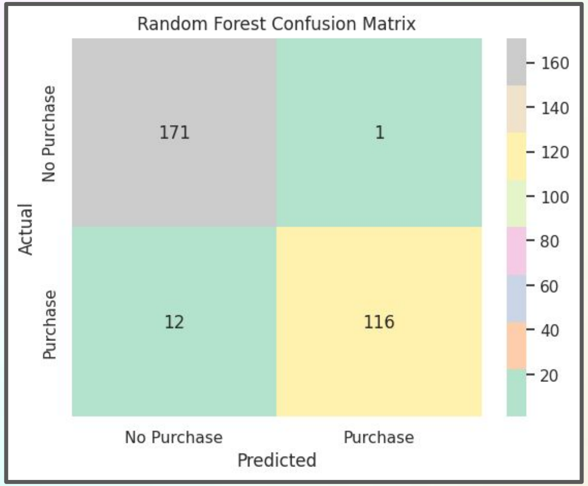
Evaluation Metrics Pre-Optimization



Evaluation Metrics for Logistic Regression:
Accuracy: 0.8433
Precision: 0.8857
Recall: 0.7266
F1 Score: 0.7983
ROC AUC Score: 0.8284

Evaluation Metrics for Decision Tree:
Accuracy: 0.9067
Precision: 0.9386
Recall: 0.8359
F1 Score: 0.8843
ROC AUC Score: 0.8976





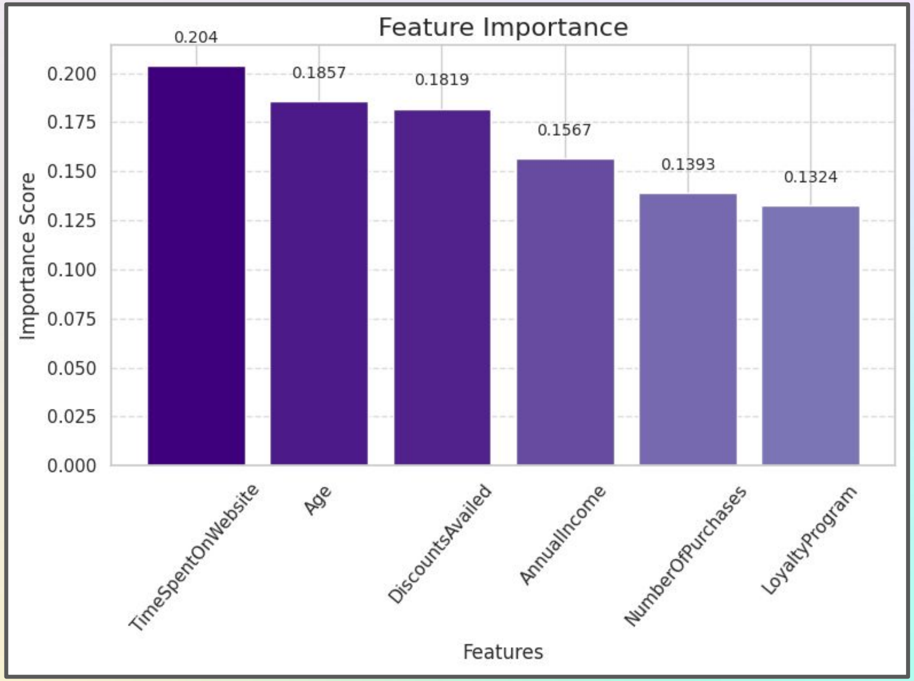
Evaluation Metrics for Random Forest:
Accuracy: 0.9567
Precision: 0.9915
Recall: 0.9062
F1 Score: 0.9469
ROC AUC Score: 0.9502

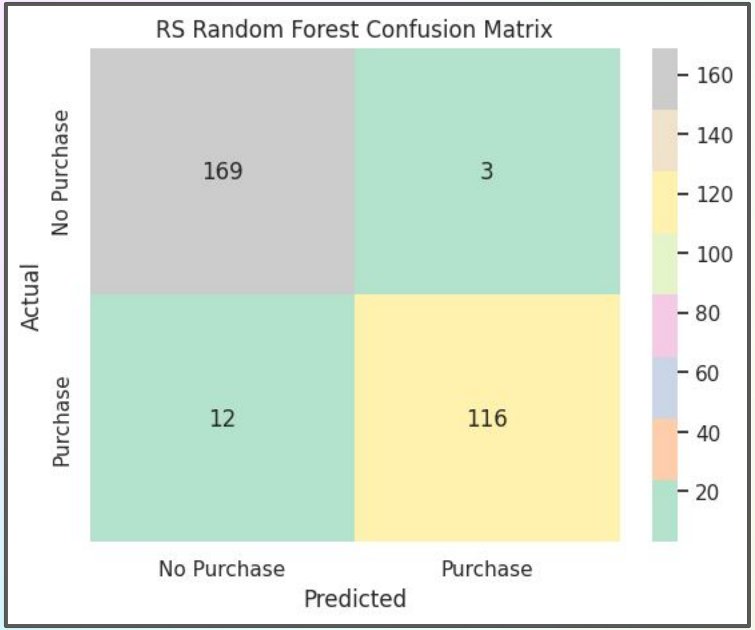
Model Optimization & Evaluation

Model Optimization

Hyperparameter Tuning

Random Search





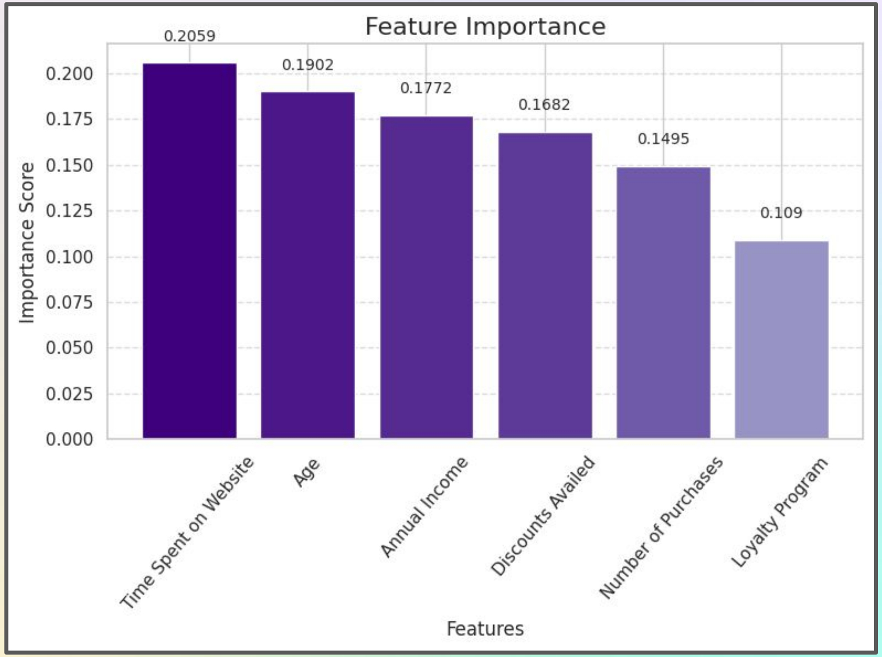
Evaluation Metrics for Random Search Model:

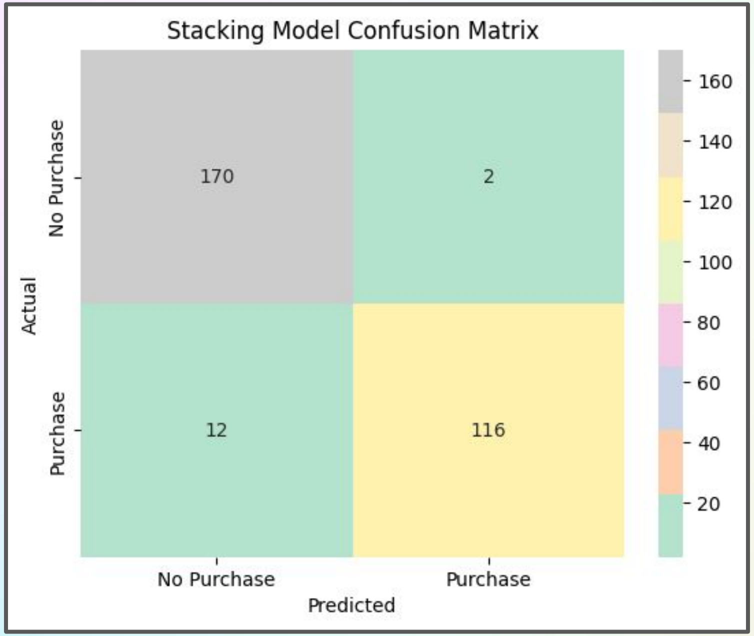
- Training: 0.9725
- Accuracy: 0.9500
- Precision: 0.9748
- Recall: 0.9062
- F1 Score: 0.9393
- ROC AUC Score: 0.9444

Model Optimization

Ensemble Methods

Stacking



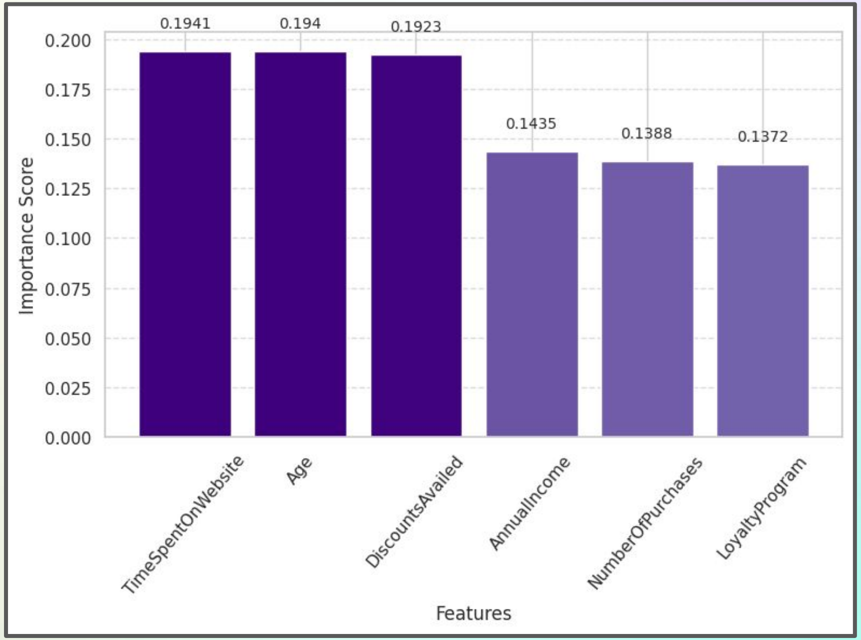


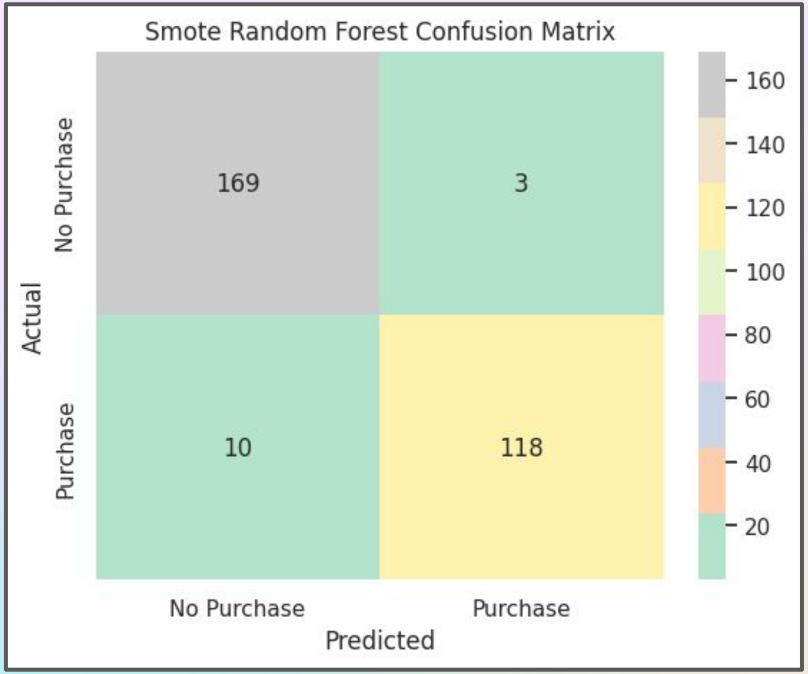
Evaluation Metrics for Stacking Model:

- Training: 0.9967
- Accuracy: 0.9533
- Precision: 0.9831
- Recall: 0.9062
- F1 Score: 0.9431
- ROC AUC Score: 0.9473

Model Optimization

Class Imbalance Handling SMOTE





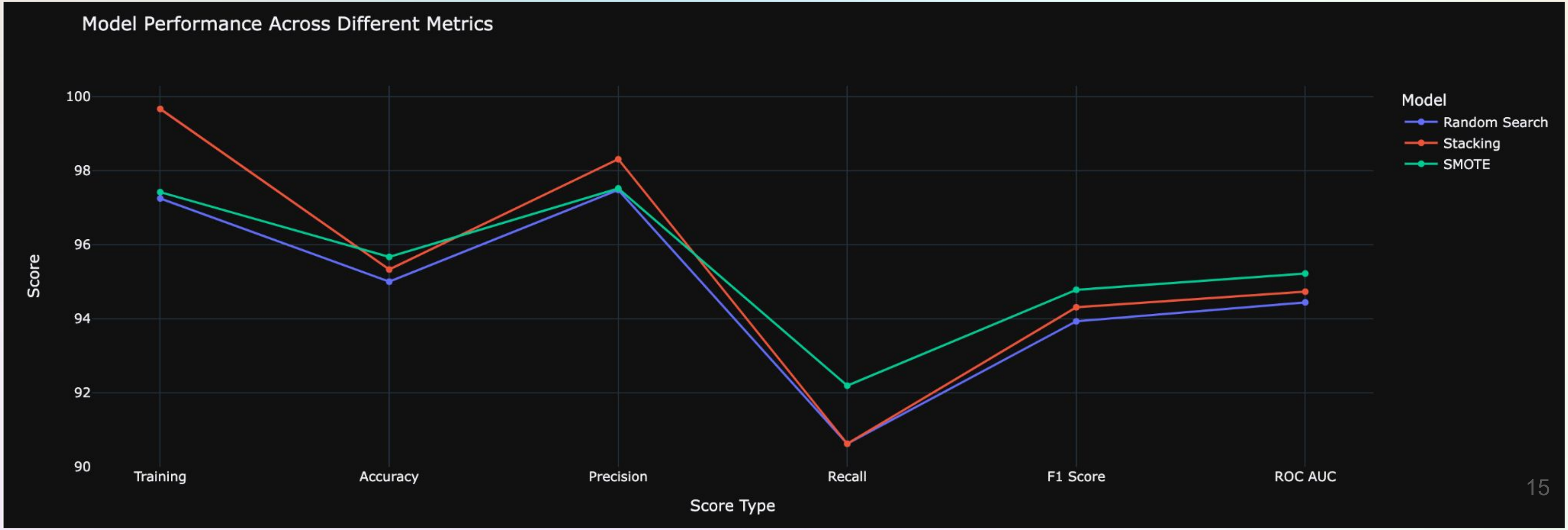
Evaluation Metrics for SMOTE Model:

- Training: 0.9742
- Accuracy: 0.9567
- Precision: 0.9752
- Recall: 0.9219
- F1 Score: 0.9478
- ROC AUC Score: 0.9522

Results / Conclusions

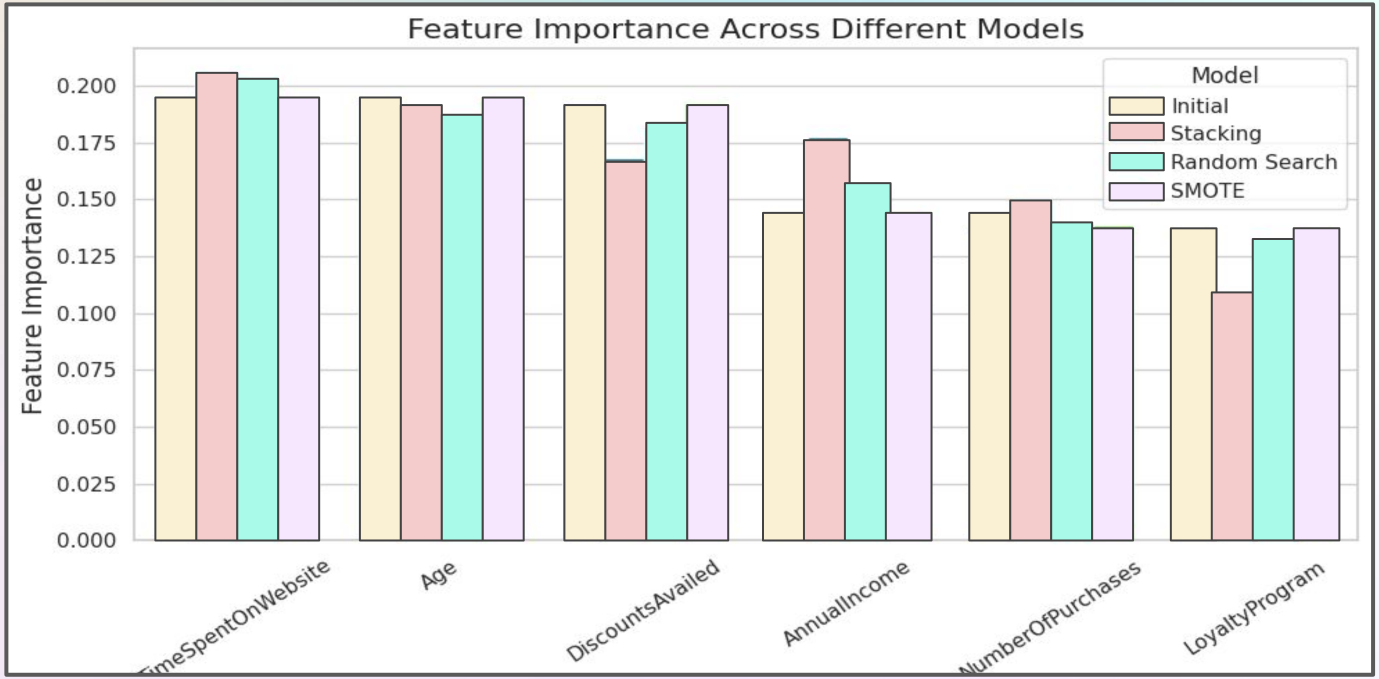
Model Optimization Results

Random Forest Model Optimization Scores			
Score Type	**Random Search**	**Stacking**	**SMOTE**
Training	97.25%	99.67%	97.42%
Accuracy	95.00%	95.33%	95.67%
Precision	97.48%	98.31%	97.52%
Recall	90.62%	90.62%	92.19%
F1 Score	93.93%	94.31%	94.78%
ROC AUC	94.44%	94.73%	95.22%



Feature Importance

Optimized Models Feature Important Values						
Model	**Time Spent on Website**	**Age**	**Discounts**	**Annual Income**	**# of Purchases**	**Loyalty Program**
Initial	0.194094	0.193980	0.192347	0.143550	0.138833	0.137196
Stacking	0.205891	0.190175	0.168159	0.177235	0.149491	0.10905
Random Search	0.203998	0.185685	0.181936	0.156703	0.139254	0.132424
SMOTE	0.194094	0.193980	0.192347	0.143550	0.138833	0.137196



Next steps

1. Experiment with More Complex Models:

- **Gradient Boosting (e.g., XGBoost, LightGBM):**
  - Marketing teams can experiment with these models to capture more intricate patterns in behavior
  - **Next Step:** Run experiments with these models and compare them against existing models using the same evaluation metrics (accuracy, precision, recall, F1-score, ROC AUC).

2. Explore Time-Series Models:

- **Time-Series Models (e.g., ARIMA, LSTM):**
  - For customers with recurring purchase patterns, time-series models can predict future purchases based on historical trends.
  - **Next Step:** Implement models that account for temporal data, such as using past purchasing behaviors to predict future buying tendencies. Long Short-Term Memory (LSTM) networks are ideal for capturing sequential data patterns.

3. Use K-Nearest Neighbors (KNN) for Behavioral Similarity:

- **KNN for Customer Segmentation:**
  - KNN can identify customers with similar behaviors and segment them for targeted marketing campaigns.
  - **Next Step:** Cluster customers with KNN to identify similar profiles and adjust marketing strategies to target specific segments with similar purchase behaviors.

Questions?

Thank You!